

**МЕЖРЕГИОНАЛЬНАЯ ОБЩЕСТВЕННАЯ ОРГАНИЗАЦИЯ
«СЕВЕРНАЯ ПРИРОДООХРАННАЯ КОАЛИЦИЯ»
(МРОО «СПОК»)**

УДК 630*61

**В. Е. Голубев, А. Ю. Зародов, А. В. Коросов,
А. В. Марковский, А. В. Родионов**

**РАЗРАБОТКА ПРИНЦИПОВ ФОРМИРОВАНИЯ АЛГОРИТМА
ДЛЯ ВЫЧИСЛЕНИЯ ПАРАМЕТРОВ РАСЧЕТНОЙ ЛЕСОСЕКИ
НЕИСТОЩИТЕЛЬНОГО ПОЛЬЗОВАНИЯ**

Петрозаводск 2014

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. КРАТКОЕ ОПИСАНИЕ ПРОЕКТА	4
2. ОПИСАНИЕ ПРИНЦИПОВ И АЛГОРИТМА ДЛЯ ВЫЧИСЛЕНИЯ ПАРАМЕТРОВ РАСЧЕТНОЙ ЛЕСОСЕКИ НЕИСТОЩИТЕЛЬНОГО ПОЛЬЗОВАНИЯ	7
2.1. Исходные предпосылки	7
2.2. Моделирование возрастной структуры насаждения	8
2.3. Моделирование рубок насаждений	10
2.4. Моделирование неистощительного лесопользования	13
2.5. Программная реализация разработанных моделей	18
2.6. Возможные направления дальнейших исследований	22
3. ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ВЫЧИСЛЕНИЯ ПАРАМЕТРОВ РАСЧЕТНОЙ ЛЕСОСЕКИ НЕИСТОЩИТЕЛЬНОГО ПОЛЬЗОВАНИЯ	24
3.1. Общие сведения о программе	24
3.2. Описание интерфейса пользователя программы	25
4. СРАВНИТЕЛЬНЫЙ АНАЛИЗ ДИНАМИКИ ИСПОЛЬЗОВАНИЯ ЛЕСОВ ДЛЯ МОДЕЛЬНЫХ ТЕРРИТОРИЙ	32
4.1. Общие сведения	32
4.2. Беломорское центральное лесничество	34
4.3. Сегежское центральное лесничество	40
4.4. Медвежьегорское центральное лесничество	46
4.5. Пудожское центральное лесничество	54
4.6. Выводы и рекомендации	62
ЗАКЛЮЧЕНИЕ	63
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	64
ПРИЛОЖЕНИЕ 1. Листинг программы на языке JavaScript	66

ВВЕДЕНИЕ

При рассмотрении вопросов неистощительного использования лесов в действующем законодательстве РФ большое внимание уделяется определению допускаемых объемов заготовки древесины – «расчетной лесосеки».

Приказом Рослесхоза от 27.05.2011 г. № 191 утвержден «Порядок исчисления расчетной лесосеки», с целью определения допустимого ежегодного объема изъятия древесины в эксплуатационных и защитных лесах.

Статья 29 «Лесного кодекса» РФ прямо запрещает заготовку древесины в объеме, превышающем расчетную лесосеку. Таким образом, по закону заготовка древесины в объемах, ведущих к истощению лесов, запрещается.

При этом утверждается, что расчетная лесосека определяет ежегодный объем изъятия древесины в эксплуатационных и защитных лесах, обеспечивающий многоцелевое, рациональное, непрерывное, неистощительное использование лесов, исходя из установленных возрастов рубок, сохранение биологического разнообразия, водоохранных, защитных и др. полезных свойств лесов.

Однако, по мнению ряда экспертов, существующие в РФ методики и подходы к определению расчетной лесосеки, практически не менявшиеся на протяжении XX века, не обеспечивают реальной неистощительности лесопользования по причине наличия существенных недостатков.

В связи с вышеизложенным, для России в целом и Республики Карелия в частности, актуальным является пересмотр действующих методик исчисления расчетной лесосеки и выработка новых подходов к определению допускаемых объемов заготовки древесины, обеспечивающих реальную неистощительность лесопользования, с учетом установленных возрастов рубок, закономерностей строения и развития древостоев и прогнозных данных о потребности в древесине.

Особое значение это имеет для традиционных «многолесных» районов страны (к которым принято относить и Карелию), имеющих развитую транспортную инфраструктуру и длительную историю истощительного освоения лесов.

В настоящей монографии представлены результаты работ по разработке принципов формирования алгоритма для вычисления параметров расчетной лесосеки неистощительного пользования на основе моделирования динамики развития лесных участков, подвергающихся регулярным антропогенным воздействиям в виде рубок спелых и перестойных лесных насаждений (т. н. «рубков главного пользования»).

1. КРАТКОЕ ОПИСАНИЕ ПРОЕКТА

В феврале 2014 г. «Всемирный фонд природы» одобрил грантовое соглашение RU 009309-FY-13 на сумму 400,0 тыс. рублей на выполнение проекта «Разработка принципов формирования алгоритма для вычисления параметров расчетной лесосеки неистощительного пользования для лесных участков, включающих малонарушенные лесные территории». Руководитель – консультант Межрегиональной общественной организации «Северная природоохранная коалиция» (МРОО «СПОК»), А. В. Родионов.

Обоснование необходимости проекта:

Согласно экспертным оценкам, особенно остро истощительность пользования проявляется при пионерном освоении лесов, когда значительную часть аренды предприятия занимают малонарушенные лесные территории. Для обеспечения долговременного сохранения малонарушенных лесных территорий необходима разработка новых принципов исчисления предельных объемов пользования лесом, учитывающих естественную динамику лесов и соблюдающих принцип непрерывности пользования лесом на протяжении всего цикла его развития.

В связи с этим, актуальным является анализ действующих методик определения расчетной лесосеки и выработка новых подходов к определению предельных объемов заготовки древесины, обеспечивающих реальную неистощительность лесопользования в пределах конкретных территорий. Особое значение это имеет для лесных предприятий, сертифицированных или сертифицируемых по системе FSC (поскольку наличие сертификата FSC подразумевает соблюдение предприятием не только норм и требований действующего законодательства, но и принципов, критериев и требований национального стандарта FSC).

Предполагается разработать альтернативные алгоритмы исчисления объемов пользования и обеспечить их публичную доступность путем размещения программы для расчета в сети Интернет.

Объект и предмет исследований проекта:

Объект исследований – научные основы обеспечения многоцелевого, рационального, непрерывного, неистощительного использования лесов, сохранения биологического разнообразия, водоохранных, защитных и иных полезных свойств лесов.

Предмет исследований – методы определения допустимого ежегодного объема изъятия древесины, обеспечивающие рациональное, непрерывное, неистощительное

использование лесов, исходя из установленных возрастов рубок, закономерностей строения и развития древостоев.

Исходные нормативные документы исследований проекта:

- Приказ Рослесхоза от 19.02.2008 г. N 37 (ред. от 29.12.2011 г.) «Об установлении возрастов рубок»;
- Приказ Рослесхоза от 27.05.2011 г. N 191 «Об утверждении Порядка исчисления расчетной лесосеки»;
- Экспресс-методика оценки неистощительности объема лесопользования в пределах арендного участка сертифицированного (проходящего процедуры сертификации) по системе FSC.

Цель исследований проекта:

Анализ закрепленных в действующем законодательстве методик определения расчетной лесосеки и выработка на их основе новых подходов к определению предельных объемов заготовки древесины, обеспечивающих рациональное, непрерывное, неистощительное использование лесов, исходя из установленных возрастов рубок, закономерностей строения и развития древостоев и прогнозных данных о потребности в древесине.

Этапы исследований проекта:

Этап 1. Февраль-март 2014 г. Разработка новых принципов формирования алгоритма для вычисления параметров расчетной лесосеки на основе данных о лесном фонде.

Этап 2. Апрель 2014 г. Разработка программы (автономной программы и интернет-версии) для вычисления параметров расчетной лесосеки на основе новых подходов к определению расчетной лесосеки, ретроспективных данных о лесном фонде и прогнозных данных о потребности в древесине.

Этап 3. Май 2014 г. Сравнительный анализ действующей и авторской методик определения расчетной лесосеки с использованием разработанной программы расчетов.

Ожидаемые результаты исследований проекта:

- Новые алгоритмы и математические модели на их основе для вычисления параметров объемов заготовки древесины (т. н. «расчетной лесосеки») на основе ретроспективных и прогнозных данных о лесном фонде и потребности в древесине и т.п.;
- Программа (в виде автономной программы и интернет-версии) для вычисления параметров расчетной лесосеки на основе разработанных подходов к определению расчетной лесосеки;
- Сравнительный анализ динамики использования лесов для модельных

территорий с использованием новых и традиционных моделей для определения параметров расчетной лесосеки;

- Рекомендации по выбору наилучшего варианта (и/или оптимизации) величины расчетной лесосеки на основе новых математических моделей;

- Рекомендации по дальнейшим направлениям исследований в указанном направлении.

2. ОПИСАНИЕ ПРИНЦИПОВ И АЛГОРИТМА ДЛЯ ВЫЧИСЛЕНИЯ ПАРАМЕТРОВ РАСЧЕТНОЙ ЛЕСОСЕКИ НЕИСТОЩИТЕЛЬНОГО ПОЛЬЗОВАНИЯ

2.1. Исходные предпосылки

Исходя из современных требований рыночных отношений в области освоения лесов, «расчетная лесосека» для эксплуатационных лесов в лесничествах должна определяться с целью гарантировать устойчивое (например, постоянное неснижаемое) обеспечение перерабатывающих предприятий лесного сектора необходимым древесным сырьем на протяжении заранее заданного промежутка времени.

Задачи сохранения биологического разнообразия, обеспечения водоохранных, защитных и др. полезных свойств лесов в условиях лесничеств должны выполняться преимущественно за счет защитных лесов. Вопрос о необходимой площади таких лесов на территории соответствующих лесничеств является весьма актуальным, но выходит за рамки настоящего анализа.

Предлагаемый базовый подход к определению «расчетной лесосеки» в эксплуатационных лесах лесничеств позволяет нам предложить следующее понимание идеи «непрерывного неистощительного лесопользования» применительно к таким лесам.

«Непрерывное неистощительное лесопользование» в эксплуатационных лесах определяется как возможность использования лесных ресурсов с постоянной высокой эффективностью (в т. ч. при полном задействовании естественных продуктивных сил леса) на протяжении заранее заданного промежутка времени или бесконечно. При этом по окончании заданного промежутка времени лес не должен физически исчезнуть на занимаемых площадях, просто его возможности снабжать промышленность лесным сырьем (древесиной) требуемого качества должны полностью исчерпаться.

Очевидно, что чем короче планируемый период лесопользования, тем выше возможен уровень использования лесных ресурсов до полного истощения возможности снабжать лесным сырьем (древесиной) требуемого качества.

Поэтому, здесь должны быть установлены ограничения – заданная продолжительность периода лесопользования (эксплуатации) и постоянство (т. е. устойчивость) максимально возможного изъятия лесных ресурсов на этот период – что диктует необходимость пересмотреть применяемые на современном этапе методы определения «расчетной лесосеки» с помощью разнообразных формул.

Предлагаемый подход к определению «расчетной лесосеки» для

эксплуатационных лесов базируется на имитационной модели динамики всей возрастной структуры насаждений – от начальных классов возраста (т. н. «молодняки») до старших классов (т. н. «спелые»).

В отличие от применяемых ныне формул расчета лесосеки, основанных на мысленной модели истощения запасов перестойных, спелых, приспевающих и др. насаждений через то или иное время, имитационная модель рассчитывает этот процесс во всех деталях относительно всех разновозрастных групп деревьев. По существу в числовой форме (виртуально, в недрах компьютера) воссоздается вся возрастная структура древостоя на протяжении любого отрезка времени. Вырубки рассматриваются как нарушения возрастной структуры, и модель призвана рассчитать их последствия, а также дать возможность решать задачи расчета оптимальной величины «расчетной лесосеки» при задаваемых исследователем ограничениях.

2.2. Моделирование возрастной структуры насаждения

В отдельный момент времени древостой одного возраста занимает площадь a_{ij} (i – момент времени, j – возраст, a – площадь). В отдельный момент времени i возрастная структура насаждения представлена набором значений относительной площади древостоев разного возраста.

Сумма относительных площадей всех участков разновозрастных древостоев составляет единицу (вся площадь, занимаемая насаждением), т. е.:

$$\sum_j a_{i,j} = 1. \quad (1)$$

В следующий момент времени $i+1$ моделируется переход древостоев каждого возрастного класса в более старший $a_{i+1, j+1} = a_{i,j}$ – древостой на некоем участке на следующем временном шаге переходит в следующий возрастной класс. В модели выбран самый низкий уровень возрастной и временной шкалы – один год.

В модели учитываются процессы естественного отмирания (s_{dj}), нарушения из-за внешних (включая антропогенные) случайных воздействий (s_{aj}), последствия рубок (s_{cj}). Они задаются в форме коэффициентов выживания (s – survival; d – death, a – anthropogenic, c – cut). В зависимости от возраста (j) эти коэффициенты могут меняться.

Значения коэффициентов, учитывающих процессы естественного отмирания (s_{dj}) и нарушений из-за внешних случайных воздействий (s_{aj}), выбираются по соответствующей специальной литературе и помещаются во встроенных базах данных при программной реализации математической модели.

Например, для древостоя сосны обыкновенной в возрасте 5 лет вероятность

выживания составляет 100 %, $s_{d5} = 1$; вероятность нарушения внешними случайными воздействиями составляет 5 % (а сохранения – 95 %; $s_{a5} = 0,95$); вероятность быть вырубленным – нулевая ($s_{c5} = 1$). Аналогичные цифры для 100 лет: $s_{d100} = 1,00$; $s_{a100} = 0,95$; s_{c100} = от 0 до 1.

Полная формула перехода древостоев каждого возрастного класса в более старший имеет вид:

$$a_{i+1,j+1} = s_{dj} \cdot s_{aj} \cdot s_{cj} \cdot a_{ij}. \quad (2)$$

Значения **коэффициентов естественного отмирания** (s_{dj}) назначаются, при программной реализации математической модели, исходя из данных о продолжительности жизни отдельных деревьев, древостоев в целом и данных о динамике прироста запасов древостоев на корню.

В литературе [1] имеются указания на то, что запас наиболее характерных для Европейской части России лесообразующих хвойных пород (сосны и ели) постепенно увеличивается до возраста естественной спелости, затем начинает уменьшаться, в возрасте около 180 лет прирост становится практически нулевым. Известно [2], что возраст естественной спелости у деревьев сосны или ели наступает к 300-350 годам, а у их насаждений — к 200-250 годам. Из работ [3, 4] также известно, что береза живет в среднем до 100-120 лет, осина – до 80-100 лет. В модели принято, что предельный возраст хвойных (сосна, ель) насаждений составляет 250 лет, березовых – 130 лет, осиновых – 100 лет.

С учетом изложенного, для программной реализации математической модели, были приняты параметры отмирания наиболее характерных для Европейской части России лесообразующих пород деревьев – см. таблицу 2.1.

Таблица 2.1 – Параметры отмирания некоторых лесообразующих пород деревьев

Показатель	Порода		
	сосна, ель	береза	осина
Возраст начала вымирания, лет	180	90	70
Возраст полного вымирания, лет	250	130	100
Уровень ежегодной гибели	0,05	0,08	0,11

Переводя вышеизложенные факты в используемые в модели единицы площади, приходим к следующим оценкам (для сосны): до возраста насаждений 180 лет естественной убыли площади древостоя не происходит ($s_{d,0-180} = 1$); начиная со 180 лет задается прогрессивное снижение площади на 5% (сохранение – 95 %) $s_{d,180-249} = 0,95$.

К 249 годам доля перестойных древостоев снизится до 3 % от доли в 180 лет. Для возраста 250 лет принята убыль 100 % – $s_{d,250} = 0$.

Значения **коэффициентов случайного отмирания** (s_{dj}) назначаются, при программной реализации математической модели, исходя из данных исследований случайной гибели древостоев разного возраста. Например, по данным [5], за период 2004–2008 гг. в Финляндии серьезно пострадало 4,4 % лесных площадей, умеренно пострадало – 23,9 %. Мы приняли значение 5 % (сохранение 95 %) – $s_{aj} = 0,95$.

Однако, проблема определения конкретных значений коэффициента случайного отмирания (s_{dj}) является весьма актуальной (см. например [6, 7]), и требует дополнительного исследования.

Значения **коэффициентов гибели от рубок** (s_{cj}) и составляют предмет модельного исследования. Они варьируются в модели с целью поиска оптимального варианта лесопользования. Эта технология рассмотрена ниже.

Исходя из формулы перехода древостоев в более старший возрастной класс на каждом i -м шаге, общие потери площадей древостоя составят, соответственно

$$(a_{ij} - s_{dj} \cdot s_{aj} \cdot s_{cj} \cdot a_{ij}). \quad (3)$$

Любые нарушения сплошного растительного покрова (вывалы и ветровалы, пожары и рубки) ведут к появлению территорий без деревьев, которые рассматриваются как пополнение площади (доли) древостоев нулевого возрастного класса ($a_{i,0}$). Поскольку такие нарушения возможны на территории древостоя любого возраста, нулевой класс в следующий ($i+1$) момент времени сформирован суммой площадей с нарушенным древостоем по всем возрастным классам:

$$a_{i+1,0} = \sum_j (a_{ij} - s_{dj} \cdot s_{aj} \cdot s_{cj} \cdot a_{ij}). \quad (4)$$

Результатом расчетов по модели является таблица из значений $a_{i,j}$ размером $m \cdot k$, где m – число шагов модели (длина периода времени для прогноза); k – число возрастных классов. Набор этих значений можно анализировать, представлять в виде диаграмм и пр., а также использовать при настройке параметров функционирования при оценке возможностей неистощительного лесопользования.

2.3. Моделирование рубок насаждений

Вырубки представляют собой изъятие деревьев старших возрастных классов (в примере – старше 80 лет) и переход этих площадей в категорию нулевого возрастного класса. Однако вырубаться может только лес на корню, из площади которого уже

вычли потери от действия внутренних и внешних факторов ($s_{dj} \cdot s_{aj} \cdot a_{ij}$):

$$c_{i+1,j>80} = \sum_{j>80} (s_{dj} \cdot s_{aj} \cdot a_{ij} - s_{dj} \cdot s_{aj} \cdot s_{cj} \cdot a_{ij}). \quad (5)$$

Зная запас древесины на 1 га для каждого возрастного класса (R_j), несложно рассчитать объемы рубки древесины (v_{i+1}) на каждом временном шаге модели:

$$v_{i+1} = \sum (R_j \cdot c_{i+1,j>80}). \quad (6)$$

Запас древесины на 1 га для древостоев всех классов возраста при программной реализации модели определяется по математическим моделям хода роста соответствующих насаждений, построенным на основе общих таблиц хода роста нормальных древостоев [1].

Поскольку в лесохозяйственных регламентах центральных лесничеств для хвойных пород используются две категории продуктивности: 1-3 и 4-5 классы бонитетов, а для лиственных – одна категория продуктивности, то для построения математических моделей хода роста значения из [1] усреднялись – см. в таблице 2.2.

Таблица 2.2 – Запасы стволовой древесины в нормальных древостоях, кмб/га

Возраст, лет	сосна		ель		береза	осина
	1-3 бон.	4-5а бон.	1-3 бон.	4-5а бон.	1-5 бон.	1-5 бон.
10	28		19	-	18	34
20	79	25	64	-	51	82
30	137	44	135	21	88	132
40	202	68	226	36	123	177
50	266	92	326	56	155	218
60	320	116	426	79	182	252
70	367	137	521	103	204	280
80	405	156	604	128	222	302
90	439	172	675	152	237	320
100	466	186	735	175	249	335
110	486	197	783	195	-	-
120	504	206	820	211	-	-
130	520	215	852	225	-	-
140	534	223	878	238	-	-
150	547	230	901	249	-	-
160	558	236	921	259	-	-

К сожалению, указанные материалы (см. таблицу 2.2) недостаточны для непосредственного моделирования хода роста соответствующих древостоев по двум причинам. Во-первых, они не дают представления о запасе в древостоях старше 160 лет, во-вторых, запас т. н. «нормального древостоя» существенно выше запаса реальных древостоев.

Например, для сосны 4-5 классов бонитета нормальный запас в возрасте 100 лет составляет 186 кбм/га (см. таблицу 2.2), а по лесохозяйственному регламенту Медвежьегорского центрального лесничества [8] для спелых и перестойных сосновых насаждений 4-5 бонитета в эксплуатационных лесах – 154 кбм/га.

По этим причинам данные из таблиц хода роста нормальных древостоев [1] были преобразованы нами следующим образом.

Во-первых, по имеющимся исходным данным (см. таблицу 2.2) были рассчитаны кривые изменения запасов древесины с возрастом; использованы уравнения полиномов 2-й степени вида:

$$R_n = a_0 + a_1 \cdot i + a_2 \cdot i^2. \quad (7)$$

где a – коэффициенты полинома (см. таблицу 2.3), i – возраст древостоя.

Таблица 2.3 – Коэффициенты для кривых изменения запасов
в нормальных древостоях

Коэффициент	сосна		ель		береза	осина
	1-3 бон.	4-5а бон.	1-3 бон.	4-5а бон.	1-5 бон.	1-5 бон.
a_0	-56.4111	-16.5509	-120.995	-12.9605	-27.9833	-29.55
a_1	7.589699	2.584391	8.631785	2.520494	6.153719	4.539637
a_2	-0.02381	-0.00616	-0.02038	-0.00589	-0.02527	-0.01746

Полученные кривые изменения запаса позволили, в частности, рассчитать возможный запас нормального древостоя для возрастных групп старше 160 лет. Уравнения получены: для сосны и ели двух групп бонитетов (1-3 и 4-5а), для осины и березы всех классов бонитета; всего 6 уравнений.

Во-вторых, был введен поправочный коэффициент (K) для расчета кривых изменения запасов древостоя с возрастом для условий каждого конкретного лесничества ($R_{л}$), исходя из простого соображения, что эта величина будет отличаться от запаса нормального древостоя (R_n) на величину:

$$K = R_{л} / R_n. \quad (8)$$

Поправочный коэффициент вычисляется исходя из соотношения величины

запаса, приведенной в лесохозяйственном регламенте (для древостоя в указанном возрасте) и величины запаса, рассчитанной по модели (см. формулу (7) и таблицу 2.3) для этого возраста для нормального древостоя.

Например, для насаждений сосны 4-5 бонитета Медвежьегорского центрального лесничества [8] в возрасте 100 лет модель дает «нормальный» запас, равный

$$R_{n100} = -16,5509 + 2,584391 \cdot 100 - 0,00616 \cdot 100^2 = 180,3 \text{ кбм/га},$$

а в лесохозяйственном регламенте указана величина 154 кбм/га, следовательно, искомый коэффициент равен

$$K = R_{л100} / R_{n100} = 154 / 180,3 = 0,854248.$$

В дальнейшем этот и другие коэффициенты (см. таблицу 2.4) используются для определения запасов сосновых древостоев 4-5 бонитета эксплуатационных лесов Медвежьегорского центрального лесничества любого возраста i по соотношению:

$$R_{ли} = K \cdot R_{ni}. \quad (9)$$

Таблица 2.4 – Поправочные коэффициенты для Медвежьегорского центрального лесничества

Коэффициент	сосна		ель		береза	осина
	1-3 бон.	4-5а бон.	1-3 бон.	4-5а бон.	1-5 бон.	1-5 бон.
K	0,637542	0,854248	0,585236	0,960368	0,82142	0,922408

Аналогичным образом вычисляются поправочные коэффициенты для условий каждого конкретного лесничества. При программной реализации математической модели разработан соответствующий блок программы, рассчитывающий эти коэффициенты. От пользователя требуется ввести значения запаса на 1 га для спелых и перестойных древостоев, приведенного в лесохозяйственном регламенте данного региона, программа сравнивает введенное значение с табличными и рассчитывает необходимый поправочный коэффициент для всех возрастных групп.

2.4. Моделирование неистощительного лесопользования

При вычислении параметров «расчетной лесосеки», обеспечивающей непрерывное неистощительное лесопользование в эксплуатационных лесах, возможно рассматривать два варианта.

Вариант I. «Непрерывное неистощительное лесопользование» в эксплуатационных лесах можно понимать как ежегодное изъятие *максимально*

возможного объема древесины на протяжении заданного числа лет путем ежегодной вырубki территории одного и того же размера (т. е. постоянство рубок по площади [9]).

Этим постулируется, что площадь рубок на протяжении всего периода прогноза должна быть постоянной и максимально возможной; например для спелых древостоев:

$$c_{i,j>80} = \text{const} \rightarrow \max. \quad (10)$$

При этом в зависимости от запасов разновозрастной древесины и закономерного изменения возрастного состава общий вырубаемый объем древесины (v) будет год от года варьировать.

Вариант II. «Непрерывное неистощительное лесопользование» в эксплуатационных лесах можно понимать как изъятие *максимально возможного и постоянного объема древесины* на протяжении заданного числа лет путем ежегодной вырубki территории различных размеров (т. е. постоянство рубок по объему древесины [9]).

Этим постулируется, что площадь рубки на протяжении периода прогноза может варьировать, но общий вырубаемый объем древесины должен быть постоянным и максимально большим:

$$v_i = \text{const} \rightarrow \max. \quad (11)$$

При этом в зависимости от запасов разновозрастной древесины и закономерного изменения возрастного состава год от года варьировать будет уже площадь рубок.

Электронные компьютерные средства позволяют рассчитать не один, но множество сценариев, при которых объемы рубок могут варьировать. Каждый из сценариев будет иметь свой критерий реализации. Если критерий количественно определен, он вводится в процедуру оптимизации, которая подбирает такие параметры модели (объемы и/или площади рубок), чтобы заданный критерий был выполнен.

При этом могут быть решены три основных задачи.

1. Вычислить продолжительность периода времени, в течение которого можно постоянно изымать произвольно заданный объем древесины требуемого возраста (в гектарах площади необходимых вырубок или в кубометрах необходимого объема заготовок); эта задача не ограничена никакими критериями и выполняется как простой расчет динамики древостоя при рубках той или иной интенсивности.

Расчет останавливается, когда запасы древесины требуемого возраста будут полностью истощены; это и будет возможная продолжительность рубок.

Обычная динамика возрастной структуры насаждения сводится к тому, что если принять расчетную лесосеку т. н. «главного пользования», вычисленную по известным

формулам [10], то ресурсы спелых и перестойных древостоев заканчиваются через 35-50 лет – см. рисунок 2.1.

