Федеральное агентство по образованию
Государственное образовательное учреждение
высшего профессионального образования
Уфимский государственный авиационный технический университет

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ В САПР

Лабораторный практикум
по дисциплине
«Основы систем автоматизированного проектирования»

Уфа 2007

Федеральное агентство по образованию
Государственное образовательное учреждение
высшего профессионального образования
Уфимский государственный авиационный технический университет

Кафедра двигателей внутреннего сгорания

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ В САПР

Лабораторный практикум
по дисциплине
«Основы систем автоматизированного проектирования»

Уфа 2007

Составители: Ю.Р. Вахитов, С.А. Загайко

УДК 004.42:658.52.011 (07)
ББК 32.973-018.2:30.3-5-05 (я7)

Имитационное моделирование в САПР: Лабораторный практикум по дисциплине «Основы систем автоматизированного проектирования» / Уфимск. гос. авиац. техн. ун-т; Сост.: Ю.Р. Вахитов, С.А.Загайко. – Уфа, 2007. – 39 с.

В работе рассматриваются основные элементы информационного обеспечения систем автоматизированного проектирования. На примере простейших механических систем показано преимущество и возможности имитационной системы моделирования. Лабораторные работы построены таким образом, чтобы студенты могли понять взаимодействие основных видов обеспечения САПР при проектировании.

Предназначен для подготовки бакалавров по направлению 140500 – «Энергомашиностроение» и дипломированных специалистов по направлению 140500 специальности 140501 – «Двигатели внутреннего сгорания», изучающих дисциплину «Основы систем автоматизированного проектирования».

Табл. 21. Ил. 7. Библиогр.: 7 назв.

Рецензенты: канд. техн. наук, доц. Ахмедзянов Д.А.;
канд. техн. наук, ст. преп. Черноусов А.А.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Дисциплина «Основы систем автоматизированного проектирования» является дисциплиной по выбору общепрофессионального цикла учебного плана направления 140500 – «Энергомашиностроение» и специальности 140501 – «Двигатели внутреннего сгорания» направления подготовки дипломированных специалистов 140500. Содержание дисциплины определяется Дополнительными требованиями к Государственным образовательным стандартам высшего профессионального образования по вышеуказанным направлениям.

Целью дисциплины является получение студентами знаний и навыков, необходимых им для изучения смежных дисциплин специального цикла, при выполнении дипломного проектирования и для последующей профессиональной деятельности.

Одним из наиболее перспективных способов повышения качества, сокращения сроков разработки, доводки и подготовки производства сложных технических объектов области энергомашиностроения является автоматизация процесса проектирования отдельных узлов и всей технической системы в целом с помощью систем автоматизированного проектирования (САПР) сложных технических систем (СТС) на основе широкого внедрения вычислительной техники и использования проблемно-ориентированных программных комплексов.

Таким образом, уже на стадии начальных конструкторских проработок оценивается качество, работоспособность и характеристики как элементов, так и всей конструкции энергоустановок. Если элементы САПР, анализирующие работу конструкции энергоустановок, достаточно точно отражают физическую суть процессов, то уже на ранних стадиях разработки СТС можно исключить ошибочные решения до стадии экспериментальной доработки СТС.

Курс «Основы автоматизированного проектирования» ставит своей целью дать всестороннее описание принципов осуществления автоматизации проектирования объектов энергомашиностроения, их классификацию, а также видов обеспечения автоматизированного проектирования, входящих в состав САПР.

В результате изучения дисциплины «Основы автоматизированного проектирования» студент должен иметь представление:

- о возможностях информационных технологий;

- о многообразии существующих прикладных программ для вычислительно-информационной техники;
- о системах автоматизированного проектирования.

Студент должен знать и уметь использовать:

- автоматизированные системы управления, изготовления и испытания двигателей;
- методы моделирования, расчета для разработки новых эффективных конструкций двигателя.

Студент должен иметь представления и владеть:

- навыками пользования вычислительной техникой для решения специальных задач;
- навыками использования расчетных модулей отдельных процессов объектов энергомашиностроения.

Изучению дисциплины «Основы автоматизированного проектирования» должны предшествовать дисциплины «Информатика» и «Инженерная графика».

Развитию этих знаний и умений служат нижеследующие лабораторные работы.

ЛАБОРАТОРНАЯ РАБОТА № 1. ИМИТАЦИОННОЕ МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ МЕХАНИЧЕСКИХ СИСТЕМ

1. Цель работы

Целью работы является изучение метода имитационного моделирования, положенного в основе работы интерактивной системы имитационного моделирования «Альбея», на примере движения механических систем. Работа рассчитана на 8 часов.

2. Общие сведения

Математическое моделирование можно разделить на два вида – ***аналитическое*** и ***имитационное***.

Для аналитического моделирования характерно то, что процессы функционирования элементов системы записываются в виде некоторых функциональных соотношений (алгебраических, интегро-дифференциальных, конечно- разностных и т.п.) или логических условий. Аналитическая модель может быть исследована следующими методами:

а) аналитическим, когда стремятся получить в ***общем виде явные зависимости*** искомых характеристик;

б) численным, когда, не умея решать уравнения в общем виде, стремятся получить числовые результаты при конкретных начальных данных;

в) качественным, когда, не имея решения в явном виде, можно найти некоторые свойства решения (например, оценить устойчивость решения).

При имитационном моделировании алгоритм, реализующий модель, воспроизводит процесс функционирования системы S ***во времени***, причем имитируются элементарные явления, составляющие процесс, с сохранением их логической структуры и последовательности протекания во времени, что позволяет по исходным данным получить сведения о состояниях процесса в определенные моменты времени, дающие возможность оценить характеристики системы S. Время, в котором имитируется модель системы, называется **модельным временем**, а промежуток времени, в течение которого все параметры считаются постоянными, называется **шагом моделирования** или **дискретом по времени.**

Основным преимуществом имитационного моделирования, по сравнению с аналитическим, является возможность решения более сложных задач. Имитационные модели позволяют достаточно просто рассчитывать такие факторы, как наличие дискретных и непрерывных элементов, нелинейные характеристики элементов системы, многочисленные случайные воздействия и др., которые часто создают трудности при аналитических исследованиях. В настоящее время имитационное моделирование – наиболее эффективный метод исследования больших систем, а часто и единственный практически доступный метод получения информации о поведении системы, особенно на этапе ее проектирования.

Метод имитационного моделирования позволяет решать задачи оценки: вариантов структуры системы, эффективности различных алгоритмов управления системой, влияния изменения различных параметров системы. Имитационное моделирование может быть положено также в основу структурного, алгоритмического и параметрического синтеза больших систем, когда требуется создать систему с заданными характеристиками при определенных ограничениях, которая является оптимальной по некоторым критериям оценки эффективности.

