

Автоматизация процессов подготовки управленческих решений в лесной отрасли

А. К. ПОПОВА

Институт динамики систем и теории управления СО РАН, Иркутск, Россия

e-mail: chudnenko@icss.ru

A software system developed for prediction of state of the forest resources is considered. The software tools allowing users to make their own applications are described.

Введение

Формирование политики использования лесных ресурсов является чрезвычайно важной задачей для лиц, принимающих решения (ЛПР) в управлении лесопромышленным регионом. Эффективность принимаемых решений в первую очередь зависит от объема, вида и качества исходных данных о состоянии лесных ресурсов, а также прогнозов развития лесных ресурсов в зависимости от принимаемых решений (политики заготовки лесных ресурсов). В связи с этим возникла необходимость создания информационно-аналитических систем поддержки принятия решения в области управления лесными ресурсами и прогнозирования их динамики. Такая система позволит удовлетворить постоянно возрастающие требования к качеству информации о состоянии лесных ресурсов в текущий момент и на перспективу путем повышения ее достоверности и надежности, а также оперативности получения.

Рассматриваемое исследование посвящено разработке инструментальных средств, направленных на решение задач автоматизации поддержки принятия решений по рациональному использованию лесных ресурсов, функционирующих на основе современных средств обработки пространственно-распределенной информации — геоинформационных систем (ГИС). В основе подхода к анализу возможных принимаемых решений лежит изучение свойств исследуемой природной системы в будущем при некотором допустимом решении ЛПР, т.е. анализ прогнозов динамики исследуемого объекта при помощи соответствующего математического и программного обеспечения, реализованного в рамках некоторой системы поддержки принятия решений (СППР). Это человеко-машинный объект, который позволяет лицам, принимающим решения, использовать разнообразные методы, данные, знания, объективные и субъективные модели для анализа и решения слабоструктурированных и неструктурированных задач. Идея СППР возникла как попытка автоматизации естественных человеческих действий по анализу имеющейся информации, планированию действий и т.п. с целью решения конкретной задачи. В данном случае СППР строится на основе анализа результатов расчета прогноза динамики лесных ресурсов по соответствующим математическим моделям.

1. Программный комплекс прогнозирования состояния лесных ресурсов

Программный комплекс предназначен для прогнозирования состояния лесных ресурсов на основе приложения системы математических моделей [1, 2] к конкретному природному объекту и некоторой политики использования лесных ресурсов, заданной набором параметров модели. Прогнозы состояния лесных ресурсов рассчитываются по соответствующим моделям в зависимости от масштаба природного объекта и решаемой задачи. Каждый прогноз — это модельный сценарий, задаваемый комбинацией параметров модели. Сценарии, предоставляемые ЛПР для дальнейшего анализа, определяются по совокупности предварительно заданных критериев из всего набора прогнозов.

Основными особенностями программного комплекса являются:

- возможность управления функционированием некоторых частей комплекса, например отображением графиков при помощи управляющих программ (скриптов);
- предоставление пользователю формализованного логического языка запросов для получения данных из БД, управления ходом расчетов.

Программный комплекс объединяет три подсистемы: ГИС, подсистему математического моделирования и систему автоматизации логических рассуждений (систему искусственного интеллекта) [3, 4].

Подсистема математического моделирования предназначена для прогнозирования состояния лесных ресурсов на основе системы математических моделей. Источником информации для подсистемы математического моделирования являются базы данных, в том числе базы данных ГИС. Пространственно-распределенные данные требуют обеспечения тесного взаимодействия с современными ГИС, используемыми, в частности, для представления результатов прогнозных расчетов в виде цифровых карт. Важной составляющей СППР является подсистема пользовательского интерфейса с ЛПР, которая позволяет получить необходимые для расчета начальные данные, критерии анализа результатов, а также предоставляет результаты моделирования в графическом виде.

Подсистема искусственного интеллекта позволяет автоматизировать процесс построения математической модели природного объекта. Ее средствами осуществляется выбор из базы моделей такой математической модели, которая наиболее соответствует задаче прогнозирования и рангу исследуемого объекта. Подсистема искусственного интеллекта реализована на основе базы знаний о базовой структуре каждой модели, об идентификации модели на основе данных об исследуемом объекте, а также поиска начальных условий модели. Правила базы знаний позволяют гибко определять такие антропогенные воздействия на лесные ресурсы, как объемы рубок, насаждений в зависимости от набора параметров. Например, объем рубок изменяется на каждом шаге интервала прогнозирования отдельно для каждого моделируемого участка в зависимости от расчетных данных.