2. Вычислить максимально возможную ежегодную площадь расчетной лесосеки, которая будет постоянно доступна для рубки в течение произвольно заданного периода времени (без перерывов/колебаний в объеме лесопользования в указанный период); в этой задаче критерием является минимальная площадь доступных для вырубki лесов в каждый прогнозируемый год, которая не должен быть меньше, чем ежегодная площадь расчетной лесосеки (каждый год постоянная на протяжении всего периода прогноза).

Обычная динамика возрастной структуры насаждения демонстрирует плавное снижение площади древостоев (например, спелых и перестойных), назначенных в рубку, вплоть до полного их сведения в последний год прогнозируемого периода – см. рисунок 2.2.

3. Вычислить площадь расчетной лесосеки, вырубka которой будет обеспечивать ежегодно максимально возможный и постоянный объем древесины в течение произвольно заданного периода времени (без перерывов/колебаний в объеме лесопользования в указанный период); критерием в этой задаче также является минимальная площадь доступных для вырубki лесов в каждый прогнозируемый год, которая не должна быть меньше, чем ежегодная площадь расчетной лесосеки (каждый год постоянная на протяжении всего периода прогноза).

Отличие от предыдущего варианта состоит в том, что при изменении возрастного состава и, следовательно, запасов на корню, для обеспечения заготовок заданного объема древесины в разные годы потребуются вырубать разные площади.

Обычная динамика выглядит вначале как постепенное возрастание площадей рубок (по мере вырубания более продуктивных старших возрастных групп и замене их менее продуктивными младшими) и последующая стабилизация площадей рубок, когда старшие древостои будут в основном сведены – см. рисунки 2.3, 2.4.

Реализация представленных моделей показала, что абсолютная неистощимость и максимальный выход продукции (древесины) несовместны – весьма важным параметром является продолжительность планируемого периода «непрерывного неистощительного лесопользования».

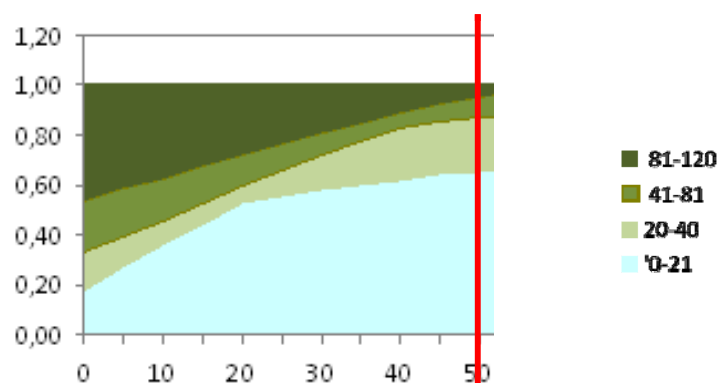


Рисунок 2.1 – Изменение соотношения площадей четырех возрастных групп насаждения при использовании завышенных размеров расчетной лесосеки. По оси ординат – доля площади данной группы от всей площади изучаемого района. По оси абсцисс – годы жизни древостоя. Красной чертой отмечена дата, после которой расчетная лесосека становится не обеспеченной лесными ресурсами

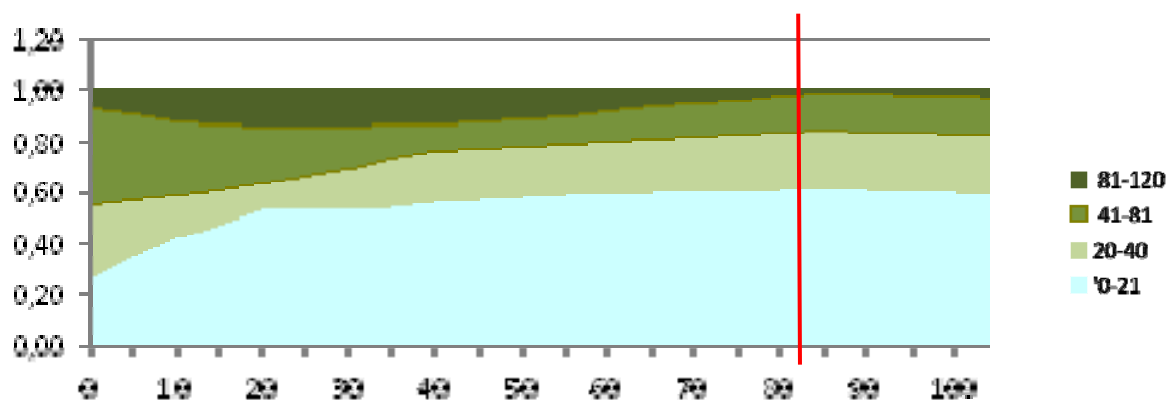


Рисунок 2.2 – Изменение соотношения площадей четырех возрастных групп насаждения при условии постоянства площади расчетной лесосеки т. н. «главного пользования» в течение 100 лет. По оси ординат – доля площади данной группы от всей площади изучаемого района. По оси абсцисс – годы жизни древостоя. Красной чертой отмечена дата, после которой расчетная лесосека становится не обеспеченной лесными ресурсами

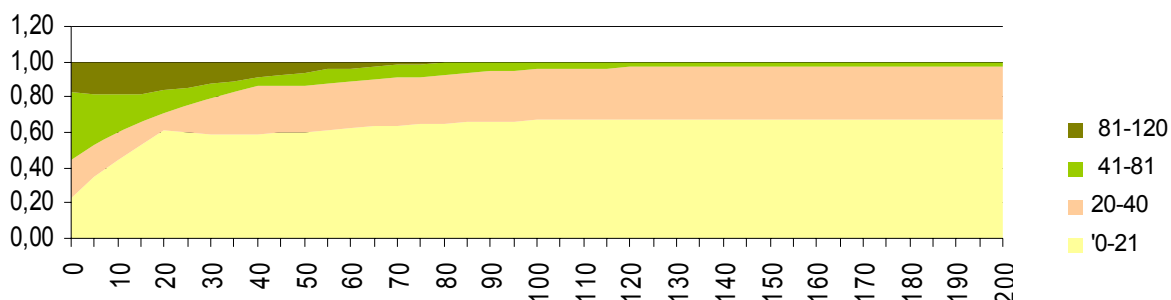


Рисунок 2.3 – Изменение соотношения площадей четырех возрастных групп насаждения при условии понижении возраста рубки и постоянства объемов заготовки древесины в течение 200 лет. По оси ординат – доля площади данной группы от всей площади изучаемого района. По оси абсцисс – годы жизни древостоя

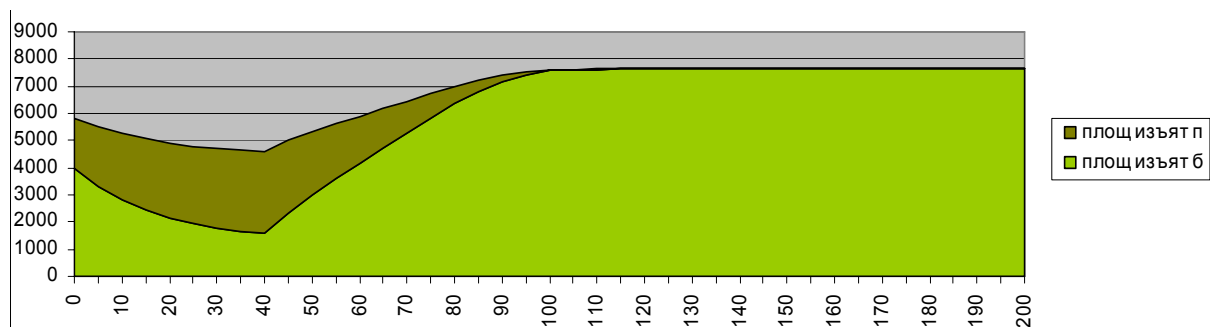


Рисунок 2.4 – Изменение соотношения площадей рубок «старших», «п» (спелых) и «младших», «б» возрастных групп в насаждении при условии понижении возраста рубки и постоянства объемов заготовки древесины в течение 200 лет. По оси ординат – доля площади данной группы от всей площади изучаемого района. По оси абсцисс – годы жизни древостоя

2.5. Программная реализация разработанных моделей

Полученные математические модели (см. выше) апробировались в трех программных средах.

Среда Excel. Построение модели возрастной структуры насаждения и проверку ее работоспособности проводили в среде пакета MS Excel. За основу была взята технология *табличного программирования* при котором на листе Excel строится имитационная система, открывающая возможность настройки модельных параметров встроенными методами оптимизации пакета [11].

При этой технологии сама модель (матрица возрастной структуры) строиться на листе Excel как набор множества стереотипных формул, вычисляющих переход одновозрастной группы во все более старшие возрастные классы с учетом потерь от различных факторов (отмирания, нарушения, рубки).

Модель настраивалась с помощью встроенного макроса процедуры оптимизации «Поиск решения». В такой форме модель имеет прозрачную структуру и средства иллюстрации и доступна для предметного обсуждения специалистами разных профилей – см. рисунок 2.5.

Помимо отмеченных преимуществ, наглядности и прозрачности, модель обладает отрицательным качеством – крайней громоздкостью; в таком виде она не годится для программирования на алгоритмических языках. По этой причине было выполнено переформулирование технологической задачи в среде языка R.

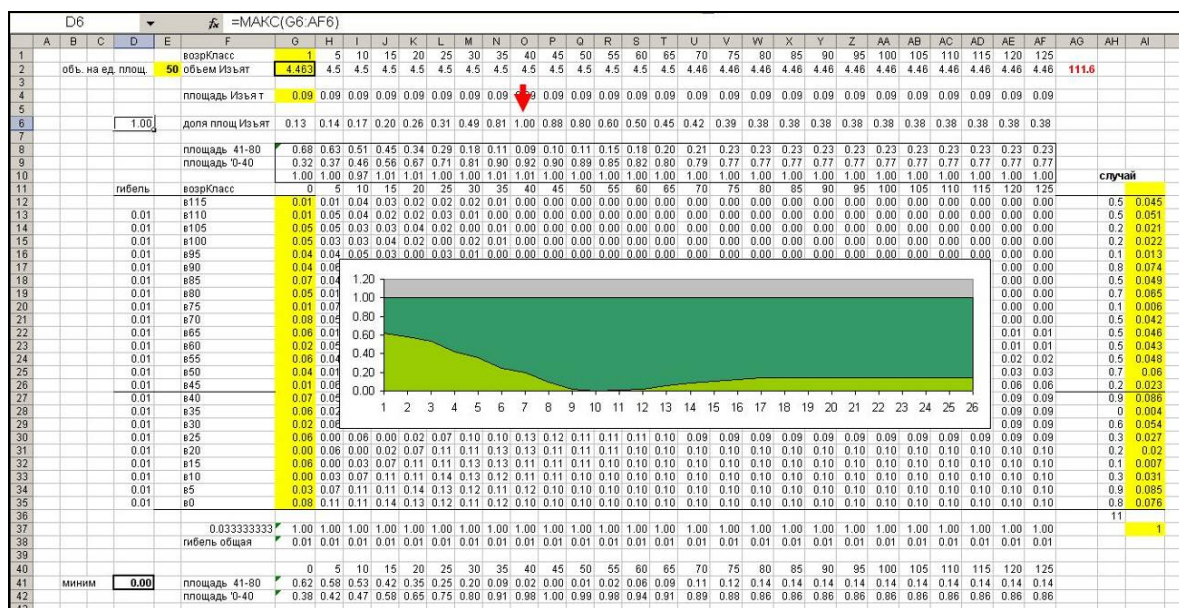


Рисунок 2.5 – Общий вид имитационной модели в среде Excel

Среда R. Язык R ориентирован на обработку векторов и матриц и по этой причине программы на этом языке имеют очень компактный формат [12]. Из числа математических средств моделирования возрастной структуры известны матричные модели [13].

Основной модели (см. рисунок 2.6) является, по-прежнему, таблица возрастной структуры древостоя в каждый момент времени (массив `nper`) размером $m \cdot k$, где m – число шагов модели (длина отрезка времени для прогноза); k – число возрастных классов. Второй важнейший объект – матрица перехода возрастных групп в более старшие, «коэффициенты выживания» (матрица `se`). Матрица перехода содержит коэффициенты выживаемости (`sur`) разновозрастных возрастных групп (коэффициенты ежегодного сохранения площадей). Умножение матрицы перехода (`se`) на вектор с площадями разновозрастных групп (`n`) в момент времени i формирует вектор с площадями разновозрастных групп в следующий момент времени $i+1$. Объем рубок (`cut`) задается как дополнительная гибель и включается в матрицу перехода. Смысл моделирования состоит в том, чтобы подобрать такую величину площади рубок (`cut4`), чтобы ее можно было вырубать в каждый год прогнозного периода до заданного уровня истощения требуемых лесных ресурсов к концу этого периода. В программе задан 0 доступной площади для рубки на последнем шаге модели, т. е. полное истощение требуемых лесных ресурсов, но эта величина может быть и иной.

Программа имеет четыре блока. Первый блок задает стартовые условия работы модели – период прогноза (`time`), примерные *заниженные* объемы рубок (`cut4`), вектор выживания (`sur`), вектор первичной возрастной структуры (`n`). Функция `fper` рассчитывает весь массив возрастной структуры в каждый момент времени, используя исходный возрастной состав `n` и матрицу перехода `se`, и возвращает величину отклонения значения площади спелых лесов, сохранившихся на последнем шаге прогноза, от нуля (`0-nper[5,time]`) – это и есть функция невязки. Функция `fcut` подготавливает матрицу перехода `se`, используя заданные векторы выживаемости (`sur`) и объема рубок (`cut`), после чего вызывает функцию `fper`. Третий блок – вызов процедуры оптимизации `nlm`; процедура стремится свести к нулю функцию невязки (`0-nper[5,time]`) путем перебора значений настраиваемого параметра (`cut4`) – в заключении он и выводится на печать, это допустимый объем рубок (`rez$estimate`) – относительная площадь расчетной лесосеки (измененное значение).

Среда R была необходима для разработки эффективного алгоритма расчетов, однако удобный интерфейс для широкого круга пользователей лучше создавать в среде JavaScript.

```

time = 100 ; cut4 = 0.3
fcut <- function(cut4,time)
{
nrep = matrix(0,5,time) ; n = c(.05,.25,.34,.17,.2)
cut = c(0,0,0,cut4,0) ; sur = c(1,1,1,1,0)
s = sur-cut ; se = diag(s) ; d = 1-s ; se = rbind(d,se) ; se = se[1:5,]
frep(n,nrep,se,time,cut4)
}
frep <- function(n,nrep,se,time,cut4)
{
for (i in 1:time)
{
nrep[,i] = n ; n1 = n ; n = se%*%n ; nd = n1-n ; n[1] = n[1]+sum(nd)}
0-nrep[5,time]
}
rez <- nlm(fcut,cut4,time)
rez$estimate

```

Рисунок 2.6 – Листинг модели на языке R для популяции с пятью разновозрастными группами

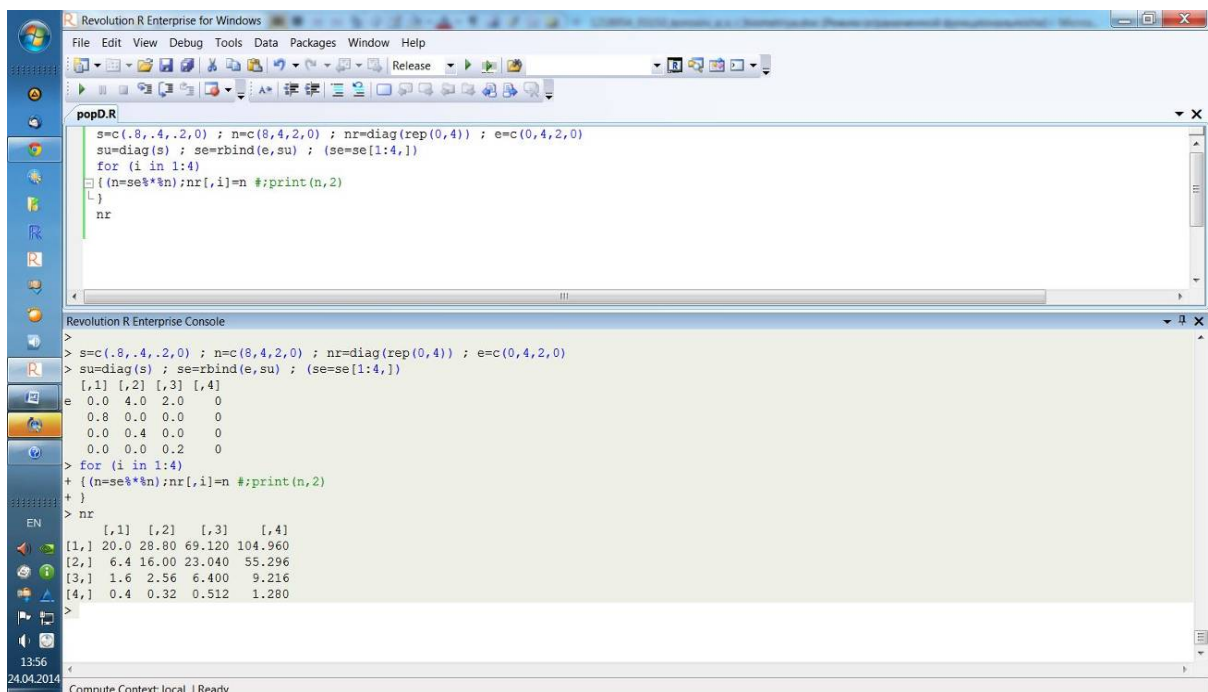


Рисунок 2.7 – Общий вид рабочего стола программной среды R

Среда JavaScript. Для практического применения полученных математических моделей широким кругом пользователей с использованием сети Интернет, была разработана программа на языке JavaScript [14].

Программная реализация модели на языке JavaScript в целом копирует структуру модели, созданной в средах Excel и R. При отладке программы на JavaScript использовались результаты моделирования в средах Excel и R.

Достоинством среды JavaScript является возможность создания интерфейса для ввода-вывода данных, удобного для пользователя. Для нас также важно, что она открывает возможность работать на ПЭВМ с любой операционной системой, в среде web-браузера, в котором включена обработка javascript-сценариев (по умолчанию она включена).

Разработанная программа на JavaScript размещена в тестовом режиме в сети Интернет по адресу: <http://forest-karelia.ru/lesoseka/> и может быть перенесена на другой адрес. Программа работает во всех современных web-браузерах (Firefox, Opera, Chrome, Safari, Yandex, Internet Explorer).

Листинг программы на языке JavaScript представлен в Приложении 1.

Разработанная на языке JavaScript программа предназначена для вычисления параметров «расчетной лесосеки» в спелых и перестойных лесах, обеспечивающей непрерывное неистощительное лесопользование в эксплуатационных лесах. При вычислении используются исходные данные из лесохозяйственных регламентов центральных лесничеств: площади древостоев различных классов возраста, запас на 1 га в спелых и перестойных древостоях.

Запас древесины на 1 га для спелых и перестойных древостоев в указанной программе определяется по данным лесохозяйственных регламентов центральных лесничеств. Математическое моделирование хода роста и изменения запаса соответствующих насаждений (см. подраздел 2.3) здесь не применяется, в соответствии с предназначением программы.

Описание интерфейса программы – см. раздел 3.

Программа позволяет достаточно точно моделировать изменения соотношения площадей возрастных групп насаждения монопородного состава и вычислять на этой основе объемы или площади изъятия лесных ресурсов (параметры «расчетной лесосеки») при сплошных рубках, обеспечивающие непрерывное неистощительное лесопользование в спелых эксплуатационных лесах на протяжении всего периода прогноза. Программа обеспечивает (хотя и менее точно) аналогичное моделирование и вычисление для постепенных и выборочных рубок в спелых эксплуатационных лесах.

Для моделирования постепенных (выборочных) рубок исчисление требуемой площади «расчетной лесосеки» производится путем деления общего запаса древесины, намеченного к изъятию при рубках в соответствующем насаждении, на соответствующий средний запас древесины на гектаре в спелых и перестойных древостоях. Если известна только площадь постепенных (выборочных) рубок, то вычисление общего запаса, намеченного к изъятию, производится путем умножения этой площади на средний запас древесины на гектаре, с учетом среднего процента выборки от общего запаса (также приводится в лесохозяйственных регламентах).

2.6. Возможные направления дальнейших исследований

Опыт разработки принципов, алгоритмов и программ ПЭВМ для вычисления параметров «расчетной лесосеки», обеспечивающей непрерывное неистощительное лесопользование в эксплуатационных лесах, подсказывает следующие направления дальнейших исследований:

1. Следует усовершенствовать уже разработанную программу моделирования динамики древостоя и вычисления параметров «расчетной лесосеки»:

a. ввести в текст программы на языке JavaScript вычислительные блоки на языке R – это упростит структуру программы и ускорит ее работу (сейчас все процедуры оформлены как циклы);

b. разработать полностью автономную версию программы, позволяющую запускать и выполнять ее на ПЭВМ без подключения к сети Интернет (сейчас реализация таких возможностей, как экспорт результатов в файл, может потребовать обращения к удаленному серверу);

2. Следует детализировать уже разработанную модель:

a. учесть в модели все типы рубок – сплошные, постепенные, выборочные и рубки ухода, а также связанные с ними эффекты изменения продуктивности, породного состава и запаса насаждения (сейчас учитываются только рубки сплошные, менее точно – постепенные и выборочные);

b. увеличить перечень пород деревьев (и соответствующих им баз данных с характеристиками), для которых может быть выполнена оценка расчетной лесосеки (сейчас охвачено 4 породы);

c. учесть в модели эффекты «накопления» запасов спелых и перестойных насаждений при эксплуатации насаждений, ранее истощенных рубками (т. е. имеющих на начальном этапе моделирования дефицит спелых и приспевающих древостоев, при избытке молодняков);

3. Выйти на моделирование территориального аспекта лесозаготовок:

а. с использованием ГИС-технологий построить модель пространственно-временной динамики насаждений для целей распределения площадей рубок по территории, проектирования сети дорог и других необходимых лесохозяйственных мероприятий;

б. с использованием ГИС-технологий построить модель распределения площадей рубок по территории с учетом транспортной и экономической доступности лесов, а также других природных и производственных факторов (в настоящее время пространственный фактор не учитывается);

с. создать программу автоматического проектирования (оптимизации) сети лесных дорог на основе прогнозируемых объемов (площадей) изъятия лесных ресурсов в долгосрочной перспективе (результаты подобных разработок уже опубликованы – см. работы [6, 7, 15]).

3. ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ВЫЧИСЛЕНИЯ ПАРАМЕТРОВ РАСЧЕТНОЙ ЛЕСОСЕКИ НЕИСТОЩИТЕЛЬНОГО ПОЛЬЗОВАНИЯ

3.1. Общие сведения о программе

Разработанная на основе предлагаемой методики (см. раздел 2) программа на языке JavaScript предназначена для вычисления параметров «расчетной лесосеки» в спелых и перестойных лесах, обеспечивающей непрерывное неистощительное лесопользование в эксплуатационных лесах. При вычислении используются данные из лесохозяйственных регламентов центральных лесничеств: площади древостоев различных классов возраста, запас на 1 га в спелых и перестойных древостоях.

Запас древесины на 1 га для спелых и перестойных древостоев в указанной программе определяется по данным лесохозяйственных регламентов центральных лесничеств. Математическое моделирование хода роста и изменения запаса соответствующих насаждений (см. подраздел 2.3) здесь не применяется, в соответствии с предназначением программы.

Программа позволяет достаточно точно моделировать изменения соотношения площадей возрастных групп насаждения монопородного состава и вычислять на этой основе объемы или площади изъятия лесных ресурсов (параметры «расчетной лесосеки») при сплошных рубках, обеспечивающие непрерывное неистощительное лесопользование в спелых эксплуатационных лесах на протяжении всего периода прогноза. Программа обеспечивает (хотя и менее точно) аналогичное моделирование и вычисление для постепенных и выборочных рубок в спелых эксплуатационных лесах.

Для моделирования постепенных (выборочных) рубок исчисление требуемой площади «расчетной лесосеки» производится путем деления общего запаса древесины, намеченного к изъятию при рубках в соответствующем насаждении, на соответствующий средний запас древесины на гектаре в спелых и перестойных древостоях. Если известна только площадь постепенных (выборочных) рубок, то вычисление общего запаса, намеченного к изъятию, производится путем умножения этой площади на средний запас древесины на гектаре, с учетом среднего процента выборки от общего запаса (также приводится в лесохозяйственных регламентах).

Разработанная программа на JavaScript размещена в тестовом режиме в сети Интернет по адресу: <http://forest-karelia.ru/lesoseka/> и может быть перенесена на другой адрес. Программа работает во всех современных web-браузерах (Firefox, Opera, Chrome, Safari, Yandex, Internet Explorer) и может быть использована в практической деятельности специалистами лесного сектора.

Листинг программы на языке JavaScript представлен в Приложении 1.

Для работы с программой требуется любой достаточно современный компьютер с предустановленной операционной системой и web-браузером (Firefox, Opera, Chrome, Safari, Yandex, Internet Explorer), в котором включена обработка javascript-сценариев (по умолчанию она включена), с подключением к сети Интернет. Минимальные требования к конфигурации ПЭВМ – IBM-совместимый персональный компьютер с характеристиками: процессор не ниже Pentium 2, RAM 32 Mb, HDD 2 Gb, монитор с разрешающей способностью 1280x1024.

Для удобства обработки результатов и оформления отчетов по результатам использования программы желательно иметь на этом же компьютере пакет офисных программ типа Microsoft Office.

3.2. Описание интерфейса пользователя программы

Интерфейс пользователя разработанной программы на языке JavaScript является графическим, т. е. элементы интерфейса (меню, кнопки, значки, списки и т. п.), представленные пользователю на дисплее ПЭВМ, исполнены в виде графических изображений. Язык интерфейса – русский.

Отдельные элементы интерфейса снабжены текстовыми поясняющими подсказками.

Предполагается, что пользователь имеет подготовку в области лесного хозяйства и ознакомился с описанием методики (см. раздел 2) для самостоятельного использования программы.

Главное окно программы (см. рисунок 3.1) появляется на дисплее ПЭВМ при вводе в адресную строку web-браузера адреса: <http://forest-karelia.ru/lesoseka/>.

1. Ввод исходных данных

Перед началом расчетов обязательно требуется ввести следующие исходные данные из лесохозяйственных регламентов (или других аналогичных источников):

- площади древостоев различных групп возраста, га;
- запас на 1 га в спелых и перестойных древостоях;
- возраст начала рубки в спелых и перестойных древостоях.