Для проведения процесса имитационного моделирования в настоящее время широко используется вычислительная техника. При всех преимуществах имитационного эксперимента его недостатком является то, что решение, полученное при анализе имитационной модели, всегда носит ***частный*** характер, т.к. соответствует фиксированным элементам структуры, алгоритмам поведения и значениям параметров системы S, начальных условий и воздействий внешней среды. Поэтому для полного анализа характеристик процесса функционирования систем, а не получения только отдельной точки приходится многократно воспроизводить имитационный эксперимент, варьируя исходные данные.

3. Порядок выполнения работы

В процессе выполнения работы необходимо знать язык программирования Си или Си++. Работу необходимо выполнять в следующей последовательности:

1. Изучить метод имитационного моделирования, используя рекомендованные источники литературы [7].

2. Получить у своего научного руководителя или у преподавателя, ведущего лабораторные работы, задачу, которую нужно решить методом имитационного моделирования. В качестве задачи можно выбрать одну из перечисленных ниже.

Задание № 1

Резиновый шарик массой 1 кг падает на пол с высоты 1 м и отскакивает от него с коэффициентом восстановления $k = 0{,}9$. На какой высоте от пола будет находиться шарик через 10 с и сколько раз шарик ударится об пол за это время? Сопротивление воздуха считать равным $F_{\text{сопр}} = 0{,}01 \cdot V_{\text{ш}}$ [Н]. Шаг моделирования по времени составляет $\Delta t = 0{,}001$ с.

Задание № 2

Вагон массой 1000 кг катится и ударяется через 200 м в тележку массой 200 кг. Начальная скорость вагона 2,5 м/с. Сила трения колес вагона и тележки составляет $F_{\text{тр}} = 0{,}01 \cdot V$ [Н]. Коэффициент восстановления при ударе $k = 0{,}8$. Какова будет скорость тележки после 1 минуты? Шаг моделирования по времени составляет $\Delta t = 0{,}001$ с.

Задание № 3

Камень массой m = 1 кг (рис. 1.1) падает с высоты 2 м на пружину, лежащую на полу. Длина пружины в свободном состоянии составляет 30 см. Жесткость пружины k = 100 Н/см. На какой высоте от пола будет находиться камень через 10 с? Шаг моделирования по времени составляет $\Delta t = 0{,}001$ с.

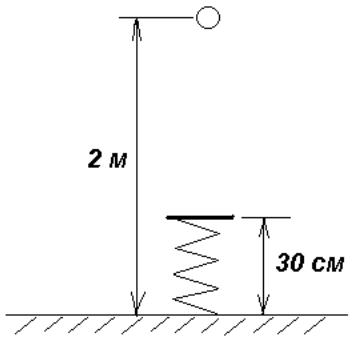

Рис. 1.1

Задание № 4

Камень массой 1 кг (рис. 1.2) падает с высоты 2 м на демпфер, верхний конец которого находится на высоте 1,7 м. Сопротивление демпфера составляет 100 Н/(м/с). Насколько сдвинется конец демпфера через 5 с после начала падения камня? Шаг моделирования по времени составляет $\Delta t = 0{,}001$ с.

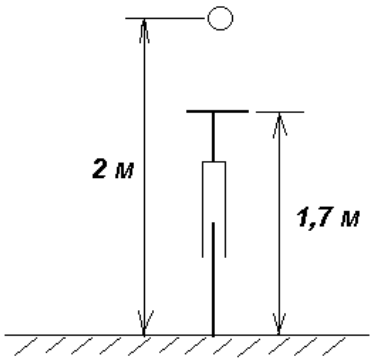

Рис. 1.2

Задание № 5

Шарик массой 1 кг (рис. 1.3) скатывается с наклонной плоскости, находящейся под углом 45° и которая плавно переходит в другую наклонную плоскость под углом 30°. Сила трения качения составляет $F_{тр} = 0{,}01 \cdot V$ [Н]. Через сколько секунд и на каком расстоянии от места начала движения шарик остановится? Шаг моделирования по времени составляет $\Delta t = 0{,}001$ с.

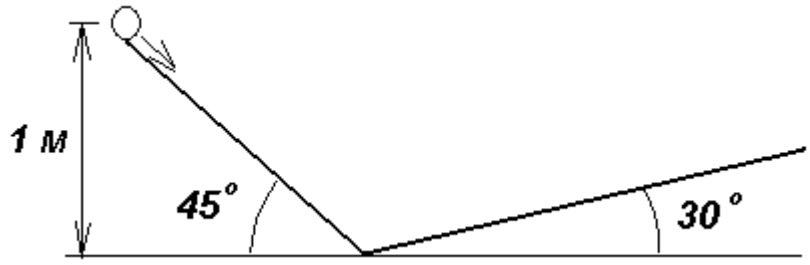

Рис. 1.3

Задание № 6

Ракета взлетела с земли с начальной скоростью $V_o = 100$ м/с под углом 60° к горизонту и начала разгоняться с ускорением 5 м/с^2 в течение 30 с. Масса ракеты составляет 20 кг. На каком расстоянии и на какой секунде от точки взлета ракета упадет? Силу трения ракеты об воздух принять равной $F_{тр} = 0{,}005 \cdot V$ [Н]. Шаг моделирования по времени составляет $\Delta t = 0{,}001$ с.

Задание № 7

Ракета взлетела с земли под углом 45° к горизонту с начальной скоростью $V_o = 150$ м/с и начальной массой 100 кг. В течение 20 с вырабатывалось топливо со скоростью 1 кг/с, что поддерживало тягу $F = 200$ Н. На каком расстоянии и на какой секунде от точки взлета ракета упадет? Шаг моделирования по времени составляет $\Delta t = 0{,}001$ с.

Задание № 8

Автомобиль массой 1000 кг ехал накатом по ровной дороге со скоростью 20 м/с. Сила трения колес составляла $F_{тр} = 0{,}001 \cdot V$ [Н]. Через 100 м дорога пошла в гору под углом 20° к горизонту. Через сколько секунд от начала отсчета автомобиль остановится и на какой высоте? Шаг моделирования по времени составляет $\Delta t = 0{,}001$ с.

Задание № 9

Автомобиль массой 1000 кг (рис. 1.4), стоя на горе на высоте 50 м начал скатываться с горы по ровной дороге. Наклон горы составляет 30^{o} к горизонту. Сила трения колес составляла $F_{тр}$ = 0,005·V [Н]. Через какое время и на каком расстоянии от начала отсчета автомобиль остановится? Шаг моделирования по времени составляет Δt = 0,001 с.