Программный комплекс предоставляет инструментальные средства, функционирующие на основе современных ГИС, предназначенные для разработки проблемно-ориентированных прикладных ГИС-пакетов поддержки принятия решения для ЛПР (стандартные средства хранения, отображения и обработки пространственно-распределенных данных). Система также позволяет отображать информацию, полученную в результате математического моделирования динамики лесных ресурсов, в виде таблиц, диаграмм, картографических произведений, а также представляет динамику ресурса в виде ани-

мации. Построение картографических материалов осуществляется средствами подсистемы ГИС.

Для разработки программного комплекса использован язык программирования Java, который обладает встроенной поддержкой работы в компьютерных сетях, возможностью переносимости программ между программно-аппаратными платформами, мощными стандартными библиотеками. Кроме того, приложения, реализованные специальным образом, встраиваются в HTML-страницу в виде так называемого апплета. Архитектура программного комплекса представлена на рис. 1.

Подсистема математического моделирования представлена блоком численных расчетов и базой моделей, подсистема пользовательского интерфейса — блоками визуализации результатов, интерфейса, запросной подсистемой, а подсистема искусственного интеллекта — блоками параметрической идентификации, базой знаний. Кроме стандартных Java-библиотек (AWT, Swing и др.) используются библиотеки JFreeChart, OpenMap, tuProlog, Rhino, Seppia.