Для ввода площади древостоя определенной группы возраста (начиная с самой младшей – «молодняков») требуется нажать на кнопку «Добавить группу возраста» (см. рисунок 3.2). В появившихся полях требуется ввести: нижний и верхний пределы группы возраста, лет; площадь группы, га; примечание (например – название группы).

Нажатие на кнопку <X> (см. рисунок 3.2) позволяет удалить введенную группу

возраста.

При вводе исходных данных программа автоматически подсчитывает и выводит на дисплей ПЭВМ следующую информацию (см. рисунок 3.2):

- общую площадь насаждения, га;
- эксплуатационную площадь (т. е. площадь спелых и перестойных древостоев), га;
- общий запас древесины на эксплуатационной площади, кубм.

При желании, пользователь может ввести т. н. «долю возможных потерь площадей» (см. рисунок 3.2), т. е. долю возможных потерь покрытых лесом площадей от негативных природных или антропогенных воздействий (пожары, вредители, болезни леса и др.). По умолчанию установлено, что ежегодно может гибнуть 0,01 часть площадей каждого возраста (в примере – от 0 до 100 лет).

Пользователь может обнулить эту величину, либо ввести для каждого года какое-то свое значение, на основании имеющейся у него информации – см. рисунок 3.3.

2. Выбор режима расчета

Программа может работать в двух режимах (см. рисунок 3.4):

- «прогноз продолжительности пользования» – прогноз продолжительности пользования спелыми и перестойными древостоями при заранее известном ежегодном объеме их рубки;
- «расчет объема пользования» – расчет объема равномерного пользования спелыми и перестойными древостоями на период прогноза.

При выборе режима «прогноз продолжительности пользования» (см. рисунок 3.5) необходимо дополнительно заполнить следующие поля интерфейса программы:

- «Планируемый объем рубки леса, м³»;
- «Продолжительность прогноза, лет».

Для получения результатов расчета в режиме «прогноз продолжительности пользования» следует нажать на кнопку «рассчитать прогноз» (см. рисунок 3.5).

При выборе режима «расчет объема пользования» (см. рисунок 3.6) необходимо дополнительно заполнить следующее поле интерфейса программы:

- «Количество лет пользования».

Для получения результатов расчета в режиме «расчет объема пользования» следует нажать на кнопку «рассчитать объем пользования» (см. рисунок 3.6).

3. Вывод результатов расчетов

Результаты расчетов в режиме «прогноз продолжительности пользования» выводятся в виде таблицы и графика динамики долей от общей площади всех возрастных групп (см. рисунок 3.7). Продолжительность пользования лесом при

заранее известном ежегодном объеме их рубки указывается на графике. В примере на рисунке 3.7 продолжительность составляет 67 лет.

Результаты расчетов в режиме «расчет объема пользования» выводятся в виде таблицы и графика динамики долей от общей площади всех возрастных групп (см. рисунок 3.8). Также указывается ежегодный расчетный объем равномерного пользования лесом (т. е. рубки леса) на период прогноза. В примере на рисунке 3.8 этот объем составляет 1562,9 м³ на период 100 лет.

Добавить группу возраста

От	До	Площадь, га	Примечание
Общая площадь, га: 0			
Эксплуатационная площадь, га: 0			
Доля возможных потерь площадей (по умолчанию)			0.01

Шаг моделирования возраста, год

Доля возможных потерь площадей

Доля от общей площади

Возраст начала рубки древостоев, лет 81

Запас древесины на 1 га, м³ 200

Общий запас древесины на эксплуатационной площади, м³: 0

выберите режим модели

Рисунок 3.1 – Главное окно программы

Добавить группу возраста

От	До	Площадь, га	Примечание
0	40	100	моподняки
41	60	200	средневозрастные
61	80	300	приспевающие
81	100	400	спелые и перестойные

Общая площадь, га: 1000

Эксплуатационная площадь, га: 400

Доля возможных потерь площадей (по умолчанию) 0.01

Шаг моделирования возраста, год

	в0	в1	в2	в3	в4	в5	в6	в7	в8	в9	в10	в11	в12	в13	в14	в15	в16	в17	в18	в19	в20
Доля возможных потерь площадей	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Доля от общей площади	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002

Возраст начала рубки древостоев, лет 81

Запас древесины на 1 га, м³ 200

Общий запас древесины на эксплуатационной площади, м³: 80000

выберите режим модели

Рисунок 3.2 – Ввод основных исходных данных

Добавить группу возраста

От	До	Площадь, га	Примечание
0	40	100	молодняки
41	60	200	средневозрастные
61	80	300	приспевающие
81	100	400	спелые и перестойные

Общая площадь, га: 1000
 Эксплуатационная площадь, га: 400
 Доля возможных потерь площадей (по умолчанию) 0.01

Шаг моделирования возраста, год	в0	в1	в2	в3	в4	в5	в6	в7	в8	в9	в10	в11	в12	в13	в14	в15	в16	в17	в18	в19	в20
Доля возможных потерь площадей	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Доля от общей площади	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002

Возраст начала рубки древостоев, лет 81
 Запас древесины на 1 га, м³ 200
 Общий запас древесины на эксплуатационной площади, м³: 80000
 прогноз продолжительности пользования
 Планируемый объем рубки леса, м³ 0
 Продолжительность прогноза, лет 100
 рассчитать прогноз

Рисунок 3.5 – Режим расчета «прогноз продолжительности пользования»

Добавить группу возраста

От	До	Площадь, га	Примечание
0	40	100	молодняки
41	60	200	средневозрастные
61	80	300	приспевающие
81	100	400	спелые и перестойные

Общая площадь, га: 1000
 Эксплуатационная площадь, га: 400
 Доля возможных потерь площадей (по умолчанию) 0.01

Шаг моделирования возраста, год	в0	в1	в2	в3	в4	в5	в6	в7	в8	в9	в10	в11	в12	в13	в14	в15	в16	в17	в18	в19	в20
Доля возможных потерь площадей	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	
Доля от общей площади	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	

Возраст начала рубки древостоев, лет 81
 Запас древесины на 1 га, м³ 200
 Общий запас древесины на эксплуатационной площади, м³: 80000
 расчет объема пользования
 Количество лет пользования 100
 рассчитать объем пользования
 Расчетный объем пользования, м³:

Рисунок 3.6 – Режим расчета «расчет объема пользования»

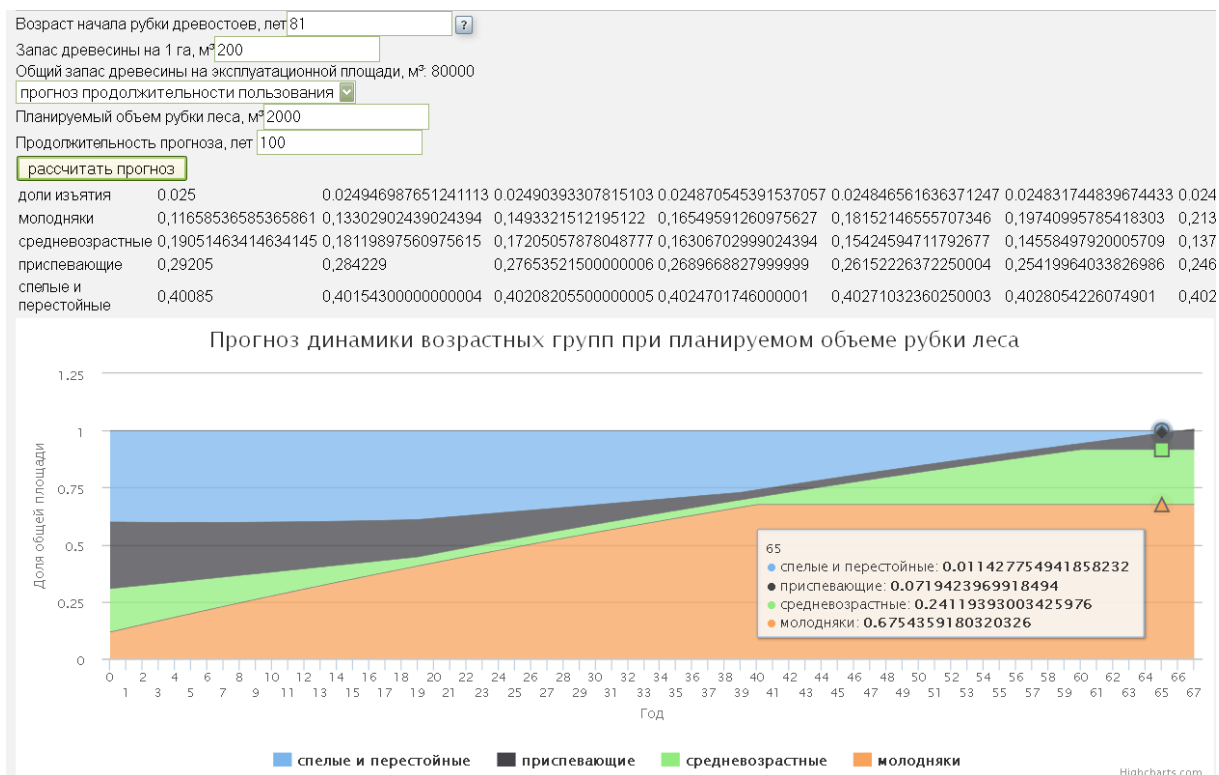


Рисунок 3.7 – Результаты расчетов в режиме «прогноз продолжительности пользования»

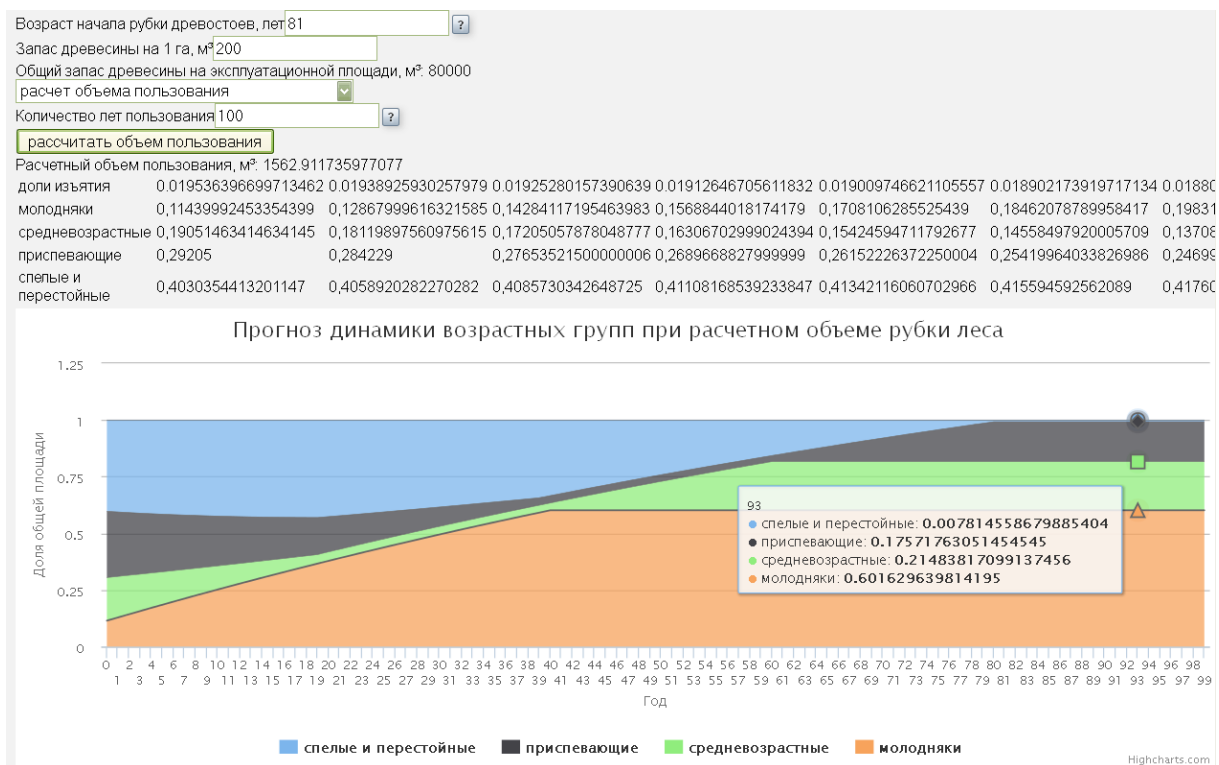


Рисунок 3.8 – Результаты расчетов в режиме «расчет объема пользования»

4. СРАВНИТЕЛЬНЫЙ АНАЛИЗ ДИНАМИКИ ИСПОЛЬЗОВАНИЯ ЛЕСОВ ДЛЯ МОДЕЛЬНЫХ ТЕРРИТОРИЙ

4.1. Общие сведения

Сравнительный анализ динамики использования лесов проведем для четырех модельных территорий Республики Карелия (см. рисунок 4.1): Беломорское центральное лесничество, Сегежское центральное лесничество, Медвежьегорское центральное лесничество и Пудожское центральное лесничество; по действующей [10] и предлагаемой методикам (см. раздел 2), на период 100 лет.

Расчет проведем по материалам лесохозяйственных регламентов [8], где в т. ч. собраны сведения о структуре эксплуатационных насаждений и исчисленных расчетных лесосеках для сплошных рубок спелых и перестойных насаждений в эксплуатационных лесах указанных центральных лесничеств, без учета транспортной и/или экономической доступности лесных ресурсов.

Рубки в защитных лесах (защитные полосы вдоль дорог, нерестоохраняемые полосы, особо охраняемые природные территории и т. п.) исключаем из настоящего анализа.

Данное допущение основано на том, что, по нашему мнению, эксплуатационные леса должны в первую очередь гарантировать устойчивое обеспечение перерабатывающих предприятий лесного сектора необходимым сырьем. Задачи сохранения биологического разнообразия, обеспечения водоохраных, защитных и др. полезных свойств лесов в условиях лесничеств должны выполняться преимущественно за счет защитных лесов. Вопрос о необходимой площади таких лесов на территории соответствующих лесничеств является весьма актуальным, но выходит за рамки анализа.

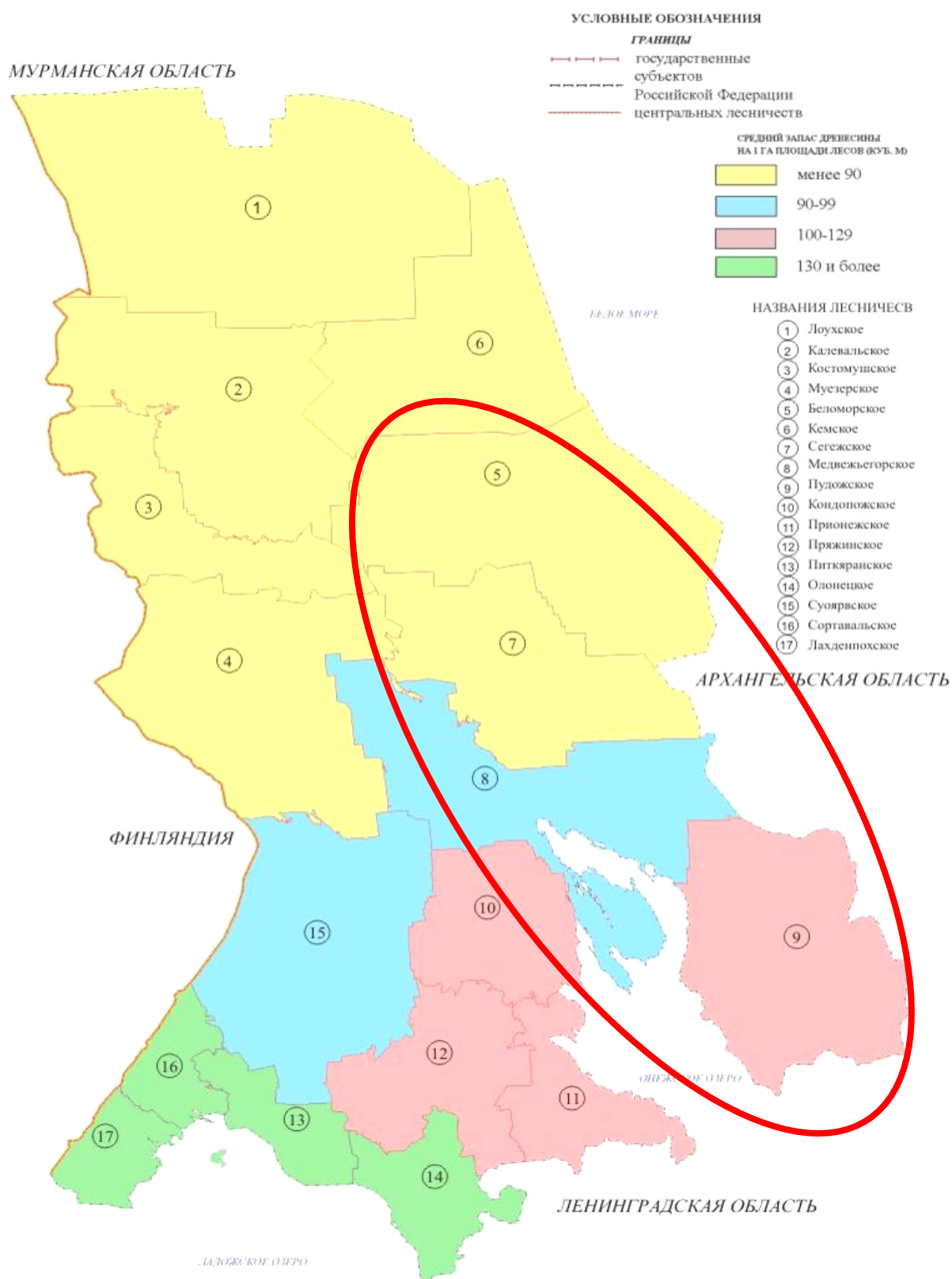


Рисунок 4.1 – Фрагмент карты-схемы центральных лесничеств
Республики Карелия [16]

4.2. Беломорское центральное лесничество

Таблица 4.1 – Сведения о структуре эксплуатационных насаждений, га
Беломорское центральное лесничество [8]

Хоз. секция	итого	молод.	средн.	присп.	спел.	перест.	Запас на 1 га, кбм
лет		0-40	41-60	61-80	81-100	101-...	
Сосна, 1-3 бон.	27295	10762	12857	2697	719	260	222
лет		0-40	41-80	81-100	101-120	121-...	
Сосна, 4-5 бон.	354946	239585	70717	7342	7246	30056	136
лет		0-40	41-60	61-80	81-100	101-...	
Ель, 1-3 бон.	147	122	1	9	12	3	247
лет		0-40	41-80	81-100	101-120	121-...	
Ель, 4-5 бон.	49575	18241	11147	3204	3472	13511	143
лет		0-20	21-40	41-60	61-70	71-...	
Береза	44256	6544	34247	1438	862	1165	130
лет		0-20	21-30	31-40	41-50	51-...	
Осина	60	38	0	5	14	3	170

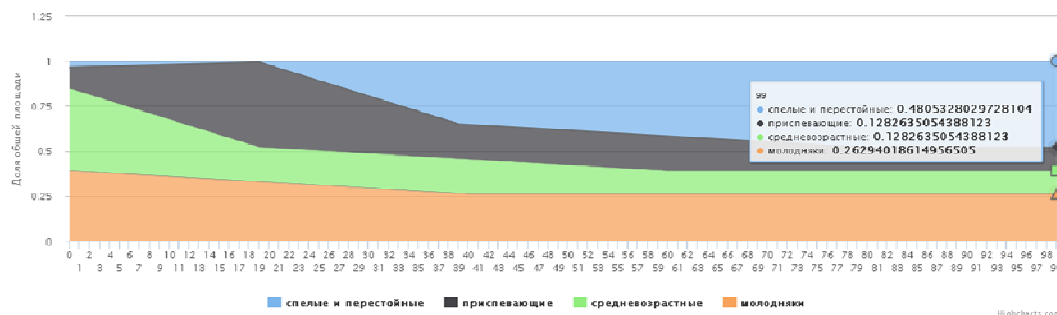
Таблица 4.2 – Исчисленные расчетные лесосеки, га
Беломорское центральное лесничество [8]

Хоз. секция	Равном.	2 возр.	1 возр.	Интегр.	Рекоменд.
Сосна, 1-3 бон.	337	276	92	154	98
Сосна, 4-5 бон.	3514	996	1116	1738	1195
Ель, 1-3 бон.	2	0	1	0	-
Ель, 4-5 бон.	491	400	505	469	442
Береза	726	454	173	509	173
Осина	1	1	1	1	-

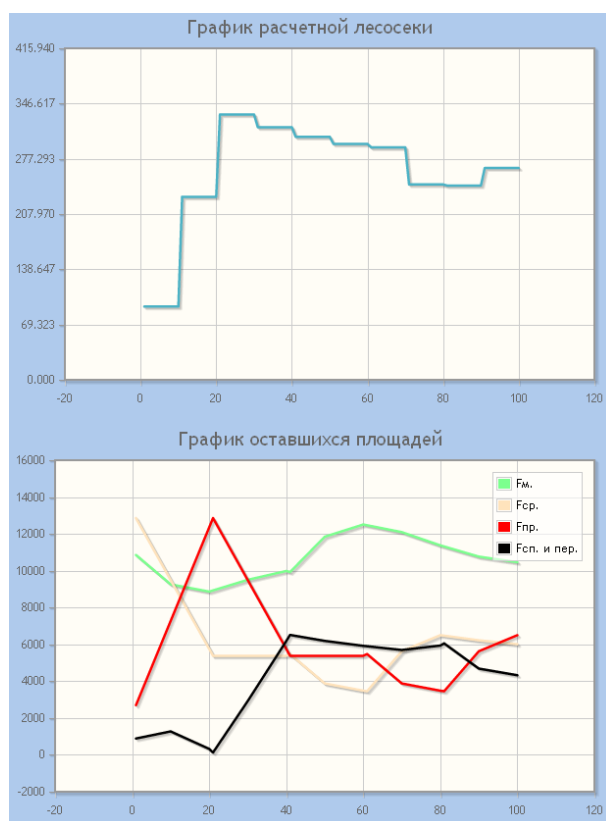
Таблица 4.3 – Существующие и предлагаемые показатели пользования лесом
Беломорское центральное лесничество

Хоз. секция	Возраст рубки, лет	Расчетная лесосека			
		Метод	Площадь, га	Запас, тыс. кубм	Период, лет
Сосна, 1-3 бон.	81*	1 возр.*	92*	20,4*	10*
	81	-	175	38,9	100
Сосна, 4-5 бон.	101*	1 возр.*	1116*	151,8*	10*
	101	-	1891	257,2	100
Ель, 4-5 бон.	101*	1 возр.*	505*	72,2*	10*
	101	-	486	69,6	100
Береза	61*	1 возр.*	173*	22,5*	10*
	61	-	165	21,5	100

* По лесохозяйственному регламенту.