Рис. 1.4

Задание № 10

Стальной шарик массой 1 кг катится по плоскости с начальной скоростью V_o = 5 м/с и ударяется об стенку с коэффициентом восстановления k = 0,7, отскакивая от нее. Начальное расстояние от шарика до стенки составляет 5 м. Сила трения качения $F_{тр}$ = 0,005·V [Н]. На каком расстоянии от стенки и на какой секунде шарик остановится? Шаг моделирования по времени составляет Δt = 0,001 с.

Задание № 11

Стальной шарик массой 2 кг катится по поверхности и через 2 м ударяется в другой шарик массой 1 кг. Коэффициент восстановления удара составляет k = 0,5. Начальная скорость первого шарика V_o = 5 м/с. Сила трения качения $F_{тр}$ = 0,005·V [Н]. Через сколько секунд второй шарик остановится? Шаг моделирования по времени составляет Δt = 0,001 с.

Задание № 12

Автомобиль, массой 1000 кг, накатом катится по ровной дороге с начальной скоростью 30 м/с. Сила трения качения $F_{тр}$ = 0,005·V [Н]. Через 100 начинается участок дороги с песком и сила трения увеличивается $F_{тр}$ = 0,5·V. Через сколько секунд и на каком расстоянии автомобиль остановится? Шаг моделирования по времени составляет Δt = 0,001 с.

Задание № 13

Резиновый шарик массой 1 кг (рис. 1.5) с начальной скоростью V_o = 1 м/с падает на пол с высоты 1 м под углом 60° к горизонту и отскакивает от него с коэффициентом восстановления k = 0,7. На каком расстоянии от начала падения будет находиться шарик через 15 с и сколько раз он отскочит от пола? Сопротивление воздуха считать равным $F_{\text{сопр}} = 0{,}1 \cdot V_{\text{ш}}$ [Н]. Шаг моделирования по времени составляет Δt = 0,001 с.

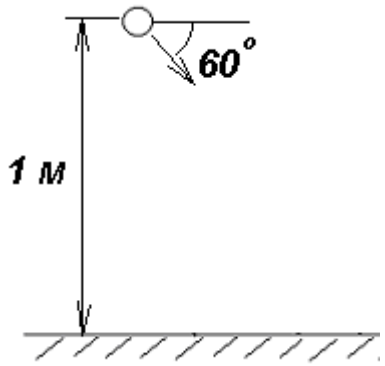

Рис. 1.5

3. На основе метода имитационного моделирования представить алгоритм решения задачи в виде блок-схемы. На блок-схеме использовать обозначения, принятые при составлении блок-схем. Например, пусть дано следующее задание:

Камень массой m = 1 кг и средним диаметром d = 0,1 м брошен под углом к горизонту 45° со скоростью V = 70 м/с. Коэффициент сопротивления воздуха составляет Cx = 0,4. Плотность воздуха принять равной 1,293 кг/м^3. Ускорение свободного падения 9,81 м/с^2. Шаг моделирования по времени составляет Δt = 0,1 с. Определить, на какой секунде и на каком расстоянии от места броска камень приземлится? Сила сопротивления воздуха определяется по формуле $F_в = Cx \dfrac{\pi d^2}{4} \rho_в \dfrac{V^2}{2}$.

Пример блок-схемы приведен на рис. 1.6.

4. По блок-схеме написать программу на языке C или C++, используя принятые стандарты написания программ [3].

5. Набрать подготовленную программу в системе программирования Borland C++ 3.1 или Builder C++ 5.0. Пример написания программы приведен в Приложении.

6. Запустить программу на расчет и провести ее отладку.

7. Отчетностью за лабораторную работу является работающая программа, решающая поставленную задачу методом имитационного моделирования.

8. Защита работы заключается в ответе на вопросы для контроля и дополнительные вопросы преподавателя.

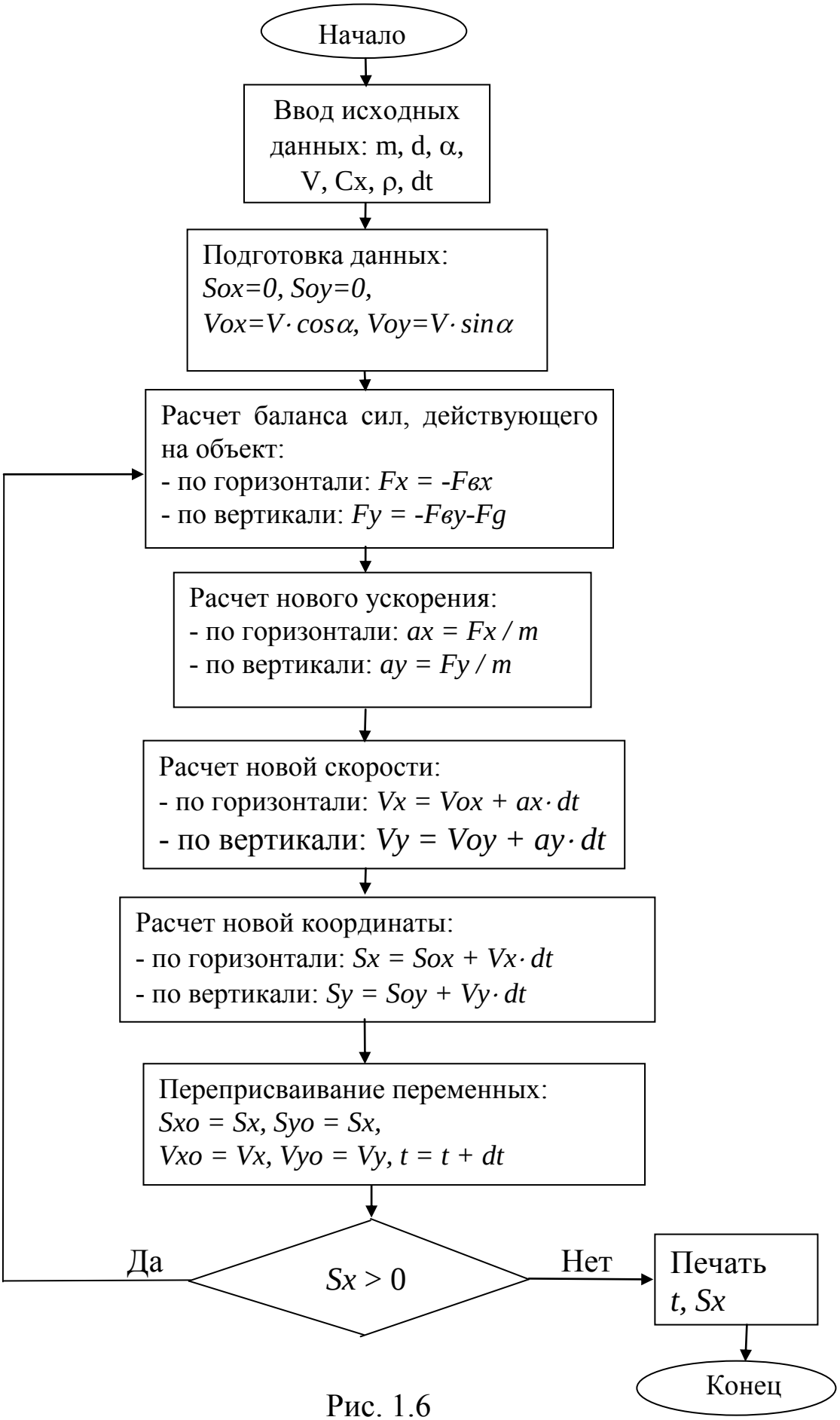
Начало
Ввод исходных данных: m, d, α, V, Cx, ρ, dt
Подготовка данных:
Sox=0, Soy=0,
Vox=V·cosα, Voy=V·sinα
Расчет баланса сил, действующего на объект:
- по горизонтали: Fx = -Fвх
- по вертикали: Fy = -Fвy-Fg
Расчет нового ускорения:
- по горизонтали: ax = Fx / m
- по вертикали: ay = Fy / m
Расчет новой скорости:
- по горизонтали: Vx = Vox + ax·dt
- по вертикали: Vy = Voy + ay·dt
Расчет новой координаты:
- по горизонтали: Sx = Sox + Vx·dt
- по вертикали: Sy = Soy + Vy·dt
Переприсваивание переменных:
Sxo = Sx, Syo = Sx,
Vxo = Vx, Vyo = Vy, t = t + dt
Sx > 0
Да
Нет
Печать
t, Sx
Конец