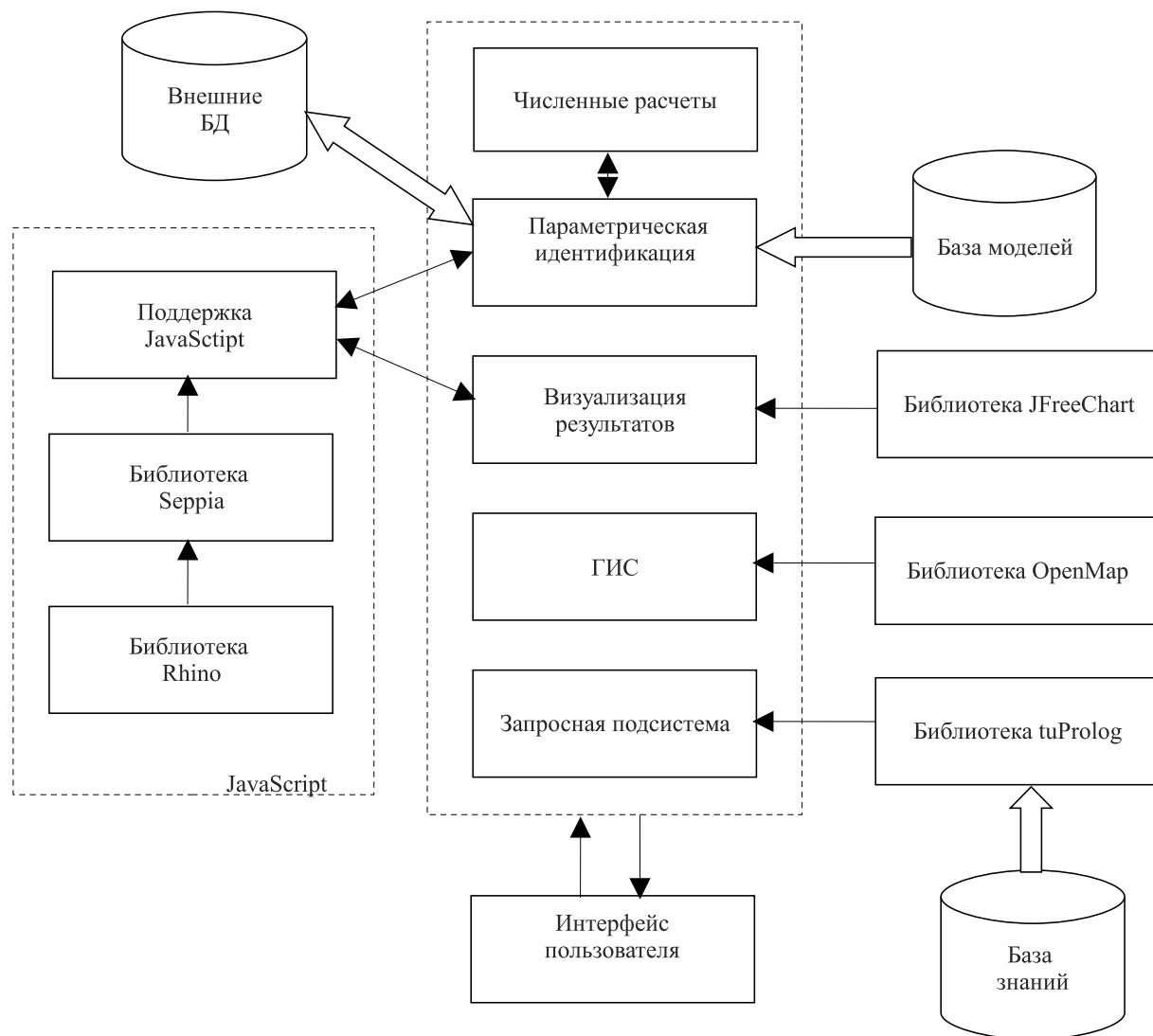


Рис. 1. Архитектура программного комплекса

Блок интерфейса пользователя взаимодействует с основными блоками программного комплекса, передавая им начальные данные об объекте моделирования и поставленной задаче, получая от них результаты расчетов в разных форматах. На основе полученных данных запросной подсистемой осуществляется подбор модели. Исходные данные для идентификации модели и вычисления начальных условий загружаются из баз данных. Запросы к базам данных генерируются запросной подсистемой в процессе построения формализованного представления модели конкретного исследуемого объекта. На основе дальнейшего анализа задачи и выбранной модели блоком параметрической идентификации устанавливаются множество параметров модели и начальные условия. Комбинации значений параметров отображают управляющее воздействие на объект (гипотетическое решение ЛПР). Далее запросной подсистемой осуществляется построение выбранной модели: определяется последовательность смены состояний участка территории в зависимости от рассчитанных параметров модели. Прогнозные расчеты проводятся блоком численных расчетов, результатные данные передаются в блок визуализации результатов и ГИС.

2. Инструментальные средства программного комплекса

Базовые блоки программного комплекса позволяют рассчитывать динамику лесных ресурсов. Описание задачи прогнозирования, объекта моделирования, задание критериев анализа результатов производятся в процессе диалога с подсистемой пользовательского интерфейса. Однако при решении некоторых задач пользователю может потребоваться расширение возможностей программного комплекса. Для этого в его состав включены инструментальные средства, позволяющие пользователям создавать на его основе свои приложения.

Механизм программирования пользовательских приложений реализован при помощи интерпретатора языка программирования JavaScript и специальной библиотеки jsDUD, что позволяет интегрировать множество базовых функций и объектов в рамках одного приложения. Библиотека реализована в виде модуля СППР, она предоставляет доступ к различным функциям программы, например к построению графиков. Блок работы со скриптами взаимодействует с пользователем не через основной пользовательский интерфейс программного комплекса, его процедуры вызываются непосредственно скриптом на языке JavaScript. В программу-приложение JavaScript должна включаться загрузка модуля jsDUD, который непосредственно производит расчет по математической модели заданного сценария. Рассчитанные значения результата доступны через функции модуля jsDUD.

Модуль jsDUD позволяет выполнять расчет сценариев модели динамики управления древостоем и конструирование графиков и диаграмм по результатам расчетов, реализованное на основе библиотеки JFreeChart. Пользовательский интерфейс программы строится на основе библиотеки Swing, доступ к которой также обеспечен через средства JavaScript. Взаимодействие Java-модуля и интерпретатора JavaScript осуществляется с помощью библиотек Rhino и Serpia. Serpia — оболочка для создания и использования Java-приложений, позволяющая интерпретатору JavaScript управлять приложением, при этом программный код JavaScript использует библиотеки Java (jar-архивы).