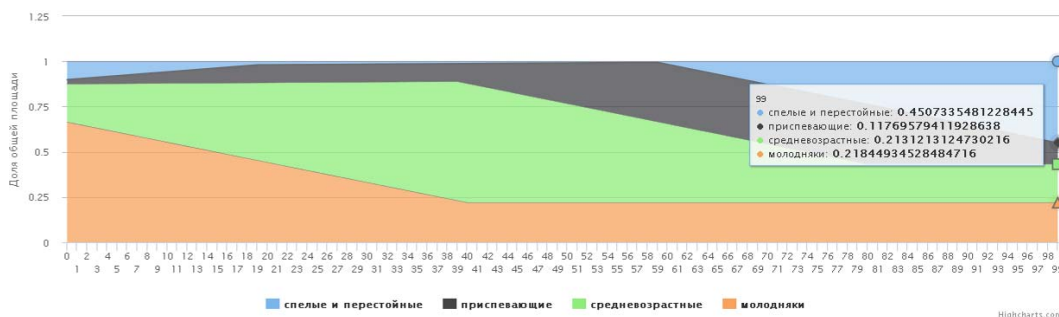


а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 175 га; общая площадь – 27295 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>



б – согласно действующему лесохозяйственному регламенту (1-я возрастная лесосека); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.2 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Сосна, 1-3 бонитет; Беломорское центральное лесничество

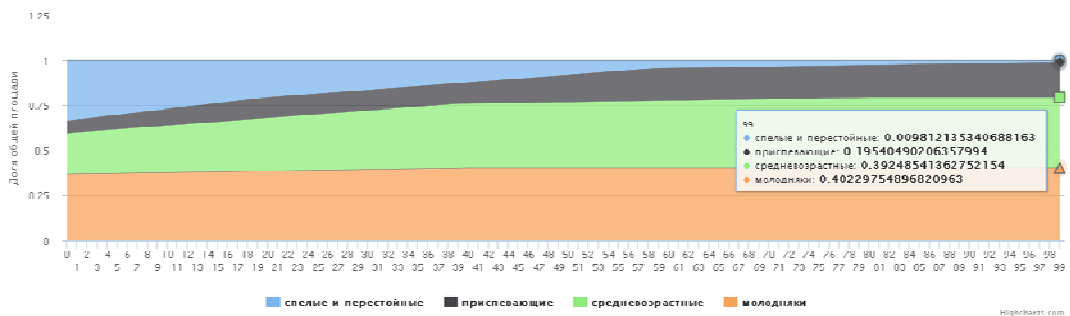


а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 1891 га; общая площадь – 354946 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>

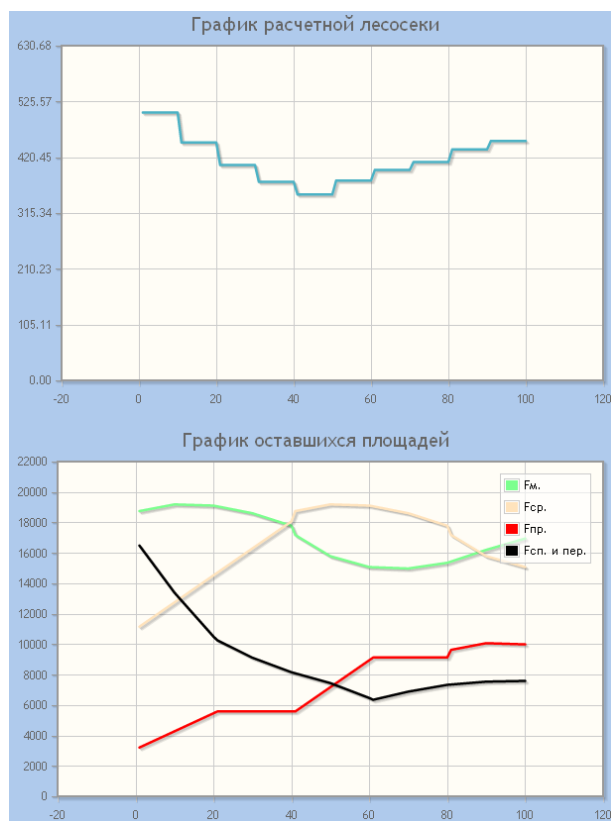


б – согласно действующему лесохозяйственному регламенту (1-я возрастная лесосека); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.3 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Сосна, 4-5 бонитет; Беломорское центральное лесничество

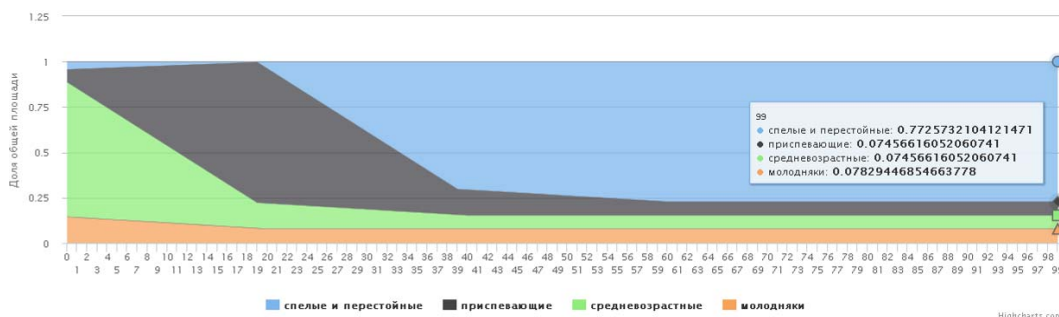


а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 486 га; общая площадь – 49575 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>



б – согласно действующему лесохозяйственному регламенту (1-я возрастная лесосека); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.4 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Ель, 4-5 бонитет; Беломорское центральное лесничество



а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 165 га; общая площадь – 44256 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>



б – согласно действующему лесохозяйственному регламенту (1-я возрастная лесосека); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.5 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Береза, все бонитеты; Беломорское центральное лесничество

4.3. Сегежское центральное лесничество

Таблица 4.4 – Сведения о структуре эксплуатационных насаждений, га
Сегежское центральное лесничество [8]

Хоз. секция	итого	молод.	средн.	присп.	спел.	перест.	Запас на 1 га, кбм
лет		0-40	41-60	61-80	81-100	101-...	
Сосна, 1-3 бон.	50883	13152	28762	7933	883	153	231
лет		0-40	41-80	81-100	101-120	121-...	
Сосна, 4-5 бон.	263476	144167	96064	3005	20240	0	128
лет		0-40	41-60	61-80	81-100	101-...	
Ель, 1-3 бон.	593	433	91	43	20	6	250
лет		0-40	41-80	81-100	101-120	121-...	
Ель, 4-5 бон.	28048	6589	5901	1674	2602	11282	134
лет		0-20	21-40	41-60	61-70	71-...	
Береза	13748	1559	6428	1849	3011	901	162
лет		0-20	21-30	31-40	41-50	51-...	
Осина	5	0	2	0	0	3	300

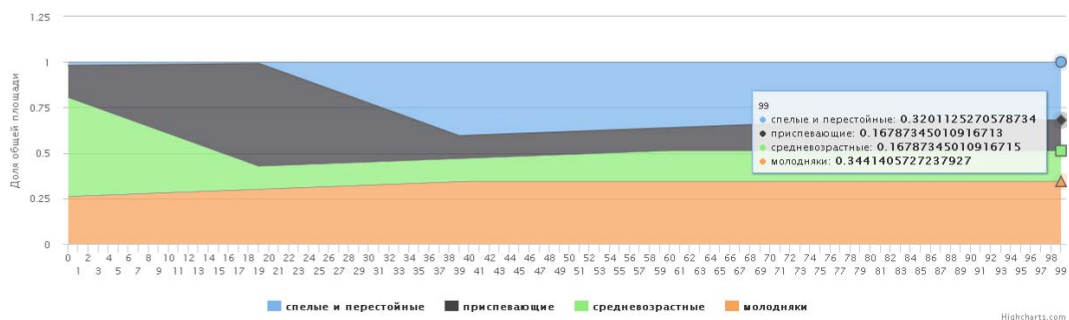
Таблица 4.5 – Исчисленные расчетные лесосеки, га
Сегежское центральное лесничество [8]

Хоз. секция	Равном.	2 возр.	1 возр.	Интегр.	Рекоменд.
Сосна, 1-3 бон.	628	629	224	329	104
Сосна, 4-5 бон.	2608	706	581	1347	706
Ель, 1-3 бон.	7	3	2	4	2
Ель, 4-5 бон.	278	310	389	336	278
Береза	225	320	288	290	165
Осина	0	0	0	0	-

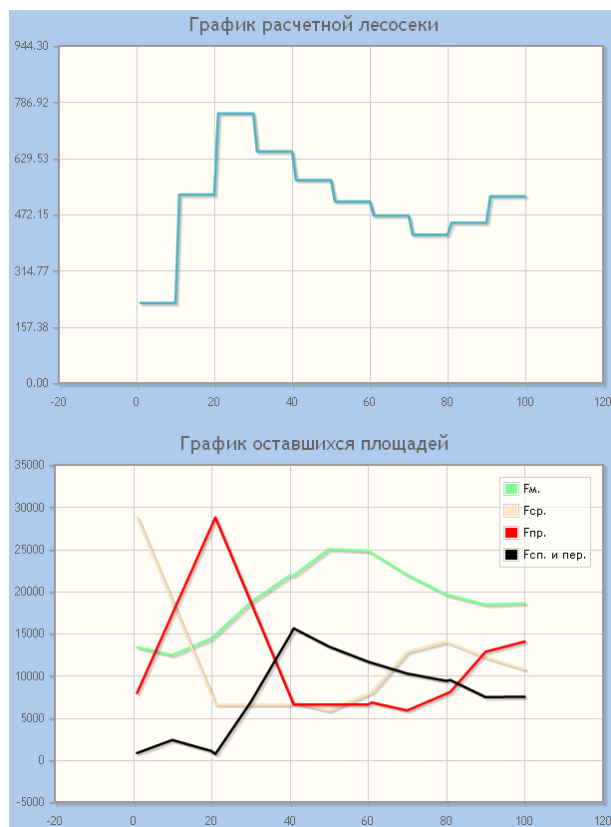
Таблица 4.6 – Возможный ежегодный объем пользования лесом
Сегежское центральное лесничество

Хоз. секция	Возраст рубки, лет	Расчетная лесосека			
		Метод	Площадь, га	Запас, тыс. кубм	Период, лет
Сосна, 1-3 бон.	81*	1 возр.*	224*	51,7*	10*
	81	-	427	98,7	100
Сосна, 4-5 бон.	101*	2 возр.*	706*	90,4*	10*
	101	-	1107	141,7	100
Ель, 4-5 бон.	101*	Равном.*	278*	37,3*	10*
	101	-	276	37,0	100
Береза	61*	Равном.*	225*	36,5*	10*
	61	-	222	35,9	100

* По лесохозяйственному регламенту.

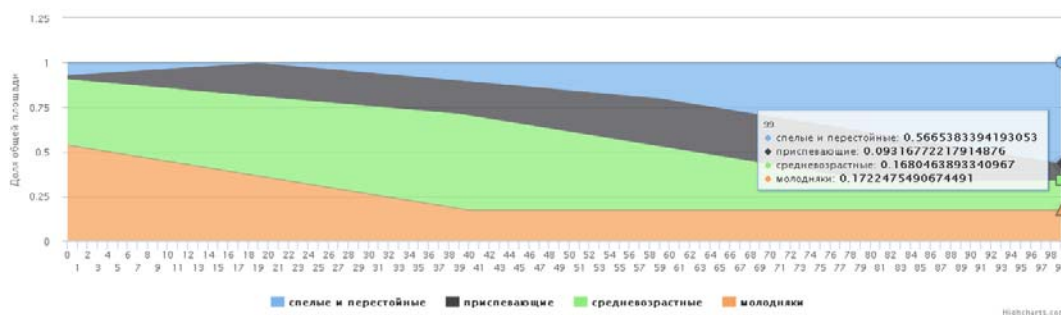


а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 427 га; общая площадь – 50883 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>

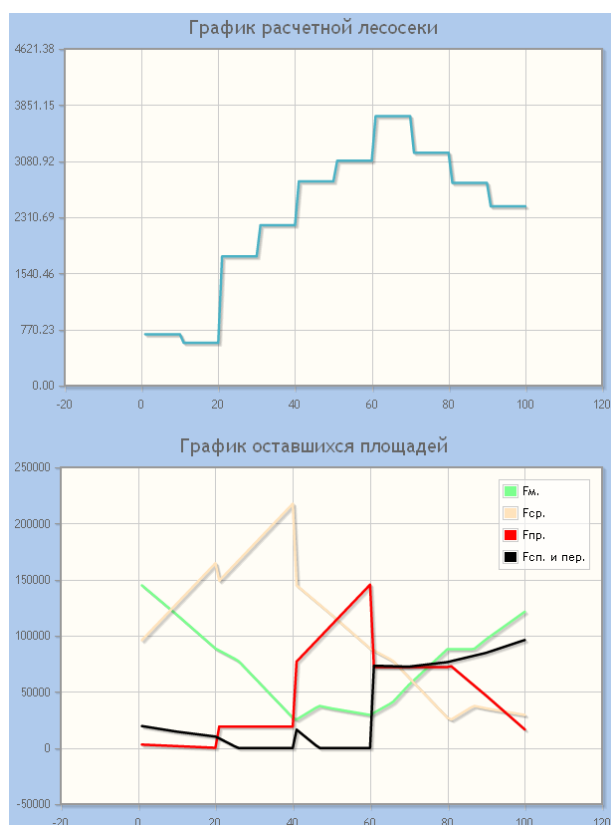


б – согласно действующему лесохозяйственному регламенту (1-я возрастная лесосека); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.6 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Сосна, 1-3 бонитет; Сегежское центральное лесничество

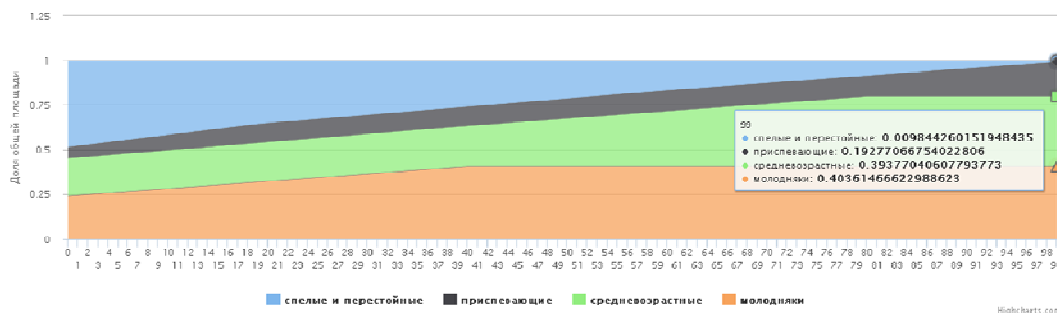


а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 1107 га; общая площадь – 263476 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>

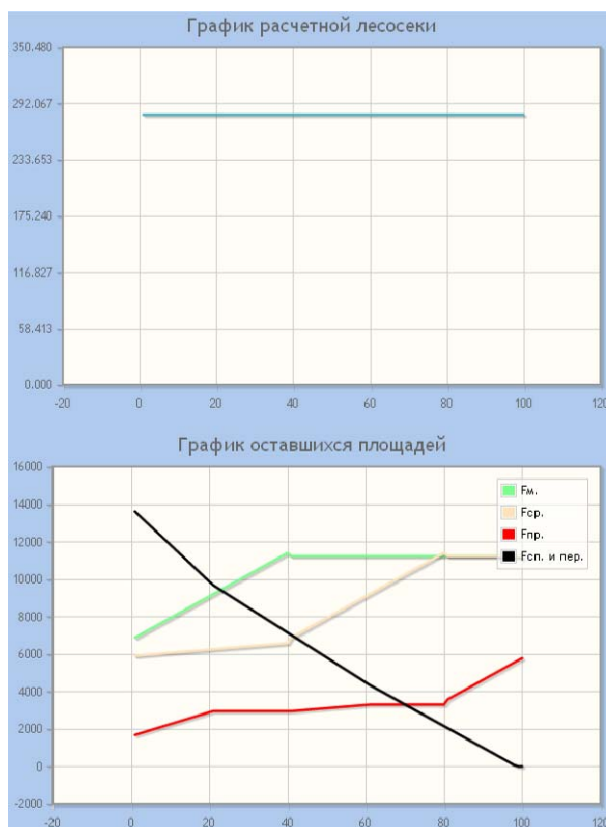


б – согласно действующему лесохозяйственному регламенту (2-я возрастная лесосека); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.7 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Сосна, 4-5 бонитет; Сегежское центральное лесничество

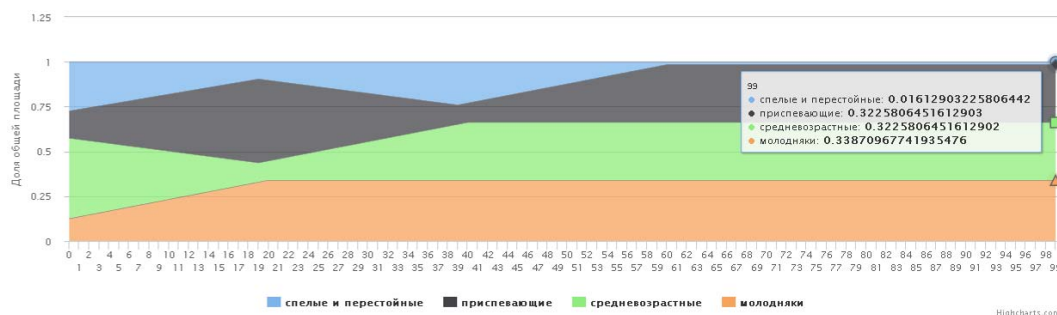


а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 276 га; общая площадь – 28048 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>



б – согласно действующему лесохозяйственному регламенту (лесосека равномерного пользования); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.8 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Ель, 4-5 бонитет; Сегежское центральное лесничество



а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 222 га; общая площадь – 13748 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>



б – согласно действующему лесохозяйственному регламенту (лесосека равномерного пользования); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.9 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Береза, все бонитеты; Сегежское центральное лесничество

4.4. Медвежьегорское центральное лесничество

Таблица 4.7 – Сведения о структуре эксплуатационных насаждений, га
Медвежьегорское центральное лесничество [8]

Хоз. секция	итого	молод.	средн.	присп.	спел.	перест.	Запас на 1 га, кбм
лет		0-40	41-60	61-80	81-100	101-...	
Сосна, 1-3 бон.	123716	43863	34904	35920	8080	949	254
лет		0-40	41-80	81-100	101-120	121-...	
Сосна, 4-5 бон.	236209	106069	80790	7915	4752	36683	154
лет		0-40	41-60	61-80	81-100	101-...	
Ель, 1-3 бон.	17281	4099	1746	3600	6217	1619	257
лет		0-40	41-80	81-100	101-120	121-...	
Ель, 4-5 бон.	134300	56096	29272	8032	15792	25108	173
лет		0-20	21-40	41-60	61-70	71-...	
Береза	74104	8854	30155	9537	16351	9207	166
лет		0-20	21-30	31-40	41-50	51-...	
Осина	7235	592	477	580	3438	2148	146

Таблица 4.8 – Исчисленные расчетные лесосеки, га

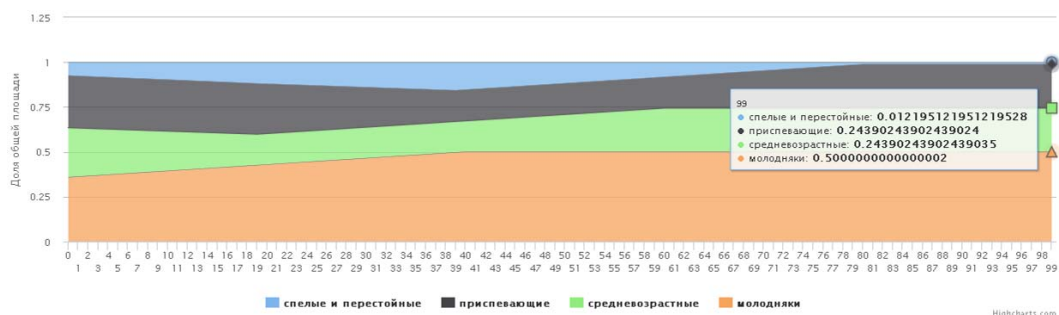
Медвежьегорское центральное лесничество [8]

Хоз. секция	Равном.	2 возр.	1 возр.	Интегр.	Рекоменд.
Сосна, 1-3 бон.	1527	1331	1124	962	903
Сосна, 4-5 бон.	2339	1285	1234	1575	1285
Ель, 1-3 бон.	213	220	286	225	220
Ель, 4-5 бон.	1330	1079	1223	1200	1079
Береза	1215	1515	1755	1552	1215
Осина	176	221	308	225	221

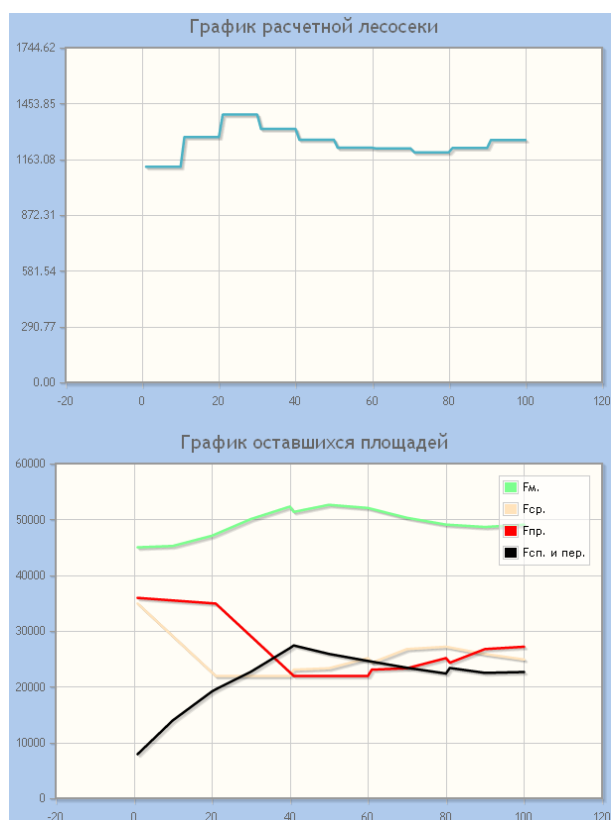
Таблица 4.9 – Возможный ежегодный объем пользования лесом
Медвежьегорское центральное лесничество

Хоз. секция	Возраст рубки, лет	Расчетная лесосека			
		Метод	Площадь, га	Запас, тыс. кубм	Период, лет
Сосна, 1-3 бон.	81*	1 возр.*	1124*	285,5*	10*
	81	-	1509	383,2	100
Сосна, 4-5 бон.	101*	2 возр.*	1285*	197,9*	10*
	101	-	2133	328,6	100
Ель, 1-3 бон.	81*	2 возр.*	220*	56,5*	10*
	81	-	211	54,2	100
Ель, 4-5 бон.	101*	2 возр.*	1079*	186,7*	10*
	101	-	1282	221,8	100
Береза	61*	Равном.*	1215*	201,7*	10*
	61	-	1195	198,4	100
Осина	41*	2 возр.*	221*	32,3*	10*
	41	-	172	25,2	100

* По лесохозяйственному регламенту.

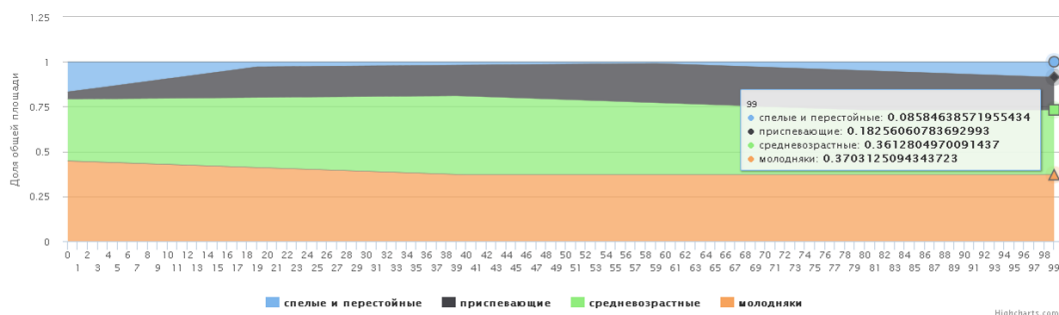


а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 1509 га; общая площадь – 123716 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>



б – согласно действующему лесохозяйственному регламенту (1-я возрастная лесосека); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.10 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Сосна, 1-3 бонитет; Медвежьегорское центральное лесничество



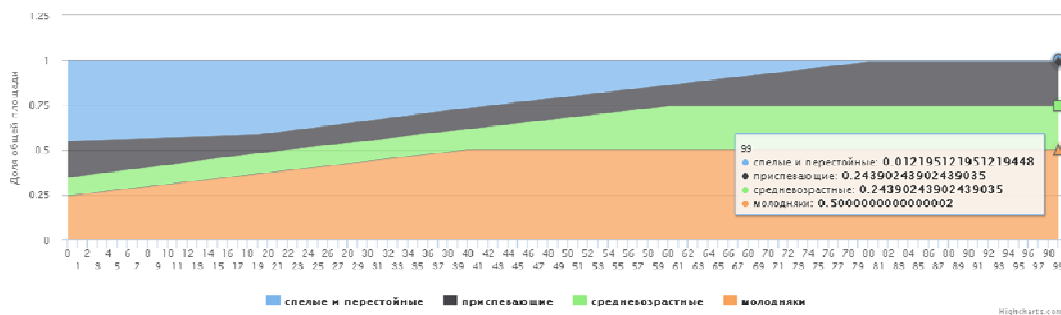
а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 2133 га; общая площадь – 236209 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>



б – согласно действующему лесохозяйственному регламенту (2-я возрастная лесосека); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.11 – Прогноз изменения соотношения площадей различных возрастных групп насаждения

Сосна, 4-5 бонитет; Медвежьегорское центральное лесничество

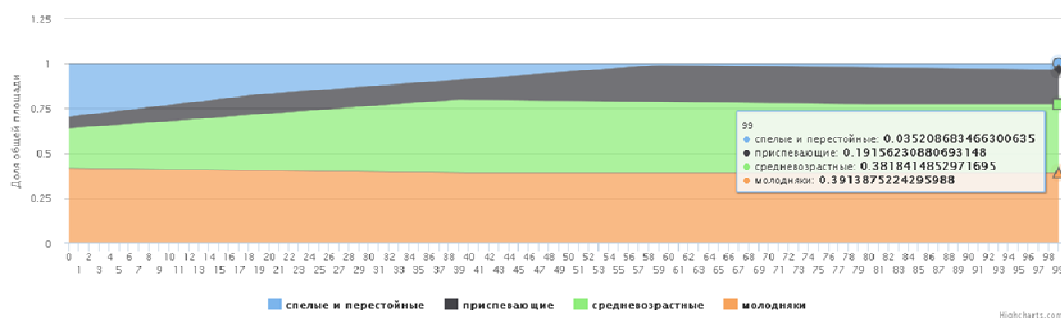


а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 211 га; общая площадь – 17281 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>

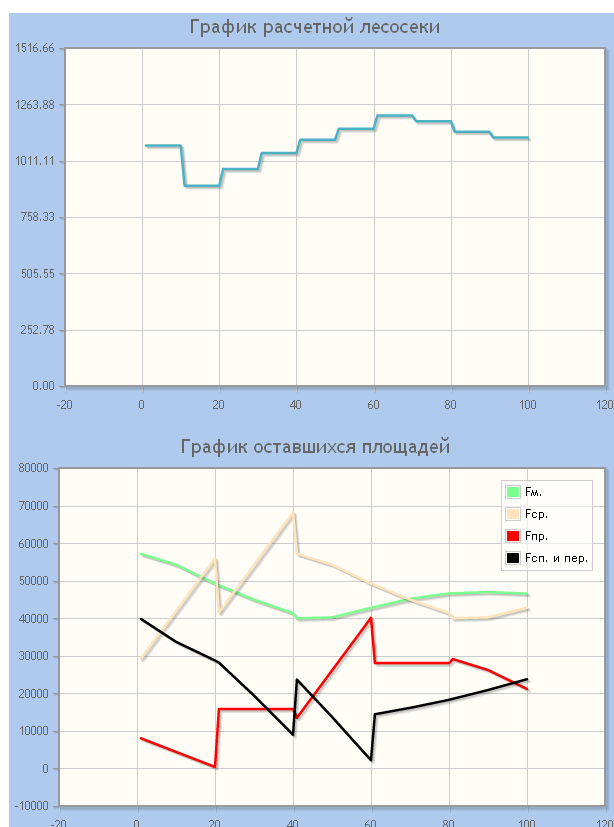


б – согласно действующему лесохозяйственному регламенту (2-я возрастная лесосека); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.12 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Ель, 1-3 бонитет; Медвежьегорское центральное лесничество

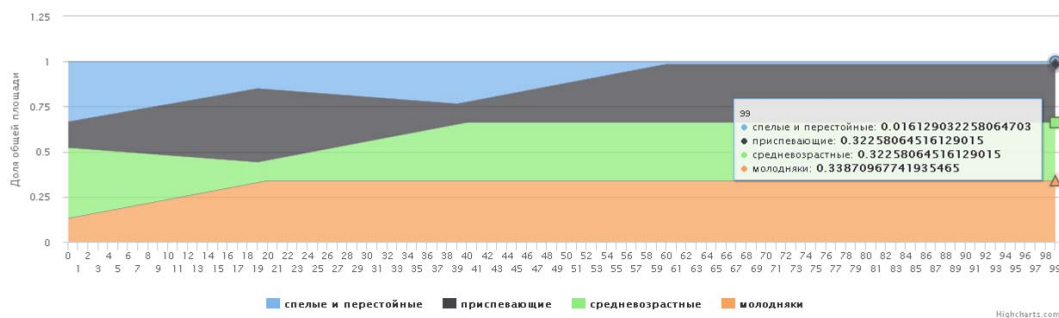


а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 1282 га; общая площадь – 134300 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>