Рис. 1.6

4. Вопросы для контроля

1. Назовите преимущества и недостатки метода имитационного моделирования по сравнению с методами аналитического моделирования.

2. В чем заключается метод имитационного математического моделирования?

3. От чего зависит точность метода имитационного моделирования?

4. Какие ресурсы компьютера важны при реализации метода имитационного моделирования?

5. Что называется модельным временем и шагом моделирования?

6. К какому виду обеспечения САПР относится метод имитационного моделирования?

7. Каким образом можно реализовать метод имитационного моделирования при проектировании двигателя?

ЛАБОРАТОРНАЯ РАБОТА № 2. РАЗРАБОТКА СТРУКТУРЫ БАЗЫ ДАННЫХ В ПАКЕТЕ ACCESS MICROSOFT OFFICE 2000

1. Цель работы

Целью работы является изучение архитектуры баз данных и получение основных навыков создания и ведения базы данных на примере пакета Access Microsoft Office 2000 (далее Microsoft Access). Работа рассчитана на 4 часа.

2. Общие сведения

Компьютер с помощью тех или иных программ способен обрабатывать разнообразные данные, в том числе и базы данных. **Базой данных** называется набор данных, который достаточен для установленной цели и представлен на машинном носителе в виде, позволяющем автоматизированную переработку содержащейся в нем информации.

Совокупность программного обеспечения, необходимого для ведения и использования баз данных называется **системой управления базами данных** или **СУБД**.

Для автоматизированной переработки данных или информации они должны быть структурированы, т.е. представлены определенным способом. Данные могут быть представлены в виде различных структур. В *иерархической структуре* исходные элементы порождают другие элементы, причем эти элементы в свою очередь порождают следующие элементы и т. д. Такую структуру можно представить как дерево, ствол которого – это набор объектов, а ветки – объекты боле низкого уровня. В *сетевых структурах* каждый порожденный элемент может иметь более одного порожденного элемента. Кроме того, данные могут быть сведены в совокупность двумерных таблиц, которая называется *реляционной базой данных*.

Microsoft Access представляющей собой систему управления реляционными базами данных.

Совокупность структурированных данных и комплекса аппаратно – программных средств для хранения данных и манипулирования ими представляют собой **информационную систему**.

Информационные системы выполняют следующие функции:

1) хранение информации;

2) просмотр и поиск информации в базе;

3) создание выборки по определенным критериям;
4) создание форматированных отчетов;
5) ввод и редактирование информации;
6) контроль информации;
7) отображение информации в требуемом виде;
8) архивирование и создание копий архивов.

Организационно-техническая система, включающая совокупность баз данных, технические и программные средства формирования и ведения этих баз и коллективов специалистов, обеспечивающих функционирование системы, называется **банком данных.**

Говоря проще:

база данных – это, например, дискета со структурированными данными;

СУБД – это компьютерная программа для работы с базой данных;

информационная система – это компьютер с базой данных и установленной на нем программой;

банк данных – это информационная система (компьютер, программа и данные) со специалистами.

В базу данных Access может входить шесть типов объектов. Основным объектом базы данных является **таблица**. Каждый столбец таблицы соответствует полю данных, а каждая строка – одной записи данных. В каждое поле записывается однотипная информация. Например, в задании 1 полями являются **Адресат, Фамилия** и т.д.

Таблица может содержать одно или несколько ключевых полей, так называемый ключ. **Ключ** – это поле (или поля), однозначно характеризующее запись. Например, в задании 1 ключевым полем может быть только поле **Адресат.**

Самостоятельное формирование таблицы проводится в режиме конструктора. Каждому полю приписывается определенный тип данных, которые приведены в табл. 2.1.

Если тип данных поля имеет значение **Числовой**, то допустимыми являются значения свойства **Размер поля**, приведенные в табл. 2.2.

Таблица 2.1

Тип данных	Описание
Текстовый	Текст длиной до 255 символов
Поле MEMO	Текст длиной до 64 000 символов
Числовой	Число
Дата/время	Дата и время
Денежный	Денежная сумма с точностью 15 целых и четырех десятичных разрядов
Счетчик	Целое число, автоматически увеличивающееся на единицу при добавлении новой записи. Не редактируется.
Логический	Логические значения Да/Нет (Истина/Ложь, Вкл/Выкл)
Поле объекта OLE	Объекты OLE размером до 128 Мб
Гиперссылка	Ссылка на другой файл

Таблица 2.2

Значение	Описание
Байт	Целое число от 0 до 255
Целое	Целое число от -32 768 до 32 767
Длинное целое	Целое число от -2 147 483 648 до 2 147 483 647
С плавающей точкой (4 байт)	Число от $-3{,}402823\cdot10^{38}$ до $-1{,}401298\cdot10^{-45}$ и от $1{,}401298\cdot10^{-45}$ до $3{,}402823\cdot10^{38}$
С плавающей точкой (8 байт)	Число от $-1{,}79769313486232\cdot10^{308}$ до $-4{,}94065645841247\cdot10^{-324}$ и от $1{,}79769313486231\cdot10^{308}$ до $4{,}94065645841247\cdot10^{-324}$

3. Порядок выполнения работы

1. Последовательно выполнять задания 1-15.