Интерфейс прикладного программирования (API — Application Programming Interface) модуля расчетов jsDUD представлен следующими функциями:

— *public double[] getData(String lesh, String porod, String kl)* — возвращает массив

данных по лесам лесхоза *lesh* (породам *porod*, классу возраста *kl*). Параметры функции являются строковыми переменными;

- `public void defineData(String name, double[] ar)` — добавляет к графику массив данных *ar* с названием (заголовком, отображаемым на легенде графика) *st*;
- `public void makeMap(int year, String por, String vozr)` — создает карту для лесов на заданный год моделирования *year* (по породам *por*, классу возраста *vozt*);
- `public void CalcDUD()` — запускает расчет по модели с учетом указанных параметров.

Функция `CalcDUD()` использует следующие переменные модуля `jsDUD`:

- `int time` — количество лет интервала моделирования, целое число;
- `String lesh` — название лесхоза, для которого необходимо произвести расчет, строка;
- `boolean mlt` — логическая переменная — указывает, как будет производиться расчет — по одному конкретному лесхозу (*false*) или по всем (*true*), значение по умолчанию — *false*;
- `String uh, gp, pz` — соответственно уровни проведения рубок ухода, рубок главного пользования и пожаров, строковая переменная со значениями *low*, *middle*, *high*.

Рассмотрим пример приложения JavaScript. Вначале необходимо произвести инициализацию модулей:

```
// модуль для формы
var frame = Packages.java.swing.JFrame
// метки полей в интерфейсе пользователя
var label = Packages.java.swing.JLabel
// модуль расчетов
var jsc = Packages.first.jsDUD
//массив для хранения результатов
var ar = java.lang.reflect.Array.newInstance(java.lang.Double.TYPE, 31);
var ar1 = java.lang.reflect.Array.newInstance(java.lang.Double.TYPE, 31);
var ar2 = java.lang.reflect.Array.newInstance(java.lang.Double.TYPE, 31);
```

Основной текст скрипта выглядит следующим образом:

```
// создаем форму
var fr = new frame();
// создаем метку интерфейса
var lb = new label();
// создаем экземпляр модуля расчетов
var js = new jsc();
// задаем величину интервала моделирования
js.time = 30;
// задаем лесхоз
js.lesh = "Марковский";
// уровень рубок ухода --- высокий
js.uh = "high";
// уровень рубок главного пользования --- высокий
js.gp = "high";
// уровень пожаров --- низкий
```

```

js.pz = 'low';
// запускаем расчет по заданным данным
js.CalcDUD();
// добавляем к графику данные по молоднякам 1 класса сосны
// Марковского лесхоза
js.defineData('Марковский, сосна, молодняки 1кл',
    js.getData('Марковский', 'сосна', 'молодняки 1кл'));
// получаем данные по приспевающим елям Марковского лесхоза
ar1 = js.getData('Марковский', 'ель', 'приспевающие');
// получаем данные по молоднякам 2 класса елей Марковского лесхоза
ar2 = js.getData('Марковский', 'ель', 'молодняки 2кл');
// создаем массив со значениями разницы приспевающих и молодняков 2 класса
// ели Марковского лесхоза
for (i = 0; i<30; i++)
    ar[i] = ar1[i]-ar2[i];
// добавляем полученный массив к графику
js.defineData('Марковский, ель, приспевающие-молодняки 2 кл', ar);
// создаем объект изображение
var image = js.chart();
//отображаем график на метку
lb.setIcon(new Packages.java.swing.ImageIcon(image));
// добавляем метку на форму
fr.getContentPane().add(lb);
//показываем форму
fr.show();

```

Возможность программирования скрипта (рис. 2) позволяет комбинировать различные сочетания временных рядов, подбирать их внешний вид (например, цвета), а также создавать интерфейс, удобный для конкретной задачи. Данный механизм позволяет управлять не только выводом графиков, но и такими функциями СППР, как расчеты по моделям, построение цифровых карт в ГИС.