б – согласно действующему лесохозяйственному регламенту (2-я возрастная лесосека); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.13 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Ель, 4-5 бонитет; Медвежьегорское центральное лесничество

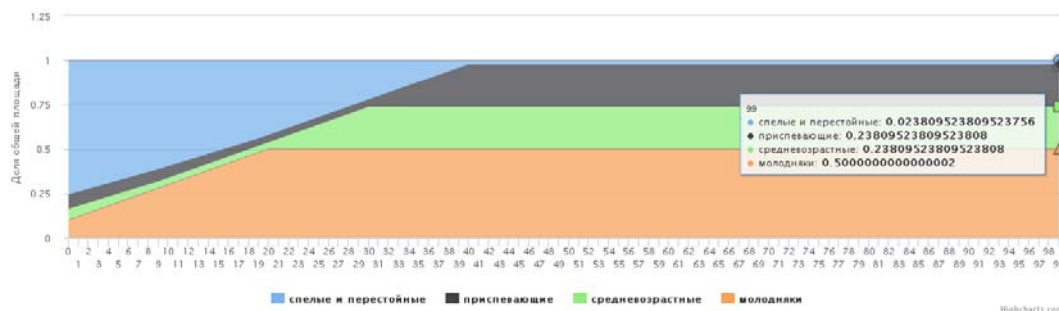


а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 1195 га; общая площадь – 74104 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>

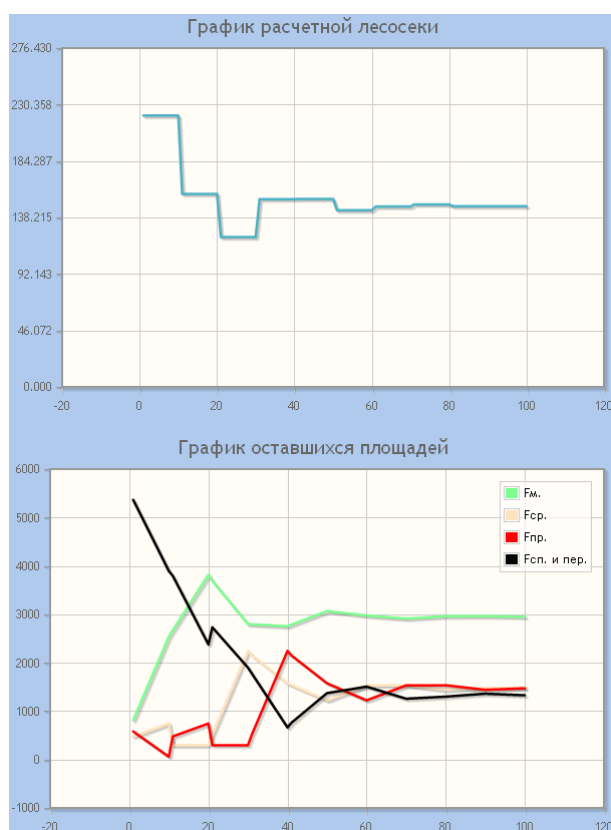


б – согласно действующему лесохозяйственному регламенту (лесосека равномерного пользования); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.14 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Береза, все бонитеты; Медвежьегорское центральное лесничество



а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 172 га; общая площадь – 7235 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>



б – согласно действующему лесохозяйственному регламенту (2-я возрастная лесосека); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.15 – Прогноз изменения соотношения площадей различных возрастных групп насаждения
Осина, все бонитеты; Медвежьегорское центральное лесничество

4.5. Пудожское центральное лесничество

Таблица 4.10 – Сведения о структуре эксплуатационных насаждений, га
Пудожское центральное лесничество [8]

Хоз. секция	итого	молод.	средн.	присп.	спел.	перест.	Запас на 1 га, кбм
лет		0-40	41-60	61-80	81-100	101-...	
Сосна, 1-3 бон.	54575	14752	6559	12435	16862	3967	276
лет		0-40	41-80	81-100	101-120	121-...	
Сосна, 4-5 бон.	96927	32160	12894	7268	7597	37008	137
лет		0-40	41-60	61-80	81-100	101-...	
Ель, 1-3 бон.	70068	43045	5176	5380	10051	6416	267
лет		0-40	41-80	81-100	101-120	121-...	
Ель, 4-5 бон.	280170	147633	33427	10510	21414	67186	174
лет		0-20	21-40	41-60	61-70	71-...	
Береза	140994	40009	74412	9519	8833	8221	189
лет		0-20	21-30	31-40	41-50	51-...	
Осина	7115	1539	963	966	1102	2545	233

Таблица 4.11 – Исчисленные расчетные лесосеки, га
Пудожское центральное лесничество [8]

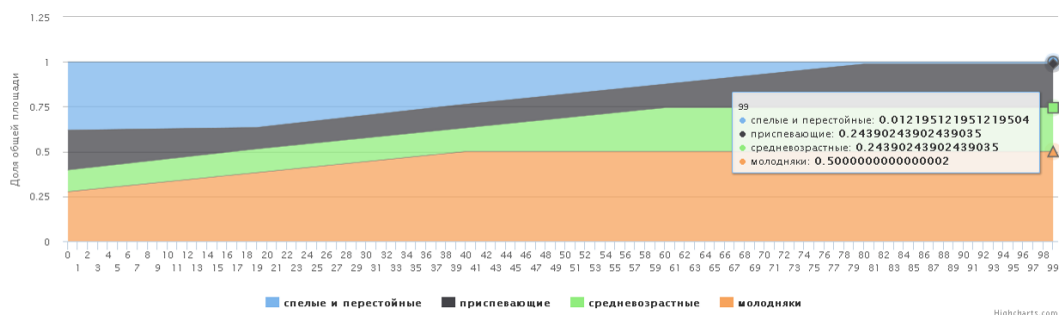
Хоз. секция	Равном.	2 возр.	1 возр.	Интегр.	Рекоменд.
Сосна, 1-3 бон.	674	664	832	618	618
Сосна, 4-5 бон.	960	952	1297	1067	952
Ель, 1-3 бон.	865	450	546	489	489
Ель, 4-5 бон.	2774	1816	2478	2277	1816
Береза	2311	1498	1328	1696	1328
Осина	173	186	231	176	173

Таблица 4.12 – Возможный ежегодный объем пользования лесом

Пудожское центральное лесничество

Хоз. секция	Возраст рубки, лет	Расчетная лесосека			
		Метод	Площадь, га	Запас, тыс. кбм	Период, лет
Сосна, 1-3 бон.	81*	2 возр.*	664*	183,3*	10*
	81	-	666	183,8	100
Сосна, 4-5 бон.	101*	2 возр.*	952*	130,4*	10*
	101	-	952	130,4	100
Ель, 1-3 бон.	81*	2 возр.*	450*	120,2*	10*
	81	-	659	176,0	100
Ель, 4-5 бон.	101*	2 возр.*	1816*	316,0*	10*
	101	-	2172	378,0	100
Береза	61*	1 возр.*	1328*	250,9*	10*
	61	-	1265	239,2	100
Осина	41*	Равном.*	173*	40,3*	10*
	41	-	169	39,5	100

* По лесохозяйственному регламенту.

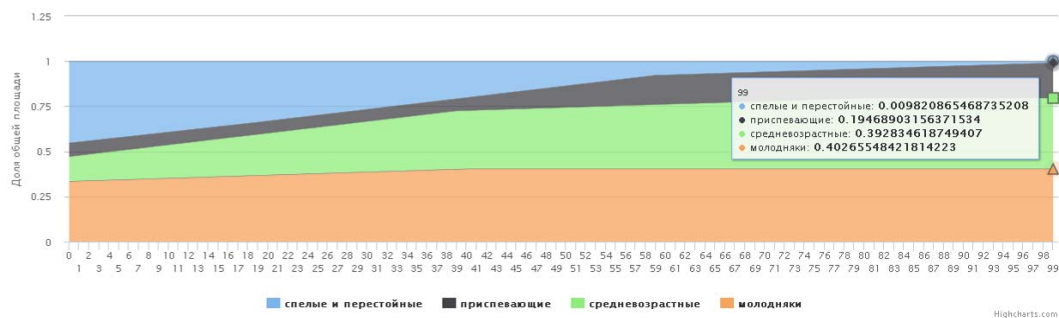


а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 666 га; общая площадь – 54575 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>

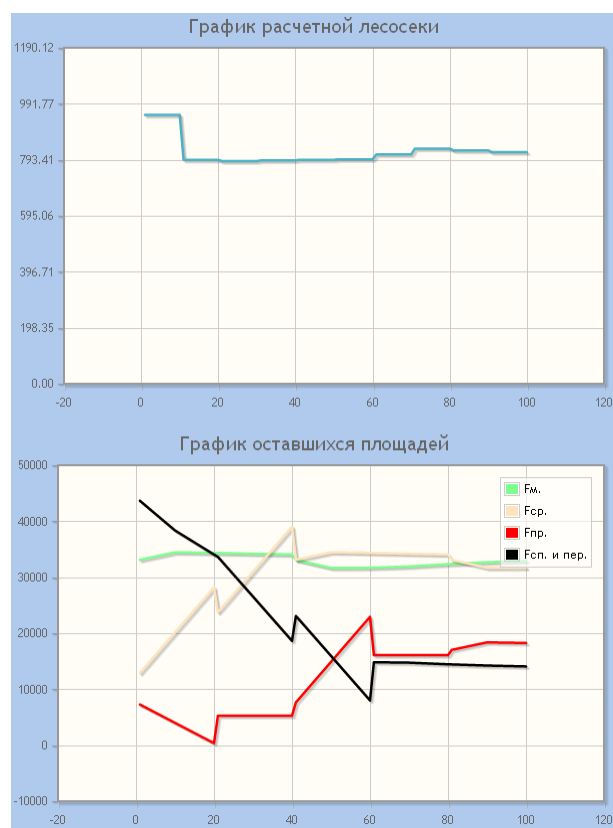


б – согласно действующему лесохозяйственному регламенту (1-я возрастная лесосека); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.16 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Сосна, 1-3 бонитет; Пудожское центральное лесничество

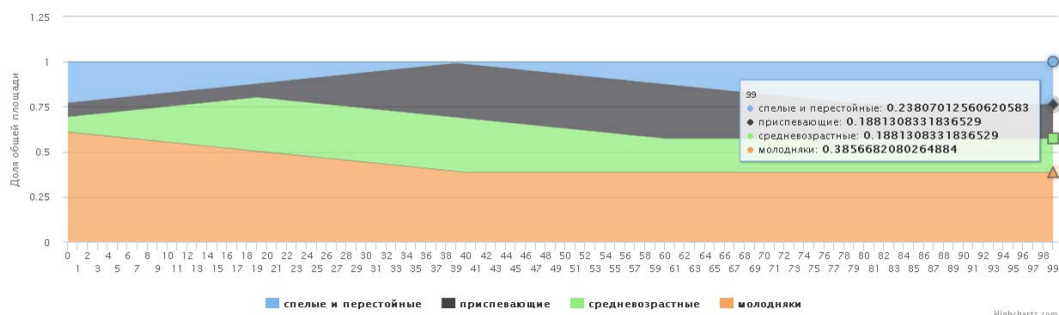


а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 952 га; общая площадь – 96927 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>

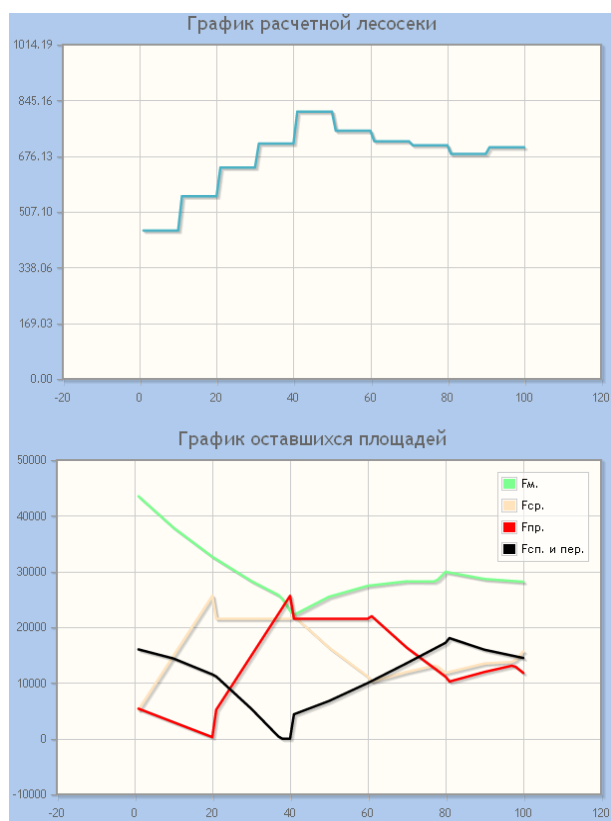


б – согласно действующему лесохозяйственному регламенту (1-я возрастная лесосека); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.17 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Сосна, 4-5 бонитет; Пудожское центральное лесничество

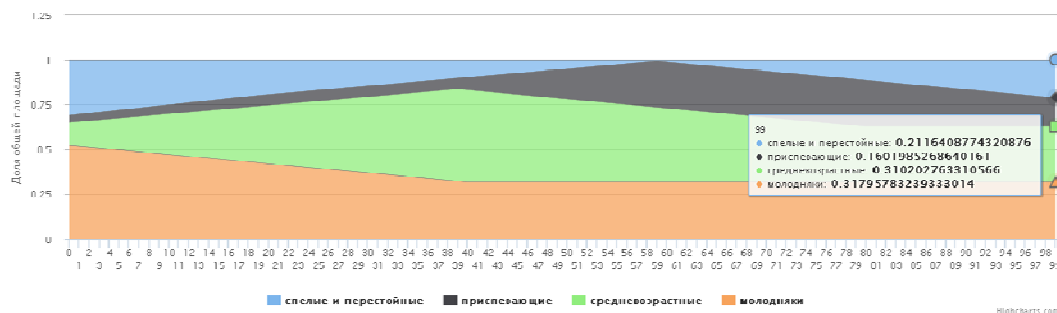


а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 659 га; общая площадь – 70068 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>

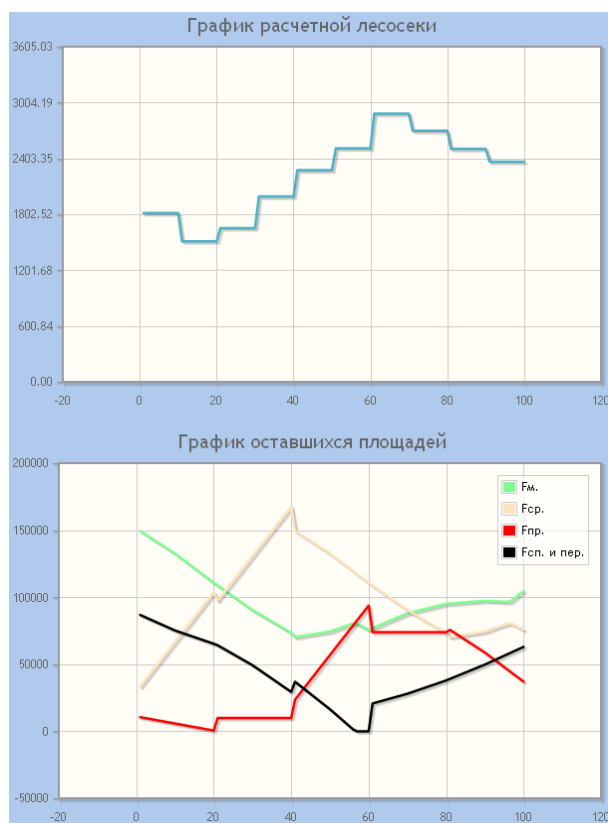


б – согласно действующему лесохозяйственному регламенту (1-я возрастная лесосека); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.18 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Ель, 1-3 бонитет; Пудожское центральное лесничество

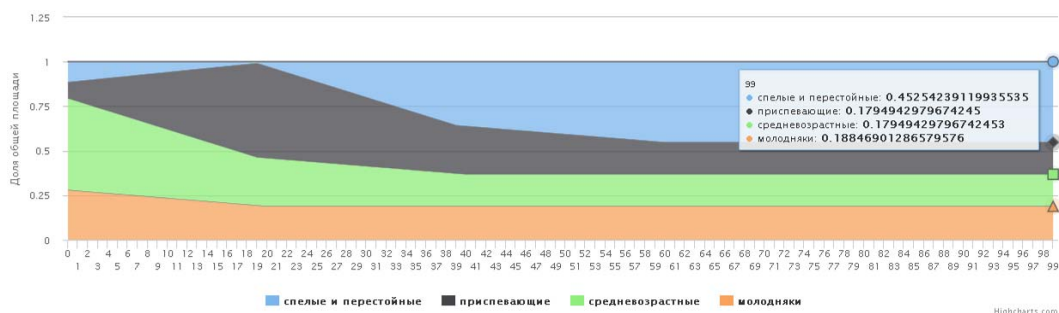


а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 2172 га; общая площадь – 280170 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>

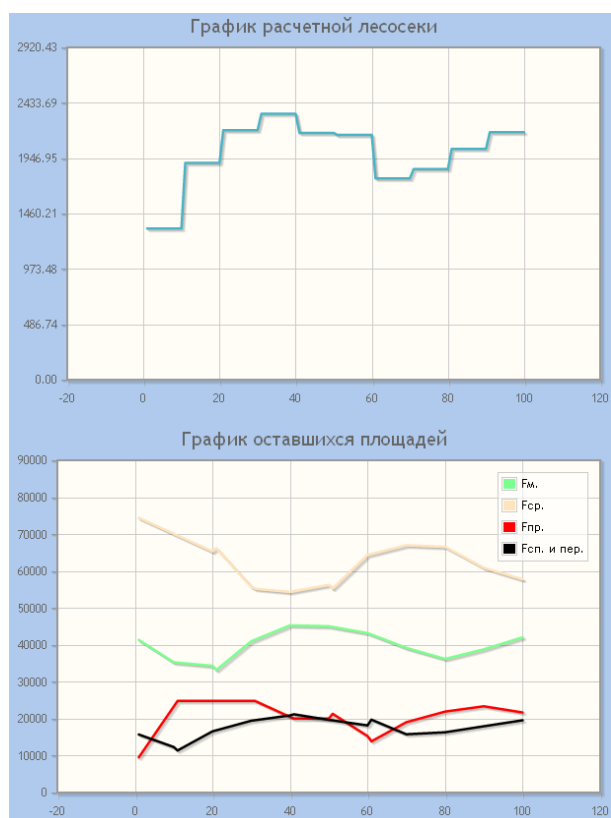


б – согласно действующему лесохозяйственному регламенту (1-я возрастная лесосека); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.19 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Ель, 4-5 бонитет; Пудожское центральное лесничество

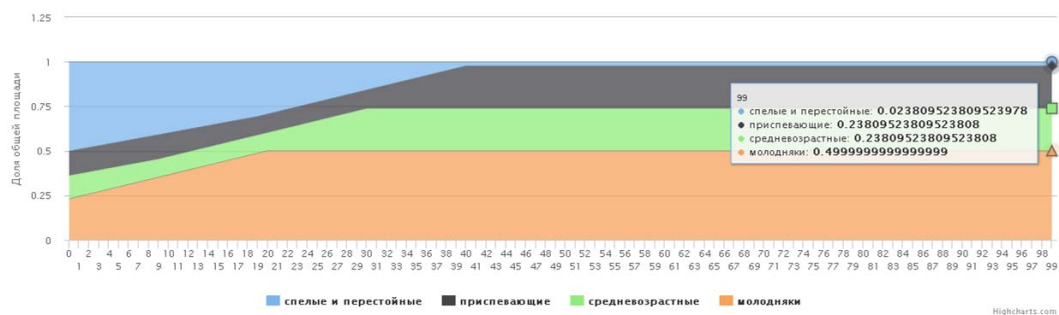


а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 1265 га; общая площадь – 140994 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>

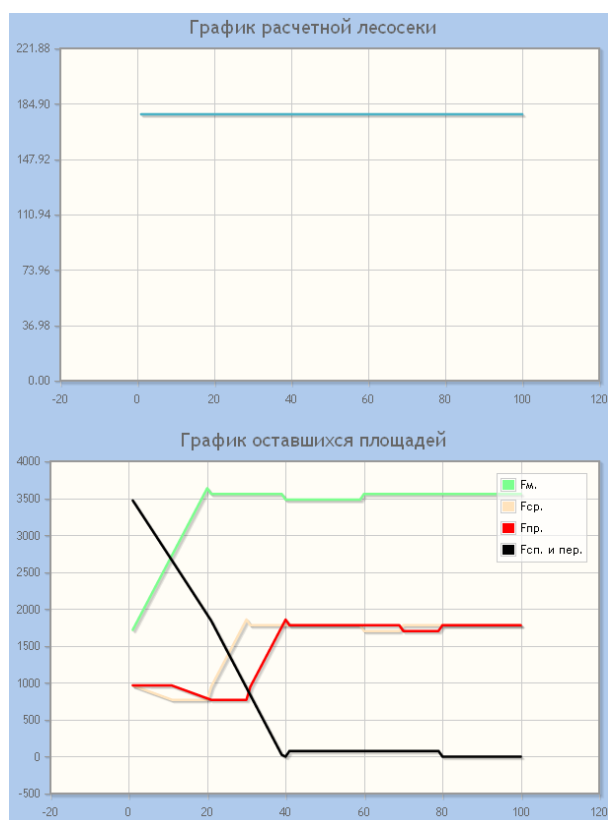


б – согласно действующему лесохозяйственному регламенту (1-я возрастная лесосека); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.20 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Береза, все бонитеты; Пудожское центральное лесничество



а – согласно предлагаемой методике расчета пользования лесом (см. раздел 2); ежегодная площадь рубок на прогнозируемый период – 169 га; общая площадь – 7115 га; по оси абсцисс – время в годах, по оси ординат – доли от общей площади; расчет выполнен с использованием программы, размещенной на <http://forest-karelia.ru/lesoseka/>



б – согласно действующему лесохозяйственному регламенту (1-я возрастная лесосека); по оси абсцисс – время в годах, по оси ординат – площадь в гектарах; расчет выполнен с использованием программы, размещенной на <http://jurist73.jurist-obzor.ru/>

Рисунок 4.21 – Прогноз изменения соотношения площадей различных возрастных групп насаждения Осина, все бонитеты; Пудожское центральное лесничество

4.6. Выводы и рекомендации

Выполненный сравнительный анализ динамики использования лесов для четырех центральных лесничеств Республики Карелия: Беломорского, Сегежского, Медвежьегорского и Пудожского, на период 100 лет, по действующей [10] и предлагаемой методикам (см. раздел 2 и программу на <http://forest-karelia.ru/lesoseka/>), позволяет сформулировать следующие выводы и рекомендации:

1. Материалы лесохозяйственных регламентов позволяют заинтересованному специалисту рассчитывать параметры расчетных лесосек (в т. ч. для неистощительного пользования лесом) в эксплуатационных лесах, без учета транспортной и/или экономической доступности лесных ресурсов;

2. Предлагаемый подход к определению расчетной лесосеки для спелых и перестойных эксплуатационных лесов и разработанная на его основе программа ПЭВМ в среде JavaScript позволяет вычислять параметры расчетной лесосеки, обеспечивающей непрерывное неистощительное лесопользование в этих лесах, на протяжении заданного числа лет;

3. В отличие от применяемых ныне формул расчета лесосеки, основанных на мысленной модели истощения запасов перестойных, спелых, приспевающих и др. насаждений через то или иное время, предлагаемый подход и разработанная на его основе программа ПЭВМ позволяют рассчитывать этот процесс во всех деталях относительно всех разновозрастных групп деревьев, что обеспечивает значительно бóльшую универсальность, а также позволяет рекомендовать указанный подход и программу ПЭВМ для практического применения.

Рекомендации по дальнейшему совершенствованию разработанных принципов, алгоритмов и программ ПЭВМ для вычисления параметров расчетной лесосеки, обеспечивающей непрерывное неистощительное лесопользование в эксплуатационных лесах, представлены в подразделе 2.6.

ЗАКЛЮЧЕНИЕ

В процессе разработки принципов формирования алгоритма для вычисления параметров расчетной лесосеки неистощительного пользования, на основе моделирования динамики развития лесных участков, подвергающихся регулярным антропогенным воздействиям в виде рубок спелых и перестойных лесных насаждений (т.н. «рубок главного пользования») было выполнено следующее:

- разработаны новые принципы формирования алгоритма для вычисления параметров расчетной лесосеки на основе данных о лесном фонде;
- разработаны программы (автономные – в среде Excel и среде R; для использования в сети Интернет – на JavaScript) для вычисления параметров расчетной лесосеки на основе новых подходов к определению расчетной лесосеки, ретроспективных данных о лесном фонде и данных о потребности в древесине;
- выполнен сравнительный анализ действующей и авторской методик определения расчетной лесосеки с использованием разработанной программы в среде JavaScript, по материалам лесохозяйственных регламентов четырех центральных лесничеств Республики Карелия: Беломорского, Сегежского, Медвежьегорского и Пудожского, на период 100 лет.
- разработаны рекомендации по выбору наилучшего варианта величины расчетной лесосеки на основе новых математических моделей;
- разработаны рекомендации по дальнейшим направлениям исследований в указанном направлении.

Разработанная программа на JavaScript для вычисления параметров расчетной лесосеки на основе авторской методики размещена в тестовом режиме в сети Интернет по адресу: <http://forest-karelia.ru/lesoseka/> и может быть перенесена на другой адрес. Программа работает во всех современных web-браузерах (Firefox, Opera, Chrome, Safari, Yandex, Internet Explorer) и может быть использована в практической деятельности специалистами лесного сектора.

Представленное исследование выполнено при поддержке проекта WWF и IKEA «Содействие устойчивому лесопользованию в России на основе сотрудничества со всеми заинтересованными сторонами в сферах государственного управления, бизнеса и образования».