Задание 1

Создать базу данных **Адреса** (табл. 2.3) с таблицей адресов, используя **Мастер таблиц**. Ввести пять произвольных записей.

Таблица 2.3

Адресат	Фамилия	Имя	Отчество	Организация	Телефон	Примечание
1						
2						
3						
4						

Задание 2

Создать самостоятельно базу данных **Каталог** (табл. 2.4).

Таблица 2.4

Код рисунка	Наименование	Дата создания	Цена	Рисунок
00-01	круг	01.01.99	100	Точечный рисунок BMP
00-02	квадрат	12.02.99	120	Точечный рисунок BMP
00-03	треугольник	13.02.99	230	Точечный рисунок BMP
00-04	овал	13.02.99	170	Точечный рисунок BMP

Указания.

Данные в поле **Код рисунка** вводить по маске. В поле **Рисунок** вставить рисунки, выполненные в Paint (тип данных – **Поле объекта OLE**).

Для вставки рисунка необходимо над полем **Рисунок** выполнить команду контекстного меню **Добавить объект**, в поле **Тип объекта** выделить Точечный рисунок BMP. В открывшемся редакторе Paint создать рисунок и выполнить команду меню **Файл | Выход и возврат в …**. Для просмотра объекта OLE необходимо выполнить двойной щелчок над соответствующей записью поля **Рисунок.**

Задание 3

Создать базу данных **Автомобили**, состоящую из двух таблиц: **Модели автомобилей** (табл. 2.5) и **Двигатели** (табл. 2.6).

Таблица 2.5

Модели автомобилей

Марка	Двигатель	Масса	Скорость	Расход топлива
Ваз 2103	Ваз 2103	1430	152	8,4
Ваз 21033	Ваз 21011	1430	147	8,3
Ваз 2106	Ваз 2106	1445	154	8,5
Ваз 21061	Ваз 21011	1445	147	8,3
Ваз 21063	Ваз 2103	1445	152	8,4

Таблица 2.6

Двигатели

Модель	Диаметр цилиндра	Ход поршня	Рабочий объем	Мощность	Момент	Обороты
Ваз 2101	76	66	1,2	64	8,9	5600
Ваз 21011	79	66	1,3	69	9,6	5600
Ваз 2103	76	80	1,45	77	10,8	5600
Ваз 2106	79	80	1,57	80	12,4	5400

Задание 4

Создать базу данных **Семестр**, состоящую из двух связанных таблиц: **Расписание занятий** (табл. 2.7) и **Преподаватели** (табл. 2.8).

Указания.

Таблицы связать по полю **Преподаватели**. Главной должна быть таблица **Преподаватели.**

Таблица 2.7

Расписание занятий

Дисциплина	Начало занятий	Всего часов	Дата зачета	Преподаватель
История	02.09.99	50	15.12.99	Иванов
Химия	23.09.99	60	17.12.99	Сидоров
Физика	01.09.99	70	22.12.99	Петров
Математика	04.09.99	70	26.12.99	Николаев
Графика	02.10.99	40	17.12.99	Егоров
Черчение	03.10.99	50	19.12.99	Егоров

Таблица 2.8

Преподаватели

Фамилия	Должность	Рабочий телефон
Иванов	Доцент	23-03-88
Сидоров	Доцент	23-45-83
Петров	Профессор	23-67-54
Николаев	Доцент	23-12-23
Егоров	Ассистент	23-67-11
Александров	Доцент	23-77-34

При связывании таблиц желательно обеспечить целостность данных, для чего при связывании таблиц необходимо установить флажок **Обеспечение целостности данных**. Кроме того, могут быть установлены флажки **Каскадное обновление полей** и **Каскадное удаление полей**. Действие этих флажков описано в табл. 2.9.

Таблица 2.9

Положение флажков		Действие
Обеспечение целостности данных *не установлено*	–	Возможны любые изменения, но целостность базы не контролируется
Обеспечение целостности данных *установлено*	Каскадное обновление полей *установлено*	При изменении записи в главной таблице автоматически изменяются записи в подчиненной таблице
	Каскадное удаление полей *установлено*	При удалении записи в главной таблице автоматически удаляются записи в подчиненной таблице
	Каскадное обновление полей *не установлено*	Изменение записи в главной таблице не возможно
	Каскадное удаление полей *не установлено*	Удаление записи в главной таблице не возможно

Вторым типом объекта базы данных является запрос, который формируется на основе существующих в базе данных таблиц. **Запросом** называют установление критериев поиска или условий отбора, по которым будут выводиться данные. Результатом запроса является **выборка**. Можно сказать, что запрос в режиме таблицы является выборкой, а запрос в режиме конструктора – собственно запросом.

Условия отбора для тех или иных полей записываются в виде выражений. Синтаксис некоторых выражений приведен в табл. 2.10. Для ускорения работы можно вводить выражения в упрощенном виде (см. второй столбец табл. 2.10); Microsoft Access автоматически исправит синтаксис.

Таблица 2.10

Выражение	**Упрощенное выражение**	**Результат**
«Иванов»	Иванов	Только Иванов
«Иванов» Or «Сидоров»	Иванов Or Сидоров	Иванов или Сидоров
Not «Иванов»	Not Иванов	Все кроме Иванова
#01.01.97#	01.01.97	только 01.01.97
Between #01.01.97# And #25.01.97#	Between 01.01.97 And 25.01.97	Между 01.01.97 и 25.01.97#
>Date()-30	> Date() – 30	За последние 30 дней
«И*»	и*	Начинается на И
«*И»	*и	Заканчивается на И
>«Н»	>Н	С буквы Н по Я
<«Н»	<Н	С буквы А до Н

Задание 5

Создать в базе данных **Семестр** (Задание 4) следующие запросы:

1. Запрос «Дисциплины, преподаваемые доцентами».

2. Запрос «Дисциплины, сдаваемые с 16.12.99 по 20.12.99». В запросе указать дисциплину, дату зачета и фамилию преподавателя.