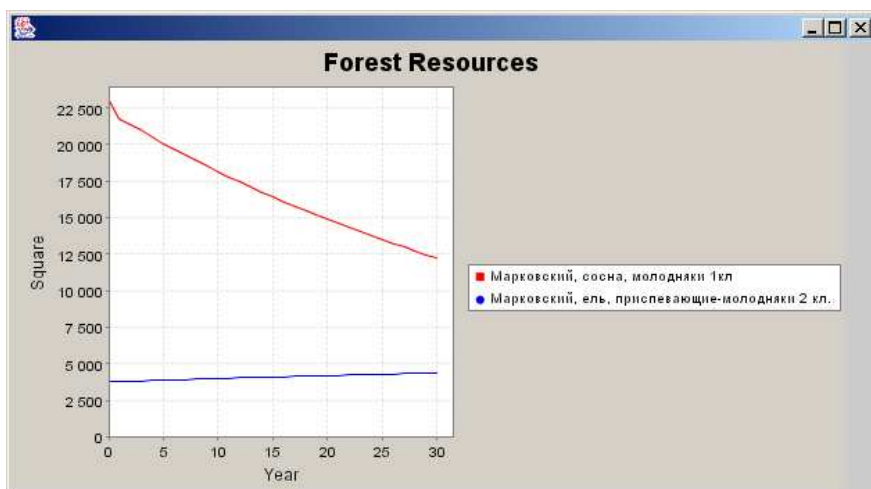


Рис. 2. Результат работы скрипта

Таким образом, модуль программирования JavaScript дает возможность на основе Java-библиотеки jsDUD разработчикам создавать собственные варианты программного обеспечения для моделирования лесных ресурсов.

3. Интеллектуальная подсистема параметрической идентификации моделей

При большом разнообразии исходных данных, математических моделей и их параметров важно так организовать систему, чтобы она могла гибко настраиваться на проведение конкретных расчетов и специфику исходных данных. Поэтому в состав программной системы включена интеллектуальная подсистема, реализованная при помощи логического языка программирования Пролог. Она содержит базу знаний (БЗ) о структурной и параметрической идентификации моделей динамики лесных ресурсов, реализует вопросно-ответную подсистему, при помощи которой можно делать запросы к внутренней структуре модели и результатам компьютерного моделирования.

Подсистема языка Пролог встроена в программную систему при помощи Java-библиотеки tuProlog 1.3, которая позволяет любые сущности Java (объекты, классы, пакеты) представлять в виде термов Пролога.

Когда пользователь начинает расчет прогноза состояния лесных ресурсов с помощью программного комплекса, ему необходимо указать некоторые начальные данные о природном объекте (например, тип геосистемы) и задаче прогнозирования (например, длительность). Далее на основе знаний, хранящихся в базе знаний, осуществляется поиск соответствующей модели. Например, начальные данные для модели “Динамика управления древостоем” (ДУД) имеют следующий вид:

$$models(dud, problem(model_dinam), rang(land), type(les), progn(desytillet)).$$

Это утверждение, представленное в виде факта языка Пролог, означает, что для модели ДУД важны данные об объекте, такие как ранг ландшафта, тип ландшафта — лес, временной интервал моделирования — десятилетия.

После того как с помощью БЗ и механизма логического вывода будет найдена модель, подходящая для конкретной задачи прогнозирования, начинается построение выбранной модели. Например, для состояния объекта моделирования в начальный момент времени t_0 получаем, используя правило:

$$fs0(model(dud), Lesh, S, Snep, Nas) : -s0(Lesh, t0, S, Snep, Nas).$$

Результатом работы правила будут значения общей площади лесхоза S , непокрытой площади $Snep$ и численности населения Nas для лесхоза $Lesh$.

Далее для расчетов необходимо получить исходные данные о площадях, занятых породой определенного класса возраста. Для этого применяются правила вида

$$square(Lesh, Prd, vozrast(“молодняки 1кл”), t0, S) : -sq(Lesh, Prd, _, S, _, _, _).$$

Структура типа $sq(Lesh, Prd, S, Sm1, Sm2, Ssr, Spr, Ssp)$ содержит данные о площадях лесхоза $Lesh$ породы Prd : общей S , молодняков 1 кл $Sm1$ и 2 кл $Sm2$, средневозрастных Ssr , приспевающих Spr , спелых и перестойных Ssp .