Авторы благодарят Координатора проектов по лесам высокой природоохранной ценности WWF России Кобякова Константина Николаевича за активное участие в обсуждении полученных результатов.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Общесоюзные нормативы для таксации лесов (утв. Приказом Госкомлеса СССР от 28.02.1989 г. № 38) [Электронный ресурс] // Консультант Плюс. – Режим доступа: <http://base.consultant.ru/cons/cgi/online.cgi?req=doc;base=ESU;n=15893>, свободный (дата обращения: 13.05.2014). – Загл. с экрана. – Яз. рус.
2. Спелость леса [Электронный ресурс] // Яндекс словари. – Режим доступа: [http://slovari.yandex.ru/~книги/БСЭ/Спелость леса/](http://slovari.yandex.ru/~книги/БСЭ/Спелость_леса/), свободный (дата обращения: 13.05.2014). – Загл. с экрана. – Яз. рус.
3. Булыгин Н. Е. Дендрология: учебник для высш. учеб. заведений [Текст] / Н. Е. Булыгин. – 2-е изд., перераб. и доп. – Л.: Агропромиздат, Ленингр. отд-ние, 1991. – 352 с.
4. Арефьев С. П. О максимальном возрасте деревьев на территории Тюменской области [Электронный ресурс] / С. П. Арефьев // Путь в Сибирь. – Режим доступа: <http://library.ikz.ru/georg-steller/aus-sibirien-2013-2009/arefev-s.-p.-o-maksimalnom-vozraste-derevev-na>, свободный (дата обращения: 13.05.2014). – Загл. с экрана. – Яз. рус.
5. Forest Finland in Brief 2013 [Текст] / ed. Yrjö Sevola; Finnish Forest Research Institute, METLA. – Vantaa: Kopijyvä, 2013. – 48 p.
6. Болотов О. В. Основы расчета и планирования устойчивого управления лесами и лесопользованием: монография [Текст] / О. В. Болотов, Ю. М. Ельдештейн, А. С. Болотова и др. – Красноярск: СибГТУ, 2005. – 181 с.
7. Болотов О. В. Информационное обеспечение для планирования рационального лесопользования [Текст] / О. В. Болотов // ИВУЗ. «Лесной журнал». – 2011. – № 4. – С. 11–16.
8. Лесохозяйственные регламенты центральных лесничеств Карелии на 2012-2021 гг. [Электронный ресурс] // Официальный интернет-портал Республики Карелия «Карелия официальная». – Режим доступа: http://gov.karelia.ru/gov/Power/Committee/Forest/lh_regl21.html, свободный (дата обращения: 13.05.2014). – Загл. с экрана. – Яз. рус.
9. Сеницын С. Г. Расчет размера лесопользования [Текст] / С. Г. Сеницын, Н. А. Моисеев, В. В. Загребов и др. – М.: Лесная промышленность, 1973. – 176 с.
10. Приказ Федерального агентства лесного хозяйства (Рослесхоз) от 27.05.2011 г. № 191 «Об утверждении Порядка исчисления расчетной лесосеки» (зарегистрирован в Минюсте РФ 06.07.2011 г. № 21276) [Электронный ресурс] // Интернет-портал

«Российской Газеты». – Режим доступа: <http://www.rg.ru/2011/07/07/lesoseka-site-dok.html>, свободный (дата обращения: 13.05.2014). – Загл. с экрана. – Яз. рус.

11. Коросов А. В. Имитационное моделирование в среде MS Excel (на примерах из экологии) [Текст] / А. В. Коросов. – Петрозаводск: Изд-во ПетрГУ, 2002. – 212 с.

12. Буховец А. Г. Статистический анализ данных в системе R: учебное пособие [Текст] / А. Г. Буховец, П. В. Москалев, В. П. Богатова и др. – Воронеж: Изд-во ВГАУ, 2010. – 124 с.

13. Джефферс Дж. Введение в системный анализ: применение в экологии [Текст] / Дж. Джефферс. – М.: Мир, 1981. – 252 с.

14. JavaScript [Электронный ресурс] // Википедия – свободная энциклопедия. – Режим доступа: <http://ru.wikipedia.org/wiki/JavaScript>, свободный (дата обращения: 13.05.2014). – Загл. с экрана. – Яз. рус.

15. Владимиров И. Н. Моделирование пространственно-временной динамики лесных ресурсов с использованием интеллектуальной ГИС [Электронный ресурс] / И. Н. Владимиров, А. К. Попова // Академическое издательство «ГЕО». – Режим доступа: <http://www.izdatgeo.ru/pdf/gipr/2009-1/26.pdf>, свободный (дата обращения: 13.05.2014). – Загл. с экрана. – Яз. рус.

16. Лесной план Республики Карелия [Электронный ресурс] // Официальный интернет-портал Республики Карелия «Карелия официальная». – Режим доступа: http://www.gov.karelia.ru/gov/Power/Committee/Forest/Docum/forest_plan.zip, свободный (дата обращения: 13.05.2014). – Загл. с экрана. – Яз. рус.

ЛИСТИНГ ПРОГРАММЫ НА ЯЗЫКЕ JAVASCRIPT

```

/*****
* lesoseka
* install: place div element with class="lesoseka" into DOM
* author: Alexander Zarodov
* requires prototype v 1.7 library
*****/

var lesoseka = Class.create({

    //initialize: function(element = null) {
    yearStep: 5, // шаг модели
    ageClasses: 24, // количество возрастных классов
    iterationsNumber: 26, // количество итераций модели
    death: 0.01, // отпад по умолчанию
    volumePerArea: 60, // объем на единицу площади
    harvestVolume: 1.5, // объем изъятия
    area: 1, // площадь, занимаемая возрастным классом (значение по умолчанию)
    arrAgeGroups: new Array(), // массив возрастных классов
    harvestAge: 81, // возраст начала рубки
    harvestVolume: 0, // планируемый объем изъятия
    forecastDuration: 100, // продолжительность временного отрезка для прогноза
    harvestDuration: 100, // количество лет для оптимизации объема изъятия
    volumePerArea: 200, // запас объема на площадь
    volumePPerArea: 254,
    volumeBPerArea: 176,
    volumeDeposit: 0, // запас объема древесины на всей площади
    harvestedArea: 0, // площадь, отведенная для рубок

    initialize: function(element) {
        // конструктор экземпляра класса
        if(element != null){

```

```

        //this.element = element;
        this.rootElement = element;

        this.volumeMode = lesoseka.VOLUME_FIXED;
        this.lesosekaMode = lesoseka.FORM_NEW_LESOSEKA;

        //this.baseElement = element;
        this.arrId = parseInt(element.readAttribute('arr'));
        this.rootElement.writeAttribute('arr', this.arrId);

        this.makeNewLesosekaForm();
        this.makeOldLesosekaForm();
        this.makeFormSelectControl();

        this.makeHelpers();
    }
},

// параметры объекта по умолчанию

    modelMode: 0, // режим работы модели: 0 - не выбран, 1 - прогноз по указанному
объему изъятия, 2 - оптимизация объема изъятия, для пользования в течение
указанного времени
    arrModelModes: new Array(
        {name: "", caption: 'выберите режим модели', value: 0, method: false},
        {name: 'прогноз продолжительности пользования', caption: 'прогноз
продолжительности пользования', value: 1, method: 'setModel1'},
        {name: 'расчет объема пользования', caption: 'расчет объема пользования',
value: 2, method: 'setMode2'}
    ),

    //////////////////////////////////////

    // создание форм приложения и элементов управления ими
    makeFormSelectControl: function(){
        // создает элементы переключения форм приложения

```

```

this.elFormSelector = document.createElement('div');
this.elFormSelector.className('lesFormSelector');

var objClasses = {mode1: new Array('lesLesosekaModeBut'), mode2: new
Array('lesLesosekaModeBut')};
objClasses['mode' + this.lesosekaMode].push('active');

var mode1Class = ' class=' + objClasses.mode1.join(' ') + '';
var mode2Class = ' class=' + objClasses.mode2.join(' ') + '';

this.elFormSelector.innerHTML = '<div mode=' +
lesoseka.FORM_NEW_LESOSEKA + ' ' + mode1Class + '>Новый метод расчета
лесосеки</div><div mode=' + lesoseka.FORM_OLD_LESOSEKA + ' ' + mode2Class +
'>Формулы исчисления расчетной лесосеки</div>';

this.rootElement.insert({top: this.elFormSelector});

this.elLesosekaMode1 = this.elFormSelector.select('div[mode="1"]')[0];
this.elLesosekaMode2 = this.elFormSelector.select('div[mode="2"]')[0];
this.arrLesosekaModeButtons = new Array(this.elLesosekaMode1,
this.elLesosekaMode2);

this.elFormSelector.observe('click', function(e){
    var element = e.findElement('div.lesLesosekaModeBut');
    if(element == null) return;
    else{
        var lesosekaObj = lesoseka.getObject(element);
        //var mode = element.readAttribute('rel');
        lesosekaObj.setLesosekaMode(element);
    }
});

},
makeNewLesosekaForm: function(){
    // создает форму новой версии расчета лесосеки

```

```

this.baseElement = document.createElement('div');
this.rootElement.insert({top: this.baseElement});
this.makeAgeClassesForm();

// форма выбора режима работы по фиксированному или динамическому
объему запаса
this.makeVolumeModeForm();

this.showStartForm();
this.setGrowthMatrixForm();
this.setModelForm();

},

makeOldLesosekaForm: function(){
    // создает форму старой версии расчета лесосеки
},

setLesosekaMode: function(element){
    // переключает режимы расчета лесосеки (новый метод или старые
формулы)
    for(var i = 0; i < this.arrLesosekaModeButtons.length; i++){
        if(this.arrLesosekaModeButtons[i] != element) {
            if(this.arrLesosekaModeButtons[i].hasClassName('active'))
this.arrLesosekaModeButtons[i].removeClassName('active');
        }
    }
    element.addClassName('active');
    this.lesosekaMode = parseInt(element.readAttribute('mode'));

    if(this.lesosekaMode == lesoseka.FORM_NEW_LESOSSEKA){
        // форма новой методики расчета лесосеки
        this.baseElement.show();
    }
}

```

```

    }
    else if(this.lesosekaMode == lesoseka.FORM_OLD_LESOSEKA){
        // форма расчета по старым формулам
        this.baseElement.hide();

    }
},

//////////
// создание хэлперов
makeHelpers: function(){
    var arrHelpElements = this.rootElement.select('[help]');
    for(var i = 0; i < arrHelpElements.length; i++){
        var helpElement = arrHelpElements[i];
        //alert(helpElement.readAttribute('help'));
        var helper = this.rootElement.select('[helpid="' +
helpElement.readAttribute('help') + '"'])[0];
        if(helper == undefined){
            helper = document.createElement('span');
            helper.innerHTML = '?';
            helper.addClassName('lesHelper');
            helper.writeAttribute('title', 'нажмите для информации');
            helper.writeAttribute('helpid', helpElement.readAttribute('help'));
            helpElement.insert({after: helper});
            helper.observe('click', this.showHelpHint);
        }
    }
},

showHelpHint: function(e){
    //Event.stop(e);
    var lesosekaObj = lesoseka.getObject(e.target);
    //alert(e.target.readAttribute('helpid'));
    if(lesosekaObj.elHelperHint == undefined){
        lesosekaObj.elHelperHint = document.createElement('div');
    }
}

```

```

        lesosekaObj.elHelperHint.addClassName('lesHint');
        lesosekaObj.rootElement.insert({after: lesosekaObj.elHelperHint});
        lesosekaObj.elHelperHint.innerHTML = '<div
class="lesHintClose">X</div><div class="lesHintText"></div>';

        lesosekaObj.elHelperHintText =
lesosekaObj.elHelperHint.select('div.lesHintText')[0];
        lesosekaObj.elHelperHintClose =
lesosekaObj.elHelperHint.select('div.lesHintClose')[0];
        lesosekaObj.elHelperHintClose.observe('click', function(e){
            var hint = e.target.up('div.lesHint');
            hint.hide();
        });
        lesosekaObj.elHelperHint.hide();
    }

    if(lesosekaObj.elHelperHint.visible() && e.target == lesosekaObj.prevHint) {
        lesosekaObj.elHelperHint.hide();
        return;
    }
    lesosekaObj.elHelperHint.show();
    lesosekaObj.elHelperHintText.innerHTML =
lesosekaObj.objHints[e.target.readAttribute('helpid')];
    var offset = e.target.cumulativeOffset();
    var dimensions = e.target.getDimensions();
    lesosekaObj.elHelperHint.setStyle({
        top: offset[1] + 'px',
        left: (offset[0] + dimensions.width) + 'px'
    });

    lesosekaObj.prevHint = e.target;
    //alert(dimensions.width);
    //e.target.absolutize();
    //alert(e.target.getStyle('left') + ' - ' + e.target.getStyle('top'));
    //e.target.absolutize();

```

```

//alert(e.target.cumulativeOffset()[0])
//e.target.relativize();
//lesosekaObj.elHelperHint.
//$(element).setStyle({
// 'float': 'left', // notice how float is surrounded by single quotes
// opacity: 0.5
//});
},

objHints: {
    death: 'Доля возможных потерь покрытых лесом площадей от негативных
природных или антропогенных воздействий (пожары, вредители, болезни леса и др.)',
    harvestAge: 'Возраст древостоя, начиная с которого ведутся рубки леса',
    deathDescr: 'Доля возможных потерь покрытых лесом площадей от
негативных природных или антропогенных воздействий (пожары, вредители, болезни
леса и др.)',
    harvestDuration: 'Желаемое количество лет равномерного (неубывающего в
объеме) пользования лесом',
},

//////////

//////////

// функции для работы с режимом объема запаса древесины
makeVolumeModeForm: function(){
    // выводит кнопки переключения режима работы по объему запаса -
фиксированному или динамическому
    this.elVolumeModeForm = document.createElement('div');
    this.elVolumeModeForm.className='lesVolumeModeForm';
    var objClasses = {mode1: new Array('lesVolumeModeBut'), mode2: new
Array('lesVolumeModeBut')};
    objClasses['mode' + this.volumeMode].push('active');
    var mode1Class = ' class="' + objClasses.mode1.join(' ') + '"';
    var mode2Class = ' class="' + objClasses.mode2.join(' ') + '"';

```



```

        this.elVolumeModeForm.innerHTML = '<div mode="" +
lesoseka.VOLUME_FIXED + "" + mode1Class + '>фиксированный объем
запаса</div><div mode="" + lesoseka.VOLUME_DYNAMIC + "" + mode2Class +
'>динамический объем запаса</div>';

        this.baseElement.insert({top: this.elVolumeModeForm});
        this.elVolumeMode1 = this.elVolumeModeForm.select('div[mode="1"]')[0];
        this.elVolumeMode2 = this.elVolumeModeForm.select('div[mode="2"]')[0];
        this.arrVolumeModeButtons = new Array(this.elVolumeMode1,
this.elVolumeMode2);

        this.elVolumeModeForm.observe('click', function(e) {
            var element = e.findElement('div.lesVolumeModeBut');
            if(element == null) return;
            else {
                var lesosekaObj = lesoseka.getObject(element);
                //var mode = element.readAttribute('rel');
                lesosekaObj.setVolumeMode(element);
            }
        });

    },

    makeVolumeDynamicForm: function() {
        this.elSetVolumeDynamicForm = document.createElement('div');
        this.elSetVolumeDynamicForm.addClassName('lesVolumeDynamicForm');
        var strOptions = "";
        var strTreeName = "";
        var strTypeName = "";
        var counter = 0;
        this.arrTreeTypes = new Array(); // массив факторов для построения модели
зависимости объема запаса от возраста
        for(var i = 0; i < this.arrTreeClasses.length; i ++){
            var treeClass = this.arrTreeClasses[i];
            strTreeName = treeClass.name;
            for(var ii = 0; ii < treeClass.class.length; ii ++){
                var treeType = treeClass.class[ii];
                strTypeName = treeType.name;

```

```

        treeType.fullName = strTreeName + ' ' + strTypeName;
        strOptions += '<option value="' + counter + '">' + strTreeName + ' ' +
strTypeName + '</option>';
        this.arrTreeTypes.push(treeType);
        counter ++;
    }
}

this.elSetVolumeDynamicForm.innerHTML = '<div><table><tbody><tr><td
align="right"><label for="forestType_' + this.arrId + '">Тип
древостоя</label></td><td><select id="forestType_' + this.arrId + '"
var="forestType"><option value="">Выберите тип древостоя</option>' + strOptions +
'</select></td></tr><tr><td align="right"><label for="maturityAge_' + this.arrId +
'">Возраст технической спелости</label></td><td><input id="maturityAge_' + this.arrId +
'" var="maturityAge" type="text" value=""></td></label></tr><tr><td align="right"><label
for="matureStorage_' + this.arrId + '">Запас на 1 га</label></td><td><input
id="matureStorage_' + this.arrId + '" var="matureStorage" type="text"
value=""></td></tr><tr><td align="right">Расчетный запас древесины, м³</td><td><span
var="volumeDepositDynamic"></span></td></tr></tbody></table></div>';

this.elVolumeModeForm.insert({after: this.elSetVolumeDynamicForm});

this.iniElements('forestType maturityAge matureStorage
volumeDepositDynamic'); // активирование элементов
this.setListeners('maturityAge matureStorage', 'keyup', this.filterInput);
this.setListeners('forestType maturityAge matureStorage', 'change', function(e){
    var lesosekaObj = lesoseka.getObject(e.target);
    //lesosekaObj.changeVolumeDynamic(e);
    lesosekaObj.changeValue(e);
    lesosekaObj.updateCurve();
});
// устанавливаются
//this.setListeners('yearStep ageClasses death', 'keyup', this.filterInput); //
указывается список переменных, а список элементов на основе переменных
//this.setListeners('death', 'keyup', this.filterInput); // указывается список
переменных, а список элементов на основе переменных

```

```

    },

    updateCurve: function(){
        // вычисляет кривую значений объема древесины и др. параметры
        if(this.curveData == undefined) this.curveData = {};
        if(this.forestType != undefined && !isNaN(this.forestType)){
            // выбран тип древостоя
            if(this.curveData[this.forestType] == undefined) {
                // эта кривая еще не строилась, выполняем вычисления значений
                this.curveData[this.forestType] = {};
                this.curveData[this.forestType].model = {};
                //alert(this.forestType);
                //alert(this.forestType + ' - ' +
this.arrTreeTypes[this.forestType].fullName);
                var factors = this.arrTreeTypes[this.forestType].factors;
                //alert(factors.a + ' - ' + factors.b + ' - ' + factors.c);

                // построение кривой хода роста
                for(var i = 0; i < 250; i++){
                    this.curveData[this.forestType].model[i] = factors.a + i *
factors.b + i * i * factors.c;
                }
            }

            if(this.maturityAge != undefined && !isNaN(this.maturityAge) &&
this.matureStorage != undefined && !isNaN(this.matureStorage)){
                // расчет поправочного коэффициента
                this.curveData[this.forestType].factor = this.matureStorage /
this.curveData[this.forestType].model[this.maturityAge];
                //alert('factor: ' + this.curveData[this.forestType].factor);
                if(this.curveData[this.forestType].real == undefined)
this.curveData[this.forestType].real = {};
                for(var i = 0; i < 250; i++){
                    this.curveData[this.forestType].real[i] =
this.curveData[this.forestType].model[i] * this.curveData[this.forestType].factor;

```

```

    }
  }
  this.calculateDynamicVolume();
}
},

calculateDynamicVolume: function(){
  // рассчитывает объем на основе объемов матрицы роста и динамической
  // кривой зависимости запаса объема от возраста
  // кривая хода роста уже построена
  if(this.arrGrowthMatrix != undefined && this.harvestAge != undefined &&
  !isNaN(this.harvestAge) && this.curveData != undefined && this.curveData[this.forestType]
  != undefined && this.curveData[this.forestType].real != undefined){
    // расчет объема запаса согласно кривой
    if(this.harvestAge > 0 && this.harvestAge < this.arrGrowthMatrix.length){

      var volume = 0;

      for(i = this.harvestAge; i < this.arrGrowthMatrix.length; i++){
        //var matrixIndex = i - 1; // поправка на сдвиг индекса матрицы
        // хода роста
        var matrixIndex = i;
        volume += this.arrGrowthMatrix[i].area *
        this.curveData[this.forestType].real[i];
      }
      //alert(strTrace);
      this.volumeDepositDynamic = volume;
      this.elVolumeDepositDynamic.innerHTML =
      this.volumeDepositDynamic;
    }
  }
},

calculateCurrDynamicVolume: function(objAreas){

```

```

        // выполняет расчет объема, доступного для рубки на основе объектного
массива объемов objAreas[i]=volume
        // i - соответствует году роста
        // volume - текущий объем для этого года
        // this.curveData[this.forestType].real[i] - запас на гектар для этого года роста
        var volume = 0;
        if(this.curveData != undefined && this.curveData[this.forestType] != undefined
&& this.curveData[this.forestType].real != undefined){
            var strTrace = "";
            for(var i in objAreas){
                strTrace += i + ' - ' + objAreas[i] + "\n";
                volume += objAreas[i] * this.curveData[this.forestType].real[i];
            }
            //alert(strTrace);
        }
        return volume;
    },

```

```

setVolumeMode: function(element){
    // переключает режимы работы формы по типу запаса объема древесины
    for(var i = 0; i < this.arrVolumeModeButtons.length; i++){
        if(this.arrVolumeModeButtons[i] != element) {
            if(this.arrVolumeModeButtons[i].hasClassName('active'))
this.arrVolumeModeButtons[i].removeClassName('active');
        }
    }
    element.addClassName('active');
    this.volumeMode = parseInt(element.readAttribute('mode'));

    if(this.volumeMode == lesoseka.VOLUME_FIXED){
        if(this.elSetVolumeDynamicForm != undefined)
this.elSetVolumeDynamicForm.hide();
        this.lbVolumePerArea.show();
        this.lbVolumeDeposit.show();
    }
}

```

```

        else if(this.volumeMode == lesoseka.VOLUME_DYNAMIC){
            if(this.elSetVolumeDynamicForm == undefined)
this.makeVolumeDynamicForm();
            else this.elSetVolumeDynamicForm.show();
            this.lbVolumePerArea.hide();
            this.lbVolumeDeposit.hide();
        }
    },
    //////////////////////////////////////

    //////////////////////////////////////

    // описание классов древостоя
    arrTreeClasses: new Array(
        {name: 'Сосна', class: new Array(
            {
                name: '1-3',
                factors: {
                    a: -56.4111066938262,
                    b: 7.58969884659398,
                    c: -0.0238062105521096
                },
            },
            {
                name: '4-5',
                factors: {
                    a: -16.5509318720207,
                    b: 2.58439081526528,
                    c: -0.00616127250103647
                },
            }
        )
    },
    {name: 'Ель', class: new Array(
        {
            name: '1-3',

```

```

        factors: {
            a: -120.994644002244,
            b: 8.63178519458941,
            c: -0.0203764231101946
        },
    },
    {
        name: '4-5',
        factors: {
            a: -12.9605442818638,
            b: 2.52049436199816,
            c: -0.00589495718186825
        },
    }
)
},
{name: 'Осина', class: new Array(
    {
        name: '1-5',
        factors: {
            a: -27.9833318763529,
            b: 6.1537194266615,
            c: -0.0252652711320904
        }
    }
)
},
{name: 'Береза', class: new Array(
    {
        name: '1-5',
        factors: {
            a: -29.5499966223369,
            b: 4.53963701171461,
            c: -0.0174623555337781
        }
    }
)
}

```

```

    }
  )
}
),
//////////

```

matrixEntered: {}, // объект который хранит введенные вручную данные для
возрастных классов

// справочник по типам данных

```

dataTypes: {
  modelMode: 'int_unsigned',
  yearStep: 'int_unsigned',
  ageClasses: 'int_unsigned',
  iterationsNumber: 'int_unsigned',
  death: 'float_unsigned',
  volumePerArea: 'float_unsigned',
  harvestVolume: 'float_unsigned',
  forecastDuration: 'int_unsigned',
  ageLabel: 'string',
  harvestAge: 'int_unsigned',
  harvestVolume: 'float_unsigned',
  harvestDuration: 'int_unsigned',
  volumePerArea: 'float_unsigned',
  volumeDeposit: 'float_unsigned',
  forestType: 'int_unsigned',
  maturityAge: 'int_unsigned',
  matureStorage: 'float_unsigned'
},

```

// методы экземпляра класса

```

makeAgeClassesForm: function(){
  // вывод формы редактирования возрастных классов
  this.elAgeClassesForm = document.createElement('div');
  this.elAgeClassesForm.className('lesAgeClassesForm');
}

```



```

        this.elAgeClassesForm.innerHTML = '<div><input but="addAgeClass"
type="button" value="Добавить группу возраста" class="lesButton"></div><div><table
class="lesDataGrid"><thead><th title="Нижний предел группы возраста, лет">От</th><th
title="Верхний предел группы возраста, лет">До</th><th class="lesAgeClassArea"
title="Площадь группы возраста, га">Площадь, га</th><th
class="lesAgeLabelCell">Примечание</th></thead><tbody
rel="ageClassesList"></tbody></table><div>Общая площадь, га: <span
var="totalArea">0</span></div></div><div>Эксплуатационная площадь, га: <span
var="harvestedArea">0</span></div></div>';

```

```

        this.baseElement.insert({ top: this.elAgeClassesForm });

```

```

        this.butAddAgeClass =
this.elAgeClassesForm.select('input[but="addAgeClass"]')[0];
        this.butAddAgeClass.observe('click', this.butAddAgeClassHandler);
        this.elAgeClassesList =
this.elAgeClassesForm.select('tbody[rel="ageClassesList"]')[0];
        this.elTotalArea = this.elAgeClassesForm.select('span[var="totalArea"]')[0];
        this.elHarvestedArea =
this.elAgeClassesForm.select('span[var="harvestedArea"]')[0];
        //alert(this.elTotalArea)

```

```

        this.elAgeClassesForm.observe('click', function(e) {
            var elInput = wcElements.setTextInput(e);
            if(elInput){
                var lesosekaObj = lesoseka.getObject(elInput);
                elInput.observe('change', lesosekaObj.changeAgeClass);
                //elInput.observe('keyup', lesosekaObj.filterAgeClass);
                elInput.observe('keyup', lesosekaObj.filterInput);
            }
            else{
                var button = e.findElement("[action='kill']");
                if(button != undefined){
                    // удаление элементы списка возрастных классов
                    var lesosekaObj = lesoseka.getObject(button);

```

```

        lesosekaObj.killAgeClass(button);
    }
    //alert(button);
}
//alert(elInput);
});
},

```

```

killAgeClass: function(element){
    // удаление возрастного класса
    var index = parseInt(element.readAttribute('index'));
    //var ageClass = this.arrAgeGroups[index];
    //ageClass.elForm.remove();
    this.arrAgeGroups[index].elForm.remove();
    this.arrAgeGroups.splice(index, 1);
    //alert('kill ' + index);
    this.resortAgeClassesList();
},