Запрос, проводящий вычисления, может содержать групповые функции, перечисленные в табл. 2.11.

Таблица 2.11

Функция	**Назначение**
Sum	Сумма значений
Avg	Среднее значение
Min	Минимальное значение
Max	Максимальное значение
Count	Количество записей
StDev	Стандартное отклонение значений
Var	Дисперсия значений
First	Первое значение
Last	Последнее значение

Задание 6

Создать базу данных с таблицей (см. табл. 2.12).

Таблица 2.12

Фамилия	**Отдел**	**Оклад**
Федоров А.А.	2	780
Степанов Б.Б.	1	600
Иванов И.И.	2	450
Иванов И.И.	1	300
Петров П.П.	1	900
Сидоров С.С.	2	300
Кузнецов Ч.Ч.	1	850

Создать запрос «Количество работающих в каждом отделе».

Создать запрос «Средние оклады по отделам».

Создать запрос «Средний оклад по двум отделам».

Запросы условно делятся на четыре типа. Рассмотренные запросы являются **запросами на выбор**. В запросе на выбор критерии задаются при формировании запросов в режиме конструктора. При изменении критериев изменяется сам запрос.

Вторым типом запросов является **запрос на выполнение действия**. Главной отличительной особенностью этого запроса является то, что при его выполнении изменяется содержимое таблиц или создается новая таблица.

В **параметрическом запросе** часть критериев настраивается непосредственно при его выполнении по запросу Access через диалоговые окна.

Перекрестный запрос предусматривает групповые операции и в отличие от запроса на выбор представляет данные (выборку) в более компактной форме.

Задание 7

Создать базу данных **Расписание**, состоящую из двух таблиц: **Расписание движения автобусов** (табл. 2.13) и **Типы автобусов** (табл. 2.14).

Таблица 2.13

Расписание движения автобусов

Пункт назначения	Номер рейса	Время отправления	Время в пути	Тип автобуса
Николаев	12	8.00	3:30	ЛАЗ-695
Гурьев	23	8.30	4:00	КАВЗ-685
Иваново	15	9.00	5:20	ПАЗ-672
Орловка	17	9.00	2:10	ЛАЗ-695
Александров	3	9.30	7:00	ЛАЗ-699

Таблица 2.14

Типы автобусов

Марка автобуса	Число мест для сидения	Число мест общее
КАВЗ-685	21	28
ПАЗ-672	23	37
ЛАЗ-695	34	67
ЛАЗ-699	41	70

Создать запрос «Куда ездят автобусы марки «ЛАЗ».

Создать запрос, включающий поле «Время прибытия» (вычисляемое).

Создать запрос «Автобусы, отправляющиеся после 8:30».

Создать запрос на обновление времени отправления: время, меньшее 8:30, заменить на 8:30.

Создать запрос на поиск записей, не имеющих подчиненных: автобусы, которых нет в расписании.

Создать запрос с поиском повторяющихся записей: время отправления.

Создать запрос «Количество рейсов, выполняемых автобусами ЛАЗ-695.

Экспортировать табл. 2.14 **Типы автобусов** в Excel.

Третьим объектом базы данных является форма. **Форма** – это вид представления информации, хранящейся в таблице или запросе. Форма может быть более удобной для ввода и редактирования записей, в ней OLE объекты представлены в открытом виде; форма может содержать элементы управления; внешний вид формы более привлекателен.

Задание 8

Создать базу данных, состоящую из табл. 2.15.

Таблица 2.15

Двигатели	Мощность в л.с.
ВАЗ 2101	64
ВАЗ 21011	69
ВАЗ 2103	72
ВАЗ 2106	80
ВАЗ 21081	54
ВАЗ 2108	63
ВАЗ 21083	70

Используя **Мастер**, создать на основе данных из табл. 2.15 форму-диаграмму.

Задание 9

Создать базу данных с табл. 2.16.

Таблица 2.16

Модели автомобилей Рено

Модель	Рабочий объем двигателя	Соответствие нормам Евро 2000	АБС	Максимальная скорость	Масса
RT	1150	Нет	Нет	160	975
RT/RXE	1400	Нет	Есть	170	1035
RXE/RXT	1600	Нет	Есть	181	1060
Proactive	1600	Нет	Есть	175	1075
16 V	1600	Да	Есть	195	1070

С помощью **Мастера форм** создать форму, содержащую все поля таблицы.

Задание 10

Создать базу данных с табл. 2.17.

Таблица 2.17

Книги, журналы и газеты

Название	Категория	Автор	Цена	Количество	Переплет	Язык	Зарубежное издание	Эмблема издательства
Стилист	Книга	Маринина	20	5	Твердый	Русский	Нет	OLE
Черный список	Книга	Маринина	20	4	Твердый	Русский	Нет	OLE
Детект. романы	Книга	Кристи	30	6	Твердый	Русский	Нет	OLE
Noice	Книга	Taylor	15	1	Твердый	Иност.	Да	OLE
Verena	Журнал		22	1	Мягкий	Иност.	Да	OLE
Бурда	Журнал		30	2	Мягкий	Русский	Да	OLE
Известия	Газета		5	1		Русский	Нет	OLE

Создать форму-диаграмму (категория – общее количество экземпляров).

Создать форму, включающую данные из таблицы.

В форме создать следующие элементы:

- Поля – **Название, Автор, Цена.**
- Поле со списком – **Переплет** (твердый, мягкий).
- Группа переключателей – **Категория** (Книга, Журнал, Газета).
- Флажок – **Зарубежное издание.**

- Присоединенная рамка объекта – **Эмблема.**
- Выключатель – **Язык** (Иностранный/Русский).
- Командную кнопку, открывающую форму-диаграмму.
- Вычисляемое поле – **Цена комплекта** = **Цена** * **Количество.**
- В заголовок формы вставить надпись и рисунок.

Четвертым типом объекта базы данных является отчет. **Отчет** – это еще один вид представления информации, но в отличие от формы содержит результаты анализа данных и вычислений.

Задание 11

Создать базу данных **Библиотека**, состоящую из таблицы **Книги** (табл. 2.18).

Таблица 2.18

Книги

Инвентарный номер	**Автор**	**Название**	**Год издания**	**Цена**	**Количество экземпляров**
1	Драйзер	Титан	1988	30	10
2	Платонов	Проза	1990	25	9
3	Доде	Евангелистка	1988	20	5
4	Леонов	Беспредел	1999	15	15
5	Рогожин	Новые русские	1999	21	13

Используя **Мастер отчетов**, создать отчет. Отчет должен содержать поля **Инвентарный номер**, **Год издания**, **Цена** и **Количество экземпляров**; уровень группировки – **Год издания**; итоговые значения для вычисления – **Количество экземпляров.**

Самостоятельное формирование отчета проводится в режиме конструктора. Макет отчета может содержать несколько областей. Назначение областей приведено в табл. 2.19.