После того как из БЗ получены все исходные данные, начинается построение последовательности смены участками леса своих возрастных классов по модели ДУД с помощью следующего правила:

$$perekhod(model(dud), Prd, Kl, Kl2, In) : -smena(Kl2, Kl), intens(Prd, In).$$

В модели “Динамика управления древостоем” переход леса породы Prd из класса Kl в класс $Kl2$ с интенсивностью In осуществляется, если $Kl2$ сменяет Kl и интенсивность для породы Prd равна In .

Смена классов возраста представлена термами вида

$$smena(\text{“молодняки 2 кл”}, \text{“молодняки 1 кл”}).$$

Затем строится матрица коэффициентов перехода площадей леса из одного состояния в другое, проводятся численные расчеты. При этом может быть учтено проведение в лесах рубок главного пользования и рубок ухода. Правило проведения рубок выглядит следующим образом:

$$\begin{aligned} rubkaGP(model(dud), porod(K), возраст(V), Vr) : - \\ : -porodRub(K), rubkaGP_type(V), rubGP(K, Vr). \\ rubkaGP_type(\text{“спелые и перестойные”}). \end{aligned}$$

В модели ДУД рубки главного пользования по породе K возраста V объемом Vr проводятся, если порода K может вырубаться, ее возраст V подлежит вырубке и объем ее рубки составляет Vr .

При этом породами, которые могут вырубаться $porodRub()$, являются сосна, лиственница, пихта, ель, береза и осина (кроме кедра). Также для рубок главного пользования предназначены только деревья спелого и перестойного класса возраста $rubkaGP_type()$.

Далее по построенной структуре модели и полученным исходным данным программный комплекс проводит прогнозные расчеты. Для задачи автоматизации идентификации модели ДУД определены следующие входные данные:

- характеристики исследуемого объекта (термы ранг геосистемы $rang()$, тип ландшафта $type_land()$);
- виды пород леса (термы $porod()$);
- виды пород леса, вырубаемых на рассматриваемой территории (терм $porodRub()$);
- типы классов возраста леса (термы $vozrast()$);
- классы возраста леса, который можно вырубать (термы $rubkaGP_type()$).

Алгоритм загрузки кода на языке Пролог и обработки его результатов выглядит следующим образом:

```
// создаем экземпляр класса Prolog
Prolog engine = new Prolog();
// загрузка файла базы знаний
engine.setTheory(new Theory(new FileInputStream('model.pl')));
// запрос к базе
SolveInfo info=engine.solve(quer);
//если есть решение, то...
```



```
if (info.isSuccess()){
    // получаем решение
    String res = info.getSolution(). toString();
    //если есть еще решения
    if (engine.hasOpenAlternatives())
        // обрабатываем следующее
        info = engine.solveNext();
// иначе заканчиваем
else break;
```

Как видно, база знаний содержит правила, позволяющие производить автоматическую идентификацию математической модели. Это делает наглядным процесс моделирования, дает возможность описывать сложные закономерности динамики природных объектов. Также для гибкой подстройки модели к условиям региона не требуется изменять код программной системы расчета прогноза, а необходимо лишь внести требуемые данные в базу знаний или дополнить ее новыми правилами.

Список литературы

- [1] МОДЕЛИ управления природными ресурсами / Под ред. В.И. Гурмана. М.: Наука, 1981. 204 с.
- [2] ЧЕРКАШИН А.К. Система математических моделей леса // Планирование и прогнозирование природно-экономических систем. Новосибирск: Наука, 1984. С. 46–57.
- [3] ЧЕРКАШИН Е.А., ЧУДНЕНКО А.К. Гибридная ГИС прогнозирования динамики лесонасаждений // Вест. ТГУ. Приложение № 9 (II). Докл. V Всерос. конф. с междунар. участием “Новые информационные технологии в исследовании сложных структур”. ISAM’04. Томск, 2004. С. 69–72.
- [4] БЫЧКОВ И.В., ЧЕРКАШИН Е.А., ЧУДНЕНКО А.К. Создание системы поддержки принятия решений по рациональному использованию лесных ресурсов // Матер. Междунар. конф. “Вычислительные и информационные технологии в науке, технике и образовании”. Ч. 1. Алматы—Новосибирск, 2004. С. 364–369.

Поступила в редакцию 25 января 2008 г.