```

```

changeAgeClass: function(e){
    var lesosekaObj = lesoseka.getObject(e.target);
    var variable = e.target.readAttribute('var');
    var index = parseInt(e.target.readAttribute('index'));
    //alert(variable + ' - ' + index);
    var ageClass = lesosekaObj.arrAgeGroups[index];

    switch(variable){
        case 'endAge':
            var endAge = ageClass.setEndAge(parseFloat(e.target.value));
            if(index < (lesosekaObj.arrAgeGroups.length - 1)){
                lesosekaObj.arrAgeGroups[index + 1].setStartAge(endAge);
            }
            break;
        case 'ageArea':
            ageClass.setArea(e.target.value);

```

```

        lesosekaObj.setTotalArea();
        lesosekaObj.setHarvestedArea();
        break;
    case 'ageLabel':
        ageClass.setLabel(e.target.value);
        break;
    }
},

```

```

setTotalArea: function(){
    this.totalArea = 0;
    for(var i = 0; i < this.arrAgeGroups.length; i++){
        var ageGroup = this.arrAgeGroups[i];
        var area = ageGroup.getArea();
        if(area) this.totalArea += area;
    }
    this.elTotalArea.innerHTML = this.totalArea;
    this.updateVolumeDeposit();
    //alert(this.elTotalArea);
},

```

```

setHarvestedArea: function(){
    this.setGrowthMatrixForm();// обновление матрицы хода роста для
    пролучения массива arrGrowthMatrix
    this.harvestedArea = 0;
    if(this.arrGrowthMatrix == null) return false;
    for(var i = this.harvestAge; i < this.arrGrowthMatrix.length; i++){
        this.harvestedArea += this.arrGrowthMatrix[i].area;
    }
    this.elHarvestedArea.innerHTML = this.harvestedArea;
    //alert(this.arrGrowthMatrix.length);
    this.updateVolumeDeposit();
    if(this.volumeMode == lesoseka.VOLUME_DYNAMIC)
this.calculateDynamicVolume();

```

```
},
```

```
resortAgeClassesList: function(){  
    // расставляет индексы в DOM списке возрастных классов  
    for(var i = 0; i < this.arrAgeGroups.length; i ++){  
        this.arrAgeGroups[i].setIndex(i);  
    }  
},
```

```
makeStartForm: function(){  
    // вывод стартовой формы  
    this.elStartForm = document.createElement('div');  
    this.elStartForm.addClassName('lesStartForm');  
  
    this.elStartForm.innerHTML = '<label>Доля возможных потерь площадей (по  
умолчанию)<input type="text" var="death" help="death" class="lesTextField" value="" +  
this.death + ""></label><input type="button" but="setGrowthMatrix" value="обновить  
матрицу хода роста">';  
  
    this.baseElement.insert({bottom: this.elStartForm});  
  
    //this.iniElements(new Array('yearStep', 'ageClasses', 'iterationsNumber', 'death'));  
    // активирование элементов  
    //this.iniElements(['yearStep', 'ageClasses', 'iterationsNumber', 'death']); //  
    активирование элементов  
    //this.iniElements('yearStep ageClasses death'); // активирование элементов  
    this.iniElements('death'); // активирование элементов  
  
    // устанавливаются  
    this.setListeners('death', 'keyup', this.filterInput); // указывается список  
переменных, а список элементов на основе переменных  
    //this.setListeners('yearStep ageClasses death', 'change', this.changeValue);  
    //this.setListeners('death', 'change', this.changeValue);  
  
    this.elDeath.observe('change', function(e){
```

```

        var lesosekaObj = lesoseka.getObject(e.target);
        lesosekaObj.changeValue(e);
        lesosekaObj.setGrowthMatrixForm();
    });
    //this.inputAgeClasses.observe('change', this.changeValue);

    this.butSetGrowthMatrix =
this.elStartForm.select('input[but="setGrowthMatrix"]')[0];
    this.butSetGrowthMatrix.hide();// скрываем кнопку
    //this.butSetAgeClasses.observe('click', this.setGrowthMatrixForm);
    this.butSetGrowthMatrix.observe('click', this.butSetGrowthMatrixHandler);
    //butSetAgeClassesHandler
},

iniElements: function(arrVarList){
    var elements = wcData.array(arrVarList);
    for(var i = 0; i < elements.length; i++){
        var item = elements[i];
        this['el' + wcData.strFirstUppercase(item)] =
this.baseElement.select('[var="' + item + '"]')[0];
    }
},

setListeners: function(arrVarList, eventName, handler){
    var elements = wcData.array(arrVarList);
    for(var i = 0; i < elements.length; i++){
        var item = elements[i];
        var element = this['el' + wcData.strFirstUppercase(item)];
        element.observe(eventName, handler);
    }
},

showStartForm: function(){
    if(this.elStartForm == null) this.makeStartForm();
    else this.elStartForm.show();
}

```

```

    },
    hideStartForm: function(){
        if(this.elStartForm == null) this.makeStartForm();
        this.elStartForm.hide();
    },

    //setGrowthMatrixForm: function(e){
    setGrowthMatrixForm: function(){
        // вывод формы возрастных классов
        if(this.elSetGrowthMatrixForm != null){
            // сбрасывается ранее выведенная форма
            if(this.elDeathRow != null){
                this.elDeathRow.purge();
                this.elDeathRow.remove();
            }
            this.elSetGrowthMatrixForm.purge();
            this.elSetGrowthMatrixForm.remove();
        }

        this.elSetGrowthMatrixForm = document.createElement('div');
        this.elSetGrowthMatrixForm.className('lessetGrowthMatrixForm');
        this.elStartForm.insert({after: this.elSetGrowthMatrixForm});

        // создание массива возрастных классов
        this.arrGrowthMatrix = new Array();
        this.totalArea = 0;
        var counter = 0;
        for(var i = 0; i < this.arrAgeGroups.length; i++){
            var ageGroup = this.arrAgeGroups[i];
            var iterationArea = ageGroup.getIterationArea();
            var area = ageGroup.getArea();
            var iterations = ageGroup.getIterations();
            if(iterationArea){
                this.totalArea += area;
                for(var ii = 0; ii < iterations; ii++){

```

```

        var obj = {};
        obj.name = 'b' + counter;

        //alert(this.matrixEntered.death);
        if(this.matrixEntered.death == undefined ||
this.matrixEntered.death[obj.name] == undefined) obj.death = this.death;
        else obj.death = this.matrixEntered.death[obj.name];

        obj.area = iterationArea;
        //obj.death = this.death;
        this.arrGrowthMatrix.push(obj);
        counter ++;
    }
}
else i = this.arrAgeGroups.length;

}
var strName = "";
var strDeath = "";
var strArea = "";
var strPart = "";
for(i = 0; i < this.arrGrowthMatrix.length; i ++){
    this.arrGrowthMatrix[i].part = this.arrGrowthMatrix[i].area / this.totalArea;
    var obj = this.arrGrowthMatrix[i];
    strName += '<th>' + obj.name + '</th>';
    strDeath += '<td><div var="death" index="' + i + "'
_class="lesTableTextInput">' + obj.death + '</div></td>';
    strArea += '<td><div var="area" index="' + i + "'
_class="lesTableTextInput">' + obj.area + '</div></td>';
    strPart += '<td><div var="part" index="' + i + "'
_class="lesTableTextInput">' + obj.part + '</div></td>';
}

var str = '<table class="lesDataGrid">';
// ВЫВОД ВОЗРАСТНЫХ КЛАССОВ

```

```

    str += '<thead><tr><th>Шаг моделирования возраста, год</th>' + strName +
'</tr></thead>';

    // вывод полей для ввода отпада
    str += '<tbody><tr rel="death"><td>Доля возможных потерь площадей<span
help="deathDescr"></span></td>' + strDeath + '</tr>';

    str += '<!--<tr rel="area"><td>Занимаемая площадь, ед.</td>' + strArea + '</tr>-
->';

    //str += '<tr rel="part"><td>Доля от общей площади  $\sum=1$ </td>' + strPart +
'</tr>';

    str += '<tr rel="part"><td>Доля от общей площади</td>' + strPart + '</tr>';
    str += '</tbody></table>';

    this.elSetGrowthMatrixForm.innerHTML = str;

    this.elDeathRow = this.elSetGrowthMatrixForm.select('[rel="death"]')[0];
    //this.elAreaRow = this.elSetGrowthMatrixForm.select('[rel="area"]')[0];
    //this.elPartRow = this.elSetGrowthMatrixForm.select('[rel="part"]')[0];
    //this.elSetGrowthMatrixForm.observe('click', this.setGrowthMatrixFormFocus);
    //alert(this.elDeathRow + ' - is element: ' + Object.isElement(this.elDeathRow));
    this.elDeathRow.observe('click', function(e) {
        var elInput = wcElements.setTextInput(e);
        if(elInput) {
            var lesosekaObj = lesoseka.getObject(elInput);
            elInput.observe('change', lesosekaObj.changeDeath);
            elInput.observe('keyup', lesosekaObj.filterFloat);
        }
    });
    this.makeHelpers();
},

setModelForm: function() {
    // вывод формы для ввода параметров модели
    if(this.elModelForm == null) {

        //var currVolumeDeposit = this.volumeDeposit;

```



```

var currVolumeDeposit = this.getVolumeDeposit();

this.elModelForm = document.createElement('div');
this.elModelForm.className('lesModelForm');

this.elModelForm.innerHTML = '<div><label>Возраст начала рубки
древостоев, лет<input type="text" var="harvestAge" help="harvestAge"
class="lesTextField" value="" + this.harvestAge + ""></label></div><div><label>Запас
древесины на 1 га, м³<input type="text" var="volumePerArea" class="lesTextField"
value="" + this.volumePerArea + ""></label></div><div>Общий запас древесины на
эксплуатационной площади, м³: <span var="volumeDeposit">' + currVolumeDeposit +
'</span></div>';

var strOptions = "";
var selected = "";
for(var i = 0; i < this.arrModelModes.length; i++){
    selected = "";
    var modelItem = this.arrModelModes[i];
    if(i == this.modelMode) selected = ' selected';
    strOptions += '<option value="" + modelItem.value + "" + selected + '>'
+ modelItem.caption + '</option>';
}
this.elModelForm.innerHTML += '<div><select var="modelMode">' +
strOptions + '</select></div>';

//this.elModelForm.innerHTML += '<table style="opacity:
0.5;"><tbody><tr><td>Запас пиловочника</td><td><input type="text"
var="volumePPerArea" class="lesTextField" value="" + this.volumePPerArea + ""
disabled></td></tr><tr><td>Запас баланса</td><td><input type="text"
var="volumeBPerArea" class="lesTextField" value="" + this.volumeBPerArea + ""
disabled></td></tr></tbody></table>';

this.elModelForm.innerHTML += '<div rel="modeHolder"></div>';
//this.elModelForm.innerHTML += '<div>Доля изъятия: <span
var="volumeDeposit">' + this.volumeDeposit + '</span></div>';
this.baseElement.insert({bottom: this.elModelForm});

```

```

        this.elModeHolder = this.elModelForm.select('[rel="modeHolder"]')[0]; //
        ссылка на контейнер с полями, зависящими от режима работа модели

        this.iniElements('harvestAge volumePerArea modelMode volumeDeposit');
        // активирование элементов

        this.setListeners('harvestAge volumePerArea', 'keyup', this.filterInput);//
        указывается список переменных, а не список элементов на основе переменных

        this.setListeners('harvestAge volumePerArea', 'change', this.changeValue);

        this.lbVolumePerArea = this.elVolumePerArea.up('label');
        this.lbVolumeDeposit = this.elVolumeDeposit.up('div');

        this.elVolumePerArea.observe('change', function(e){
            var lesosekaObj = lesoseka.getObject(e.target);
            lesosekaObj.updateVolumeDeposit();
        });

        this.elModelMode.observe('change', this.setModelMode);

        this.elHarvestAge.observe('change', function(e){
            var lesosekaObj = lesoseka.getObject(e.target);
            lesosekaObj.setHarvestedArea();
            if(lesosekaObj.volumeMode == lesoseka.VOLUME_DYNAMIC)
lesosekaObj.calculateDynamicVolume();
        });
    }
},

setModelMode: function(e){
    // устанавливает режим работы модели
    var lesosekaObj = lesoseka.getObject(e.target);
    var index = parseInt(e.target.value);
    lesosekaObj.modelMode = index;
    lesosekaObj.resetMode();
    //alert(lesosekaObj.arrModelModes[index].method);

```

```

        if(!lesosekaObj.arrModelModes[index].method) return;
        lesosekaObj[lesosekaObj.arrModelModes[index].method]();
        lesosekaObj.makeHelpers();
        //alert(e.target.value);
    },

    resetMode: function(){
        // деактивирование всех режимов режима
        if(this.elMode1Holder != null) this.elMode1Holder.hide();
        if(this.elMode2Holder != null) this.elMode2Holder.hide();
    },
    setMode1: function(){
        // активирование первого режима работы
        if(this.elMode1Holder == null) {
            this.elMode1Holder = document.createElement('div');
            this.elMode1Holder.innerHTML = '<div><label>Планируемый объем
рубки леса, м³<input type="text" var="harvestVolume" class="lesTextField" value="" +
this.harvestVolume + ""></label></div><label>Продолжительность прогноза, лет <input
type="text" var="forecastDuration" class="lesTextField" value="" + this.forecastDuration +
""></label></div><div><input type="button" but="calculateForecast" value="рассчитать
прогноз"></div>';
            this.elModeHolder.insert({bottom: this.elMode1Holder});
            this.iniElements('harvestVolume forecastDuration'); // активирование
элементов
            this.setListeners('harvestVolume forecastDuration', 'keyup',
this.filterInput); // указывается список переменных, а не список элементов на основе
переменных
            this.setListeners('harvestVolume forecastDuration', 'change',
this.changeValue);
            this.setListeners('harvestVolume forecastDuration', 'change',
this.butCalculateForecastHandler);
            this.butCalculateForecast =
this.elMode1Holder.select('input[but="calculateForecast"]')[0];
            //alert(this.butCalculateForecast);
            this.butCalculateForecast.observe('click', this.butCalculateForecastHandler);

```

```

    }
    var isReady = this.isReadyToCalculate();
    if(!isReady) this.butCalculateForecast.disable();
    else this.butCalculateForecast.enable();
    this.elMode1Holder.show();
  },
  setMode2: function() {
    // активирование второго режима работы
    if(this.elMode2Holder == null) {
      this.elMode2Holder = document.createElement('div');
      this.elMode2Holder.innerHTML = '<div><label>Количество лет
пользования<input type="text" var="harvestDuration" help="harvestDuration"
class="lesTextField" value="" + this.harvestDuration + ""></label></div><div><input
type="button" but="calculateHarvest" value="рассчитать объем
пользования"></div><div>Расчетный объем пользования, м³: <span
var="harvestVolumeEstimated"></span></div>';
      this.elModeHolder.insert({bottom: this.elMode2Holder});
      this.iniElements('harvestDuration harvestVolumeEstimated'); //
      активирование элементов
      this.setListeners('harvestDuration', 'keyup', this.filterInput); // указывается
      список переменных, а не список элементов на основе переменных
      this.setListeners('harvestDuration', 'change', this.changeValue);
      this.setListeners('harvestDuration', 'change',
this.butCalculateHarvestHandler);
      this.butCalculateHarvest =
this.elMode2Holder.select('input[but="calculateHarvest"]')[0];
      this.butCalculateHarvest.observe('click', this.butCalculateHarvestHandler);
    }
    var isReady = this.isReadyToCalculate();
    if(!isReady) this.butCalculateHarvest.disable();
    else this.butCalculateHarvest.enable();
    this.elMode2Holder.show();
  },

```

```

updateVolumeDeposit: function(){
    this.volumeDeposit = this.volumePerArea * this.harvestedArea;
    //this.volumeDeposit = this.volumePerArea * this.totalArea;
    this.elVolumeDeposit.innerHTML = this.volumeDeposit;
},

////////////////////////////////////

// обработчики событий
butCalculateForecastHandler: function(e){
    // режим работы 1 - расчет прогноза на заданное количество лет
    //alert('butCalculateForecastHandler');
    var lesosekaObj = lesoseka.getObject(e.target);
    lesosekaObj.calculateForecast();
},

butCalculateHarvestHandler: function(e){
    // режим работы 2 - расчет доли нистощительного изъятия
    var lesosekaObj = lesoseka.getObject(e.target);
    lesosekaObj.calculateHarvest();
},

isReadyToCalculate: function(){
    // функция проверяет все ли входные данные есть для выполнения расчетов
    var res = true;
    if(this.volumeMode == lesoseka.VOLUME_FIXED){
        // режим фиксированного объема запаса древесины
        if(isNaN(this.volumePerArea)) res = false;
        if(isNaN(this.volumeDeposit)) res = false;
    }
    else if(this.volumeMode == lesoseka.VOLUME_DYNAMIC){
        // режим динамического объема запаса древесины
        // проверка форма ввода входных данных для расчета кривой запаса
        //древостоя от возраста

```

```

        if(isNaN(this.forestType) || this.forestType === undefined) res = false;
        if(isNaN(this.maturityAge) || this.maturityAge === undefined) res = false;
        if(isNaN(this.matureStorage) || this.matureStorage === undefined) res =
false;

        }
        return res;
    },

    getVolumeDeposit: function(){
        if(this.volumeMode === lesoseka.VOLUME_FIXED) return this.volumeDeposit;
        else if(this.volumeMode === lesoseka.VOLUME_DYNAMIC){
            // запас, рассчитанный на основе кривых хода роста
            //alert(this.volumeDepositDynamic);
            return this.volumeDepositDynamic;
        }
        //var currVolumeDeposit = this.volumeDeposit;
    },

    calculateHarvest: function(){
        // расчет максимального объема изъятия

        //var harvestVolume = 1000; // объем изъятия
        // для первой отерации берем объем изъятия 3% от общего объема для рубок
        var currVolumeDeposit = this.getVolumeDeposit();

        var harvestVolume = currVolumeDeposit * 0.02;
        var step = currVolumeDeposit * 0.003;
        //this.calculateHarvestIteraion(harvestVolume);
        var arrSteps = new Array();
        arrSteps.push(step);
        var harvestVolumes = new Array();
        harvestVolumes.push(harvestVolume);
        var arrMaxVolParts = new Array();
        var currVolumePart = 0;

```

```

var sign = 1;
var increment = 0;
var arrIncrements = new Array();
var incrementStepRatio = 0;
var signPrev = sign;
var maxIterations = 400;

for(var i = 0; i < maxIterations; i++){
    currVolumePart = this.calculateHarvestIteraion(harvestVolume);

    if(arrMaxVolParts.length > 1){
        increment = Math.abs(currVolumePart -
arrMaxVolParts[arrMaxVolParts.length - 1]);
        arrIncrements.push(increment);
    }

    if(currVolumePart > 1){
        sign = -1;
    }
    else{
        sign = 1;
    }

    if(signPrev != sign){
        step = step * 0.5;
        signPrev = sign;
    }

    arrSteps.push(step);

    if(currVolumePart == arrMaxVolParts[arrMaxVolParts.length - 1]){
        i = maxIterations;
    }

    arrMaxVolParts.push(currVolumePart);

```

```

        harvestVolume += step * sign;

    }

    this.harvestVolumeEstimated = harvestVolume - step * sign
    this.elHarvestVolumeEstimated.innerHTML = this.harvestVolumeEstimated;

    this.showOptimizationResult();
},

calculateHarvestIteration: function(harvestVolume){
    // рассчитывает максимальную величину отношения изымаемой площади к
    // общей имеющейся для рубок площади за выбранный период
    //var currVolumeDeposit = this.volumeDeposit;
    var currVolumeDeposit = this.getVolumeDeposit();

    var value = 0;

    // формирование массива для первой итерации
    var arrMatrix = new Array();
    var i = 0;
    var currDeposit = 0;
    for(i = 0; i < this.arrGrowthMatrix.length; i++){
        arrMatrix.push(this.arrGrowthMatrix[i].part);
    }
    var arrResult = new Array();
    var arrHarvestParts = new Array();
    var harvestPart = harvestVolume / currVolumeDeposit;

    // выполнение прогона
    for(i = 1; i <= this.harvestDuration; i++){
        arrResult = this.makeIteration(arrMatrix, harvestPart);
    }
}

```



```

// агрегация элементов массива по отдельным классам
var currGroupIndex = 0;
var valuePart = 0;
currDeposit = 0;
var objAreas = {};
for(var ii = 0; ii < arrResult.length; ii++){
    if(this.arrAgeGroups[currGroupIndex].endAge < ii){
        //this.arrChartData[currGroupIndex].dataParts.push(valuePart);
        //this.arrChartData[currGroupIndex].dataArea.push(valuePart *
this.totalArea);

        currGroupIndex++;
        valuePart = 0;
    }
    valuePart += arrResult[ii];
    if(ii >= this.harvestAge){
        // расчет
        currDeposit += arrResult[ii];
        objAreas[ii] = arrResult[ii] * this.totalArea; // сохраняем объем
для этого года роста
    }

}
arrMatrix = arrResult;

var currHarvestedArea = currDeposit * this.totalArea;

// получаем объем рубки на основе режима работы с объемом
(фиксированным или динамическим)
var currHarvestedVolume = 0;
if(this.volumeMode == lesoseka.VOLUME_FIXED) currHarvestedVolume
= currHarvestedArea * this.volumePerArea
else if(this.volumeMode == lesoseka.VOLUME_DYNAMIC)
currHarvestedVolume = this.calculateCurrDynamicVolume(objAreas);

//var currHarvestedVolume = currHarvestedArea * this.volumePerArea;

```

```

        harvestPart = harvestVolume / currHarvestedVolume;
        arrHarvestParts.push(harvestPart);
        if(harvestPart > value) value = harvestPart;
    }
    //alert(arrHarvestParts);
    //alert(value);
    return value;
},

showOptimizationResult: function(){
    // выводит график с результатом оптимизации
    // расчет прогноза
    //harvestVolumeEstimated
    //harvestDuration
    //forecastDuration
    //var currVolumeDeposit = this.volumeDeposit;
    var currVolumeDeposit = this.getVolumeDeposit();

    this.arrMatrixAll = new Array();

    this.arrChartData = new Array(); // сюда собираются данные для построения
графиков по разным группам
    for(var i = 0; i < this.arrAgeGroups.length; i++){
        var obj = {};
        if(this.arrAgeGroups[i].isReady()){
            obj.startAge = this.arrAgeGroups[i].startAge;
            obj.endAge = this.arrAgeGroups[i].endAge;
            obj.ageLabel = "";
            if(this.arrAgeGroups[i].ageLabel != undefined) obj.ageLabel =
this.arrAgeGroups[i].ageLabel;
            obj.dataParts = new Array();
            obj.dataArea = new Array();
            this.arrChartData.push(obj);
        }
    }
}

```

```

// формируется первичный массив для итерации
var arrMatrix = new Array();
for(i = 0; i < this.arrGrowthMatrix.length; i++){
    arrMatrix.push(this.arrGrowthMatrix[i].part);
}

this.arrMatrixAll.push(arrMatrix);

var arrResult = new Array();
var harvestPart = this.harvestVolumeEstimated / currVolumeDeposit;
var currDeposit = 0;
this.arrHarvestParts = new Array();
//var currHarvestVolume = 0;
this.arrHarvestParts.push(harvestPart);
var arrHarvestedArea = new Array();
arrHarvestedArea.push(this.harvestedArea);
var arrAreaParts = new Array();
arrAreaParts.push(this.harvestedArea / this.totalArea);
for(i = 1; i <= this.harvestDuration; i++){
    arrResult = this.makeIteration(arrMatrix, harvestPart);
    // агрегация элементов массива по отдельным классам
    var currGroupIndex = 0;
    var valuePart = 0;
    currDeposit = 0;
    var objAreas = {};
    for(var ii = 0; ii < arrResult.length; ii++){
        if(this.arrAgeGroups[currGroupIndex].endAge < ii){
            this.arrChartData[currGroupIndex].dataParts.push(valuePart);
            this.arrChartData[currGroupIndex].dataArea.push(valuePart *
this.totalArea);

            currGroupIndex++;
            valuePart = 0;
        }
        valuePart += arrResult[ii];
    }

```

```

        if(ii >= this.harvestAge){
            // расчет
            currDeposit += arrResult[ii];
            objAreas[ii] = arrResult[ii] * this.totalArea; // сохраняем объем
для этого года роста
        }

    }
    arrMatrix = arrResult;

    this.arrChartData[this.arrChartData.length - 1].dataParts.push(valuePart);
    this.arrChartData[this.arrChartData.length - 1].dataArea.push(valuePart *
this.totalArea);

    var currHarvestedArea = currDeposit * this.totalArea;
    //var currHarvestedVolume = currHarvestedArea * this.volumePerArea;

    // получаем объем рубки на основе режима работы с объемом
(фиксированным или динамическим)
    var currHarvestedVolume = 0;
    if(this.volumeMode == lesoseka.VOLUME_FIXED) currHarvestedVolume
= currHarvestedArea * this.volumePerArea
    else if(this.volumeMode == lesoseka.VOLUME_DYNAMIC)
currHarvestedVolume = this.calculateCurrDynamicVolume(objAreas);

    harvestPart = this.harvestVolumeEstimated / currHarvestedVolume;

    this.arrHarvestParts.push(harvestPart);
    arrHarvestedArea.push(currDeposit * this.totalArea);
    arrAreaParts.push(currDeposit);
}
    //alert(this.arrChartData.length + ' - ' + this.arrChartData[0].dataParts.length + ' - '
+ this.arrChartData[0].dataParts[0]);

```

```

if(this.elOutput == undefined){
    this.elOutput = document.createElement('div');
    //this.elForm.className('lesAgeClassItem');
    this.baseElement.insert({bottom: this.elOutput});
}

var str = '<table><tbody>';
str += '<tr><td>доли изъятия</td><td>' + this.arrHarvestParts.join('</td><td>') +
'</td></tr>';
    //str += '<tr><td>площадь под рубки</td><td>' +
arrHarvestedArea.join('</td><td>') + '</td></tr>';
    //str += '<tr><td>доля площади под рубки</td><td>' +
arrAreaParts.join('</td><td>') + '</td></tr>';

//var boolStopMeaningData = false;
//var boolStopIndexFound = false;
for(i = 0; i < this.arrChartData.length; i++){

    str += '<tr><td>' + this.arrChartData[i].ageLabel + '</td>';

    for(ii = 0; ii < this.arrChartData[i].dataParts.length; ii++){

        var strValue = this.arrChartData[i].dataParts[ii] + ";
        strValue = strValue.replace('.', ',');
        str += '<td>' + strValue + '</td>';

    }

    str += '</tr>';
}
str += '</tbody></table>';

this.elOutput.innerHTML = str;