Таблица 2.19

Область отчета	Расположение	Назначение области
Заголовок отчета	В начале отчета	Заголовок отчета
Верхний колонтитул	Вверху каждой страницы	Имена полей данных
Заголовок группы	В начале каждой группы	Заголовок группы
Область данных	В центре каждой группы	Записи данных группы
Примечание группы	В конце каждой группы	Вычисляемые значения по группе
Нижний колонтитул	Внизу каждой страницы	Номер страницы, дата
Область примечаний	В конце отчета	Итоговые значения

Задание 12

Создать базу данных **Слушатели 1 курса**, состоящую из таблицы **Ведомость** (табл. 2.20).

Таблица 2.20

Ведомость

Порядковый номер	Фамилия	Оценка	Дата сдачи
1	Иванов	5	25.02.06
2	Сидоров	5	25.02.06
3	Петров	4	10.03.06
4	Егоров	4	25.02.06
5	Александров	3	25.02.06
6	Федоров	5	25.02.06
7	Ильин	4	10.03.06
8	Володин	4	25.02.06
9	Юрьев	5	25.02.06
10	Никонов	3	25.02.06

Создать отчет, содержащий группировку по оценкам (количество слушателей, получивших тройки, четверки и пятерки).

Создать другой отчет, выводящий количество слушателей, сдавших экзамен после 25 февраля 2006 г.

Задание 13

В базе данных **Слушатели 1 курса** (Задание) выполните следующее.

1. Создайте еще одну таблицу с домашними адресами слушателей. Адреса ввести произвольно.

2. Создайте почтовые наклейки с адресами всех слушателей для конвертов.

3. Создайте приглашения на вручение дипломов, используя операцию **Слияние с MS Word.**

Указания

Третий пункт задания можно выполнить в следующей последовательности.

Перейдите на вкладку **Таблицы** и выполните команду **Слияние с MS Word**. В окне **Слияние с документами MS Word** установите переключатель в положение **Создание нового документа…** После запуска MS Word напишите текст приглашения. В процессе написания текста вставьте в нужное место документа поле таблицы базы данных, для чего нажмите кнопку **Добавить поле слияния** и выберите нужное поле (для данного задания поле **Фамилия**). После подготовки документа выполните команду **Слияние в новый документ** и сохраните документ.

Для автоматизации часто выполняемых задач могут использоваться макросы (пятый тип объекта базы данных). **Макросом** называют набор из одной или более макрокоманд, выполняющих определенные операции. **Макрокоманда** – это замкнутая инструкция, самостоятельно или в комбинации с другими командами определяющая выполняемые в макросе действия.

Задание 14

В базе данных **Слушатели 1 курса** (см. Задание) выполните следующее.

1. Создайте форму с полями **Фамилия** и **Оценка.**

2. Создайте макрос, выводящий предупреждающее сообщение в случае ввода в форме оценок больше 5 или меньше 2.

Указания

Для создания макроса на вкладке **Макрос** нажмите кнопку **Создать**. Выполните команду меню **Вид | Условия**. В поле **Условие** введите выражение [Оценка]<2 or [Оценка]>5. В поле **Макрокоманда** установите макрокоманду **Сообщение**, и установите необходимые значения аргументов макрокоманды. Сохраните макрос под именем **Сообщение**. Откройте форму в режиме конструктора. Для поля **Оценка** выполните команду контекстного меню **Свойства** и для свойства **После обновления** установите макрос **Сообщение.**

Данный макрос не препятствует вводу недопустимых оценок, а только выводит сообщение.

Шестым типом объекта базы данных является **модуль** – набор объявлений и процедур на языке Visual Basic для приложений, собранных в одну программную единицу. Существует два типа модулей: *стандартные модули* и *модули класса*. Стандартным называется модуль, который может использоваться в любом месте базы данных (макросе, форме или отчете), а модулем класса называется модуль, связываемый только с конкретной формой или отчетом.

Задание 15

Создать базу данных, состоящую из таблицы **Прейскурант** (табл. 2.21) и ленточной автоформы **Прейскурант**. Создать стандартный модуль, автоматически заполняемый в форме поле **Цена в рублях.**

Таблица 2.21

Прейскурант

Номер	Наименование	Цена в долларах	Цена в рублях
1	Стол		
2	Стул		
3	Кресло		

Указания

Для создания модуля на вкладке **Модули** нажмите кнопку **Создать**. Выполните команду **Добавить процедуру** и введите имя **Доллар_в_рубли** (тип процедуры – функция, область определения – общая). Используя построитель выражений, введите выражение:

Forms![Прейскурант]![Цена в рублях] =
Forms![Прейскурант]![Цена в долларах] * 26.

Сохраните модуль и закройте его.

Откройте форму в режиме конструктора и активизируйте список свойств поля **Цена в рублях**. В поле **После обновления** введите имя функции: =Доллар_в_рубль(), используя построитель выражений. Сохраните форму.

2. Отчетностью за лабораторную работу являются работающие базы данных, выполненные по заданиям 1-15.

3. Защита работы заключается в ответе на вопросы для контроля и дополнительные вопросы преподавателя.

4. Вопросы для контроля

1. Дайте определение базы данных.
2. Что называется системой управления базой данных (СУБД)?
3. Что представляет собой информационная система?
4. К какому виду обеспечения САПР относится СУБД?
5. Дайте определение банка данных.
6. Что называется ключевым полем?
7. Какие типы имеют поля данных?
8. Что такое запрос?
9. Для чего предназначен мастер форм?
10. Для чего предназначен мастер отчетов?
11. Для чего используются макрокоманды (макросы)?

СПИСОК ЛИТЕРАТУРЫ

1. **Гридин, В.В.** Microsoft Access. Быстрый старт / В.В. Гридин, А. Хомоненко. – СПб: БХВ-Петербург, 2005. – 304 с.

2. **Архангельский, А.Я.** C++ Builder 6. Справочное пособие. В 2-х кн. / А.Я. Архангельский. – Спб: Бином. – 2003.

3. **Методические указания** по оформлению программных продуктов, написанных на языке С++ / Уфимск. гос. авиац. техн. ун-т; Сост.: С.А. Загайко, И.Б. Рудой, А.А. Черноусов. – Уфа, 2006. – 56 с.