```

```

        if(this.chartData == null) this.chartData = {};
        this.chartData.chartName = 'Прогноз динамики возрастных групп при
расчетном объеме рубки леса';

        this.setChartMeaningData(); // требует this.arrHarvestParts, this.arrChartData

        this.showChart();
    },

    calculateForecast: function(){
        // расчет прогноза
        //var currVolumeDeposit = this.volumeDeposit;
        var currVolumeDeposit = this.getVolumeDeposit();
        this.arrMatrixAll = new Array();

        this.arrChartData = new Array(); // сюда собираются данные для построения
графиков по разным группам
        for(var i = 0; i < this.arrAgeGroups.length; i++){
            var obj = {};
            if(this.arrAgeGroups[i].isReady()){
                obj.startAge = this.arrAgeGroups[i].startAge;
                obj.endAge = this.arrAgeGroups[i].endAge;
                obj.ageLabel = "";
                if(this.arrAgeGroups[i].ageLabel != undefined) obj.ageLabel =
this.arrAgeGroups[i].ageLabel;
                obj.dataParts = new Array();
                obj.dataArea = new Array();
                this.arrChartData.push(obj);
            }
        }

        // формируется первичный массив для итерации
        var arrMatrix = new Array();
        for(i = 0; i < this.arrGrowthMatrix.length; i++){

```

```

        arrMatrix.push(this.arrGrowthMatrix[i].part);
    }

    this.arrMatrixAll.push(arrMatrix);

    var arrResult = new Array();
    var harvestPart = this.harvestVolume / currVolumeDeposit;
    var currDeposit = 0;
    this.arrHarvestParts = new Array();
    //var currHarvestVolume = 0;
    this.arrHarvestParts.push(harvestPart);
    var arrHarvestedArea = new Array();
    arrHarvestedArea.push(this.harvestedArea);
    var arrAreaParts = new Array();
    arrAreaParts.push(this.harvestedArea / this.totalArea);
    for(i = 1; i <= this.forecastDuration; i++){
        arrResult = this.makeIteration(arrMatrix, harvestPart);
        // агрегация элементов массива по отдельным классам
        var currGroupIndex = 0;
        var valuePart = 0;
        currDeposit = 0;
        var dynamicHarvestedVolume = 0;
        var objAreas = {};
        for(var ii = 0; ii < arrResult.length; ii++){
            if(this.arrAgeGroups[currGroupIndex].endAge < ii){
                this.arrChartData[currGroupIndex].dataParts.push(valuePart);
                this.arrChartData[currGroupIndex].dataArea.push(valuePart *
this.totalArea);

                currGroupIndex++;
                valuePart = 0;
            }
            valuePart += arrResult[ii];
            if(ii >= this.harvestAge){
                // расчет
                currDeposit += arrResult[ii];

```

```

        objAreas[ii] = arrResult[ii] * this.totalArea; // сохраняем объем
для этого года роста

        //alert(ii + ' - ' + arrResult[ii] + ' - ' + this.totalArea + ' - ' +
objAreas[ii]);

        // расчет текущего объема
        //alert(currDeposit + ' - currDeposit');
        //dynamicHarvestedVolume += arrResult[ii] * this.totalArea *
    }

}

arrMatrix = arrResult;

this.arrChartData[this.arrChartData.length - 1].dataParts.push(valuePart);
this.arrChartData[this.arrChartData.length - 1].dataArea.push(valuePart *
this.totalArea);

var currHarvestedArea = currDeposit * this.totalArea;
//var currHarvestedVolume = currHarvestedArea * this.volumePerArea;

// получаем объем рубки на основе режима работы с объемом
(фиксированным или динамическим)
var currHarvestedVolume = 0;
if(this.volumeMode == lesoseka.VOLUME_FIXED) currHarvestedVolume
= currHarvestedArea * this.volumePerArea
else if(this.volumeMode == lesoseka.VOLUME_DYNAMIC)
currHarvestedVolume = this.calculateCurrDynamicVolume(objAreas);

harvestPart = this.harvestVolume / currHarvestedVolume;

this.arrHarvestParts.push(harvestPart);
arrHarvestedArea.push(currDeposit * this.totalArea);
arrAreaParts.push(currDeposit);
}

//alert(this.arrChartData.length + ' - ' + this.arrChartData[0].dataParts.length + ' - '
+ this.arrChartData[0].dataParts[0]);

```



```

if(this.elOutput == undefined){
    this.elOutput = document.createElement('div');
    //this.elForm.addClassName('lesAgeClassItem');
    this.baseElement.insert({bottom: this.elOutput});
}

var str = '<table><tbody>';
str += '<tr><td>доли изъятия</td><td>' + this.arrHarvestParts.join('</td><td>') +
'</td></tr>';
    //str += '<tr><td>площадь под рубки</td><td>' +
arrHarvestedArea.join('</td><td>') + '</td></tr>';
    //str += '<tr><td>доля площади под рубки</td><td>' +
arrAreaParts.join('</td><td>') + '</td></tr>';

//var boolStopMeaningData = false;
//var boolStopIndexFound = false;
for(i = 0; i < this.arrChartData.length; i++){

    str += '<tr><td>' + this.arrChartData[i].ageLabel + '</td>';

    for(ii = 0; ii < this.arrChartData[i].dataParts.length; ii++){

        var strValue = this.arrChartData[i].dataParts[ii] + ";
        strValue = strValue.replace('.', ',');
        str += '<td>' + strValue + '</td>';

    }

    str += '</tr>';
}
str += '</tbody></table>';

```

```

this.elOutput.innerHTML = str;

if(this.chartData == null) this.chartData = {};
this.chartData.chartName = 'Прогноз динамики возрастных групп при
планируемом объеме рубки леса';

this.setChartMeaningData(); // тpeбyет this.arrHarvestParts, this.arrChartData

this.showChart();
},

setChartMeaningData: function(){
    var i = 0;
    this.arrChartMeaningData = new Array();
    var lastMeaningDataIndex = this.arrHarvestParts.length;
    for(i = 0; i < this.arrHarvestParts.length; i ++){
        if(this.arrHarvestParts[i] < 0){
            lastMeaningDataIndex = i;
            i = this.arrHarvestParts.length;
        }
    }
    //lastMeaningDataIndex --;
    for(i = 0; i < this.arrChartData.length; i ++){
        //var arrTemp = new Array();
        var objTemp = {};
        for(var name in this.arrChartData[i]){
            objTemp[name] = this.arrChartData[i][name];
            //strTrace += name + ' = ' + objTemp[name] + "\n";
        }
        objTemp.dataParts = new Array();
        for(ii = 0; ii < lastMeaningDataIndex; ii ++){
            objTemp.dataParts.push(this.arrChartData[i].dataParts[ii]);
        }
        this.arrChartMeaningData.push(objTemp);
    }
}

```

```

    }
},

showChart: function(){
    // выводит график
    //alert(obj.elementName)
    if(this.chartData == null) this.chartData = {};
    if(this.chartData.elementName == null) this.chartData.elementName = 'elChart';
    if(this.chartData.chartName == null) this.chartData.chartName = 'Название
графика';
    if(this.chartData.yAxesName == null) this.chartData.yAxesName = 'Доля общей
площади';
    if(this.chartData.xAxesName == null) this.chartData.xAxesName = 'Год';

    //alert(obj.elementName)
    if(this[this.chartData.elementName] == undefined){
        this[this.chartData.elementName] = document.createElement('div');
        this.baseElement.insert({bottom: this[this.chartData.elementName]});
    }

    this.chartData.series = new Array();

    var strTrace = "";
    for(var i = 0; i < this.arrChartMeaningData.length; i ++){
        this.chartData.series.push({name: this.arrChartMeaningData[i].ageLabel,
data: this.arrChartMeaningData[i].dataParts});
        strTrace += this.arrChartMeaningData[i].ageLabel + " - " +
this.arrChartMeaningData[i].dataParts;
    }
    //alert(strTrace);

    this.chartData.series.reverse();
    if(this.chartData.series == null){
        this.chartData.series = new Array({
            name: 'Asia',

```

```

        data:
[0.18724685108289496,0.1972045869456774,0.20714548126700338,0.21706887510411532
,0.22697412437726547,0.23686059963834943]
        }, {
            name: 'Africa',
            data:
[0.17508236057929916,0.17291785788765046,0.170779137013777,0.16866589877159205,
0.16657784738054948,0.1645146904274506]
        }, {
            name: 'Europe',
            data:
[0.5529573932197317,0.534193804290437,0.5161500202322776,0.4987957721707895,0.48
21020987249568,0.4660412773646219]
        }, {
            name: 'America',
            data:
[0.08471339511807432,0.09568375087623551,0.10592536148694272,0.1154694539535027
1,0.12434592951722807,0.1325834325695779]
        });
    }

```

```

var chart = new Highcharts.Chart({
    chart: {
        renderTo: this[this.chartData.elementName],
        type: 'area'
    },
    title: {
        text: this.chartData.chartName
    },
    subtitle: {
        //text: 'Source: Wikipedia.org'
        text: ""
    },
    xAxis: {
        categories: [],

```

```

        tickmarkPlacement: 'on',
        title: {
            text: this.chartData.xAxesName
            //enabled: false
        }
    },
    yAxis: {
        title: {
            text: this.chartData.yAxesName
        },
        labels: {
            formatter: function() {
                return this.value;
            }
        }
    },
    tooltip: {
        shared: true,
        //valueSuffix: ' %'
        valueSuffix: ""
    },
    plotOptions: {
        area: {
            stacking: 'normal',
            lineColor: '#666666',
            lineWidth: 0.5,

            marker: {
                enabled: false,
                lineWidth: 0.5,
                lineColor: '#666666',
                states: {
                    hover: {
                        enabled: true
                    }
                }
            }
        }
    }

```

```

        }
    }

    },
    series: this.chartData.series
});
},

```

```

makeIteration: function(arrMatrix, harvestPart){
    // выполняет расчет очередной итерации
    // arrMatrix - матрица площадей для возрастов
    // harvestPart - доля изъятия
    var arrResult = new Array();
    //for(var i = 1; i < arrMatrix.length; i++){
    var value = 0;
    var harvested = 0; // вырубленная доля
    var died = 0;
    for(var i = 0; i < (arrMatrix.length - 2); i++){
        var currDeath = this.arrGrowthMatrix[i].death;
        value = arrMatrix[i] - arrMatrix[i] * currDeath;
        died += arrMatrix[i] * currDeath;
        if(i >= this.harvestAge){
            // рубка началась
            value -= arrMatrix[i] * harvestPart;
            harvested += arrMatrix[i] * harvestPart;
        }
        //var value =
        arrResult.push(value);
    }
    // расчет последнего года
    var lastIndex = arrMatrix.length - 1;
    var prevIndex = lastIndex - 1;
    var lastDeath = this.arrGrowthMatrix[lastIndex].death;
    var prevDeath = this.arrGrowthMatrix[prevIndex].death;

```

```

        // (arrMatrix[lastIndex] - arrMatrix[lastIndex] * harvestPart) - невырубленная
        часть последнего года (последний возрастной класс агрегирует ставшие старше на 1
        год деревья принадлежавшие последнему классу, накапливает разновозрастные груаа)
        value = arrMatrix[prevIndex] + (arrMatrix[lastIndex] - arrMatrix[lastIndex] *
        harvestPart) - (arrMatrix[lastIndex] * lastDeath) - (arrMatrix[prevIndex] * prevDeath) -
        (arrMatrix[prevIndex] * harvestPart);
        arrResult.push(value);
        harvested += arrMatrix[lastIndex] * harvestPart + arrMatrix[prevIndex] *
        harvestPart;
        died += arrMatrix[lastIndex] * lastDeath + arrMatrix[prevIndex] * prevDeath;

        // расчет нулевого года
        // общая гибель + общая рубка
        value = died + harvested;
        arrResult.unshift(value);
        return arrResult;
    },

```

```

    butAddAgeClassHandler: function(e){
        var lesosekaObj = lesoseka.getObject(e.target);
        //lesosekaObj.setGrowthMatrixForm();
        if(lesosekaObj.arrAgeGroups.length > 0) var ageClass = new
        lesoseka.ageClass(lesosekaObj.arrAgeGroups[lesosekaObj.arrAgeGroups.length - 1]);
        else var ageClass = new lesoseka.ageClass();
        lesosekaObj.elAgeClassesList.insert({bottom: ageClass.elForm});
        lesosekaObj.arrAgeGroups.push(ageClass);
    },

```

```

    butSetGrowthMatrixHandler: function(e){
        var lesosekaObj = lesoseka.getObject(e.target);
        lesosekaObj.setGrowthMatrixForm();
        //lesoseka.setGrowthMatrixForm(e.target);
    },

```

```

    changeDeath: function(e){

```

```

        // сохраняет изменения значения смертности для конкретного возрастного
        класса

        //matrixEntered
        var index = parseInt(e.target.readAttribute('index'));
        var lesosekaObj = lesoseka.getObject(e.target);
        lesosekaObj.arrGrowthMatrix[index].death = parseFloat(e.target.value);
        if(lesosekaObj.matrixEntered.death == undefined)
lesosekaObj.matrixEntered.death = {};
            lesosekaObj.matrixEntered.death[lesosekaObj.arrGrowthMatrix[index].name] =
lesosekaObj.arrGrowthMatrix[index].death;
        },

        changeArea: function(e){
            // вносит изменение в массив данных о занимаемой возрастными классами
            площади и рассчитывает долю этого гласса в общей площади
            var index = parseInt(e.target.readAttribute('index'));
            var lesosekaObj = lesoseka.getObject(e.target);
            lesosekaObj.arrAgeClasses[index].area = parseFloat(e.target.value);
            if(lesosekaObj.matrixEntered.area == undefined) lesosekaObj.matrixEntered.area
            = {};
            lesosekaObj.matrixEntered.area[lesosekaObj.arrAgeClasses[index].name] =
lesosekaObj.arrAgeClasses[index].area;
            lesosekaObj.setAreaParts();
        },

        setAreaParts: function(){
            // рассчитывает доли площади на основе внесенных единиц площади
            var areaSum = 0;
            var part;
            var element;
            //this.elPartRow.setOpacity(0.5);
            for(var i = 0; i < this.arrAgeClasses.length; i++){
                areaSum += this.arrAgeClasses[i].area;
            }
        }
    }

```



```

        for(i = 0; i < this.arrAgeClasses.length; i++){
            part = this.arrAgeClasses[i].area / areaSum;
            this.arrAgeClasses[i].part = part;
            element = this.elPartRow.select('[index="' + i + "']")[0];
            element.innerHTML = part;
        }
    },

    changePart: function(e){

    },

    changeValue: function(e){
        // сохранение в переменную программы введенного значения
        var lesosekaObj = lesoseka.getObject(e.target);
        var variable = e.target.readAttribute('var');
        var datatype = lesosekaObj.dataTypes[variable];
        //alert(variable + " = " + e.target.value);
        switch(datatype){
            case 'int_unsigned':
                lesosekaObj[variable] = parseInt(e.target.value);
                break;
            case 'float_unsigned':
                lesosekaObj[variable] = parseFloat(e.target.value);
                break;
            case 'string':
                lesosekaObj[variable] = e.target.value;
                break;
        }
    },

    // обработчики введения данных "на лету"
    filterInput: function(e){
        //alert('filterInput');
        Event.stop(e);
    }
}

```

```

var lesosekaObj = lesoseka.getObject(e.target);
var variable = e.target.readAttribute('var');
var datatype = lesosekaObj.dataTypes[variable];
//alert(datatype);
switch(datatype){
    case 'int_unsigned':
        e.target.value = wcData.strIntValueUnsigned(e.target.value);
        break;
    case 'float_unsigned':
        e.target.value = wcData.strFloatValueUnsigned(e.target.value);
        break;
    case 'string':
        //e.target.value =
        break;
}
//alert(datatype);

},

filterInt: function(e){
    //alert(e.target);
    // фильтр ввода целочисленных значений
    Event.stop(e);
    e.target.value = wcData.strIntValueUnsigned(e.target.value);
},

filterFloat: function(e){
    // фильтр ввода значений float
    Event.stop(e);
    e.target.value = wcData.strFloatValueUnsigned(e.target.value);
},

```

```
});
```

```
////////////////////////////////////
```

```
// статические методы класса
```

```
lesoseka.getObject = function(element){
```

```
    // возвращает экземпляр класса лесосеки на основе элемента обрабатываемого  
    DOM
```

```
    var arrId = parseInt(element.up('div.lesoseka').readAttribute('arr'));
```

```
    return this.lesosekaObjects[arrId];
```

```
}
```

```
lesoseka.ini = function(e){
```

```
    // подключает класс к DOM
```

```
    this.lesosekaElements = document.documentElement.select('.lesoseka');
```

```
    this.lesosekaObjects = new Array();
```

```
    for(var i = 0; i < this.lesosekaElements.length; i++){
```

```
        this.lesosekaElements[i].writeAttribute('arr', i);
```

```
        var obj = new lesoseka(this.lesosekaElements[i]);
```

```
        this.lesosekaObjects.push(obj);
```

```
    }
```

```
}
```

```
////////////////////////////////////
```

```
// константы класса
```

```
lesoseka.VOLUME_FIXED = 1;
```

```
lesoseka.VOLUME_DYNAMIC = 2;
```

```
lesoseka.FORM_NEW_LESOSEKA = 1;
```

```
lesoseka.FORM_OLD_LESOSEKA = 2;
```

```
////////////////////////////////////
```

```
// внутренние классы класса lesoseka
```

```
lesoseka.ageClass = Class.create({
```

```

initialize: function(prevAgeClass){
    //alert('create new age class');
    if(lesoseka.ageClass.iterationStep == undefined) this.iterationStep = 1;
    else this.iterationStep = lesoseka.ageClass.iterationStep;

    if(prevAgeClass == undefined) {
        this.id = 0;
        this.startAge = 0;
    }
    else{
        this.id = prevAgeClass.id + 1;
        this.startAge = "";
        if(!isNaN(prevAgeClass.endAge)) this.startAge = prevAgeClass.endAge +
this.iterationStep;

    }

    this.elForm = document.createElement('tr');
    this.elForm.className('lesAgeClassItem');
    this.code = '<td><div _var="startAge" title="Нижний предел группы возраста,
лет">' + this.startAge + '</div ></td><td><div var="endAge" title="Верхний предел
группы возраста, лет" index="' + this.id + '" _class=""></div></td><td><div var="ageArea"
index="' + this.id + '" _class="" class="lesAgeClassArea"></div></td><td><div
var="ageLabel" index="' + this.id + '" class="lesAgeLabelCell"
_class=""></div></td><td><input type="button" title="Удалить группу возраста"
action="kill" class="lesKillBut" value="X" index="' + this.id + '"></td></tr>';

    this.elForm.innerHTML = this.code;
    this.elStartAge = this.elForm.select('div[_var="startAge"]')[0];
    this.elEndAge = this.elForm.select('div[var="endAge"]')[0];
    this.elAgeArea = this.elForm.select('div[var="ageArea"]')[0];
    this.elAgeLabel = this.elForm.select('div[var="ageLabel"]')[0];
    this.elKillBut = this.elForm.select('input[action="kill"]')[0];
    //this.elEndAge = this.elForm.select('div[_var="startAge"]');
},

```

```

//area: 0,
//ageArea: 0,
iterationStep: 1,

getElement: function(){
    return this.elForm;
},
setStartAge: function(value){
    this.startAge = value + this.iterationStep;
    this.elStartAge.innerHTML = this.startAge;
},
setEndAge: function(value){
    this.endAge = parseFloat(value);
    //если введенный верхний возраст не укладывается в указанное значение, то
    оно корректируется таким образом, чтобы получить целочисленное количество
    итераций
    var range = this.endAge - this.startAge;
    this.iterations = Math.round(range / this.iterationStep);
    if((range % this.iterationStep) != 0){
        this.endAge = this.startAge + this.iterationStep * this.iterations;
    }

    this.iterations ++;

    if(this.endAge <= this.startAge) this.endAge = this.startAge + this.iterationStep;
    this.elEndAge.innerHTML = this.endAge;
    //alert(this.endAge + ' - ' + this.elEndAge);
    return this.endAge;
},
setArea: function(value){
    this.ageArea = parseFloat(value);
},
getArea: function(){
    if(this.ageArea == undefined) return false;

```

```

        else return this.ageArea;
    },
    setLabel: function(value){
        this.ageLabel = value;
    },
    setIterationStep: function(value){
        this.iterationStep = parseFloat(value);
    },
    getIterations: function(){
        if(this.iterations == undefined) return false;
        else return this.iterations;
    },
    getIterationArea: function(){
        //alert(this.iterations);
        if(this.iterations == undefined) return false;
        if(this.ageArea == undefined) return false;
        var iterationArea = this.ageArea / this.iterations;
        return iterationArea;
    },
    setIndex: function(id){
        this.id = id;
        this.elStartAge.writeAttribute('index', this.id);
        this.elEndAge.writeAttribute('index', this.id);
        this.elAgeArea.writeAttribute('index', this.id);
        this.elAgeLabel.writeAttribute('index', this.id);
        this.elKillBut.writeAttribute('index', this.id);
    },
    isReady: function(){
        if(this.ageArea != undefined && this.endAge != undefined) return true;
        else return false;
    },
});
lesoseka.ageClass.iterationStep = 1;

```

// вспомогательный класс для работы с элементами

```

var wcElements = Class.create({
    initialize: function(element) {
    }
});

wcElements.setTextInput = function(e){
    // заменяет содержимое контейнера текстовым элементом ввода.
    var element = e.findElement('[var]');
    if(element == undefined) return false;
    if(element.nodeName == 'INPUT') return false;
    var value = element.innerHTML;
    var dimensions = element.getDimensions();
    var arrAttr = new Array();
    for(var i = 0; i < element.attributes.length; i++){
        // считывание атрибутов
        if(element.attributes[i].name != 'class' && element.attributes[i].name != 'style'){
            if(element.attributes[i].name == '_class') arrAttr.push('class="' +
element.attributes[i].value + '"');
            else arrAttr.push(element.attributes[i].name + '=' +
element.attributes[i].value + '"');
        }
    }
    // определение параметра padding контейнера
    var paddingLeft = element.getStyle('paddingLeft');
    var paddingRight = element.getStyle('paddingRight');
    var paddingTop = element.getStyle('paddingTop');
    var paddingBottom = element.getStyle('paddingBottom');

    element.innerHTML = '<input type="text" ' + arrAttr.join(' ') + ' value="' + value + '">';
    var elInputText = element.firstDescendant();
    var dimensions2 = element.getDimensions();
    var dimensionsInput = elInputText.getDimensions();
    var heightDif = dimensionsInput.height - (dimensions2.height - dimensions.height);

```

```

        element.setAttribute('_padding', paddingTop + '' + paddingRight + '' +
paddingBottom + '' + paddingLeft);
        element.setStyle({padding: '0px'});

        elInputText.setStyle({
            width: dimensions.width + 'px',
            height: (dimensionsInput.height - (dimensions2.height - dimensions.height) - 4) +
'px',
        });

        elInputText.focus();
        elInputText.observe('blur', wcElements.unsetTextInput);
        elInputText.observe('keyup', wcElements.checkEnterTextInput);
        return elInputText;
    }

```

```

wcElements.unsetTextInput = function(e){
    var element = e.target;
    var parent = element.up();
    var parent = element.up();
    var value = element.value;
    element.purge();
    element.replace(value);
    parent.setStyle({padding: parent.readAttribute('_padding')});
    parent.setAttribute('_padding', false);
}

```

```

wcElements.checkEnterTextInput = function(e){
    // проверка нажатия клавиши ENTER
    if(e.keyCode == 13) wcElements.unsetTextInput(e);
}

```

```

// вспомогательный класс для работы с данными(будет вынесен в отдельный файл)
var wcData = Class.create({
    initialize: function(element) {

```



```

    }
});

wcData.array = function(obj){
    if(Object.isArray(obj)) return obj;
    else if(Object.isString(obj)) return obj.split(' ');
}

wcData.strFirstUppercase = function(str){
    var firstChar = str.charAt(0);
    return firstChar.toUpperCase() + str.substring(1);
}

wcData.strIntValueUnsigned = function(str){
    // проверяет строковую переменную на соответствие формату Integer unsigned
    var str2 = "";
    var charCode;
    var char_;
    for(var i = 0; i < str.length; i++){
        charCode = str.charCodeAt(i);
        char_ = str.charAt(i);
        if(charCode >= 48 && charCode <= 57) str2 += char_;
    }
    return str2;
}

wcData.strFloatValueUnsigned = function(str){
    // проверяет строковую переменную на соответствие формату Float unsigned
    var str2 = "";
    var pointCounter = 0;
    var charCode;
    var char_;
    var firstChar = str.charAt(0);
    if(firstChar == "." || firstChar == ",") {
        str = "0." + str.substring(1);
    }

```

```

        firstChar = "0";
    }
    var secondChar = str.charAt(1);
    if(firstChar == "0" && secondChar == "0") str = "0" + str.substring(2);
    for(var i = 0; i < str.length; i++){
        charCode = str.charCodeAt(i);
        char_ = str.charAt(i);
        if(charCode >= 48 && charCode <= 57) str2 += char_;
        else if(charCode == 46 || charCode == 44){
            if(pointCounter == 0) {
                str2 += ".";
                pointCounter++;
            }
        }
    }
    return str2;
}

```

```

wcData.strFloatValueSigned = function(str){
    // преобразует строку в соответствующую float signed
    var str2 = "";
    var pointCounter = 0;
    var charCode;
    var char_;
    var firstChar = str.charAt(0);
    var secondChar = str.charAt(1);
    if(str == "-") return str;
    if(firstChar == "." || firstChar == ",") {
        str = "0." + str.substring(1);
        firstChar = "0";
    }
    else if(firstChar == "-" && (secondChar == "." || secondChar == ",")) {
        str = "-0." + str.substring(2);
    }
    var firstChar = str.charAt(0);

```

```

var secondChar = str.charAt(1);
var thirdChar = str.charAt(2);
var from = 0;
if(firstChar == "-"){
    if(secondChar == "0" && thirdChar == "0") str = "-0" + str.substring(3);
    from = 1;
    str2 += "-";
}
else{
    if(firstChar == "0" && secondChar == "0") str = "0" + str.substring(2);
}
for(var i = from; i < str.length; i++){
    charCode = str.charCodeAt(i);
    char_ = str.charAt(i);
    if(charCode >= 48 && charCode <= 57) str2 += char_;
    else if(charCode == 46 || charCode == 44){
        if(pointCounter == 0) {
            str2 += ".";
            pointCounter++;
        }
    }
}
return str2;
}

```

```

Event.observe(window, 'load', function() { lesoseka.ini() });

```