4. **Культин, Н.** C++ Builder / Н. Культин – СПб: БХВ-Петербург, 2005. – 320 с.

5. **Бекаревич, Ю.** Microsoft Access 2003 / Ю. Бекаревич– СПб: БХВ-Петербург, 2005. – 752 с.

6. **Бекаревич, Ю.** Самоучитель Microsoft Access 2002 / Ю. Бекаревич, Н. Пушкина. – СПб: БХВ-Петербург, 2004. – 720 с.

7. **Горбачев, В.Г.** Система имитационного моделирования «Альбея» (ядро). Руководство пользователя. Руководство программиста. Уч. пособие. / В.Г. Горбачев [и др.]. Уфимск. гос. авиац. техн. ун-т; Уфа: 1995. – 112 с.

ПРИЛОЖЕНИЕ

```
// "program.cpp"
//
// Программа расчета движения материальной точки
// с помощью закона сохранения импульса.
//
// (C) Иванов И.И., гр. ДВ-999
//
// Численным решением системы из четырех ОДУ первого порядка динамики МТ
// моделируется движение тела массой $m$, брошенного под углом 45^{0}
// к горизонту с учетом силы лобового сопротивления.
// Направление координатных осей: x - вправо, y - вверх.
//

using namespace std; // используем объекты в станд. простр. имен

#include <iostream>  // I/O: cout, <<, endl, fixed, streamsize, precision(
#include <cmath>     // математика: sqrt(), M_PI_4

const int ULEN = 4;  // длина вектора конс. переменных ("размер" задачи)

void output(double t, double x, double y, double u, double v);

//
// main(): главная функция
//
int main()
{
  // исходные данные
  const double
    m  = 1.,   // масса, кг
    d  = 0.1,  // диаметр, м
    Cx = 0.4,  // постоянный коэффициент сопротивления, б/р
    rho= 1.293,// постоянная плотность воздуха, кг/м^{3}
    g  = -9.81,// ускорение свободного падения $g_{x}$, м/с^{2}
    x0 = 0.,   // x-компонента начальной координаты, м
    y0 = 0.,   // y-компонента начальной координаты, м
    u0 = 50.,  // x-компонента начальной скорости, м/с
    v0 = 50.,  // y-компонента начальной скорости, м/с
    dt = 0.1;  // шаг по времени, с
  const int
    N = 100;   // число шагов  (или 'numStep')

  // первичные переменные на шаге: (n)-м временнОм слое
  double x, y, u, v;

  // рабочие переменные: модули скорости и силы сопротивления
  double V, F;

  // вектор источников (правых частей уравнений - ОДУ)
  double S[ULEN];
```

```
// вектор консервативных переменных
double U[ULEN];

// время
double t;

// номер шага (или 'step') и индекс уравнения
int n, k;

// первичные переменные на старте, (0)-м временнОм слое
x = x0;
y = y0;
u = u0;
v = v0;
t = 0.;

// вывод решения на старте (начальные условия)
output(t, x, y, u, v);

// вектор консервативных переменных на старте
U[0] = x;
U[1] = y;
U[2] = m * u; // x-компонента количества движения, кг * м/с
U[3] = m * v; // y-компонента...

// цикл шагов расчета по времени
for (n = 0; n < N; n++) {

  // расчет силы сопротивления движению (для уравнений импульса)
  V = sqrt(u * u + v * v);
  F = Cx * (M_PI_4 * d * d) * rho * V * V / 2.;

  // вычисление правых частей на старом слое: $\overline{F}(U^{n})$
  S[0] = u;
  S[1] = v;
  S[2] = 0.    - F * u / V; // в проекциях
  S[3] = m * g - F * v / V;

  // обновление консервативных переменных (метод Эйлера 1-го порядка)
  for (k = 0; k < ULEN; k++) {
    //
    // \overline{U}^{n+1} = \overline{U}^{n}
    //           + \Delta t \overline{F}(U^{n})$
    //
    U[k] = U[k] + dt * S[k];
  }

  // обновление времени $t^{n+1} = t^{n} + \Delta t$
  t += dt;

  // 'раскодирование' вектора консервативных переменных до первичных
```

```
    x = U[0];
    y = U[1];
    u = U[2] / m;
    v = U[3] / m;

    // вывод решения на шаге
    output(t, x, y, u, v);

  } // end for()

  // OK
  return 0;

} // end main()

// функция вывода текущего решения на шаге в стандартный поток вывода
void output(double t, double x, double y, double u, double v)
{
  // установим режим вывода с тремя знаками после запятой
  streamsize precnOld = cout.precision(3);

  // выводим в установленном режиме, вывод всех 3 знаков после запятой
  cout
    << fixed << t << ' '
    << fixed << x << ' ' << fixed << y << ' '
    << fixed << u << ' ' << fixed << v << endl;

  // восстановим старый режим
  cout.precision(precnOld);
}

// end of "program.cpp"
```

```
#------------------------#
# Makefile для Примера 4 #
#------------------------#

# флаги компилятора
CXXFLAGS = -s -O2
# имя исполняемого файла программы
PROGRAM = program

# выполнение программы для обновления файла результатов расчета
test: output.dat
output.dat: $(PROGRAM)
  ./$(PROGRAM) > output.dat

# построение программы из исходного текста на C++
$(PROGRAM): program.cpp
  g++ $(CXXFLAGS) program.cpp -o $(PROGRAM)
```

```
// "main.cpp"
//
// Главный файл программы расчета движения материальной точки
// по закону сохранения импульса.
//
// (C) Иванов И.И., гр. ДВ-999
//

using namespace std; // используем объекты в станд. простр. имен

#include <iostream>  // I/O: cout,<<,endl,fixed,streamsize,precision()
#include <cmath>     // математика: sqrt(), M_PI_4

#include "Mass.h"    // определение объекта МАССА

//
// main(): главная функция
//
int main()
{
  const double dt = 0.1;  // шаг по времени, с
  const int     N = 100;  // число шагов  (или 'numStep')

  double t;  // время

  // заводим объект МАССА (при этом загружаются исх. данные)
  Mass mass;

  // инициализируем его
  mass.init();

  // инициализируем начальное модельное время
  t = 0.;

  // выведем параметры решения на старте (начальные условия)
  mass.output(t);

  // в цикле шагов расчета по времени
  for (int n = 0; n < N; n++) {

    // обновляем состояние моделируемого объекта
    mass.update(dt);

    // обновляем время
    t += dt;
```

```
    // выведем параметры решение на n-м шаге
    mass.output(t);

  } // end for()

  // OK
  return 0;

} // end main()

// end of "main.cpp"
```

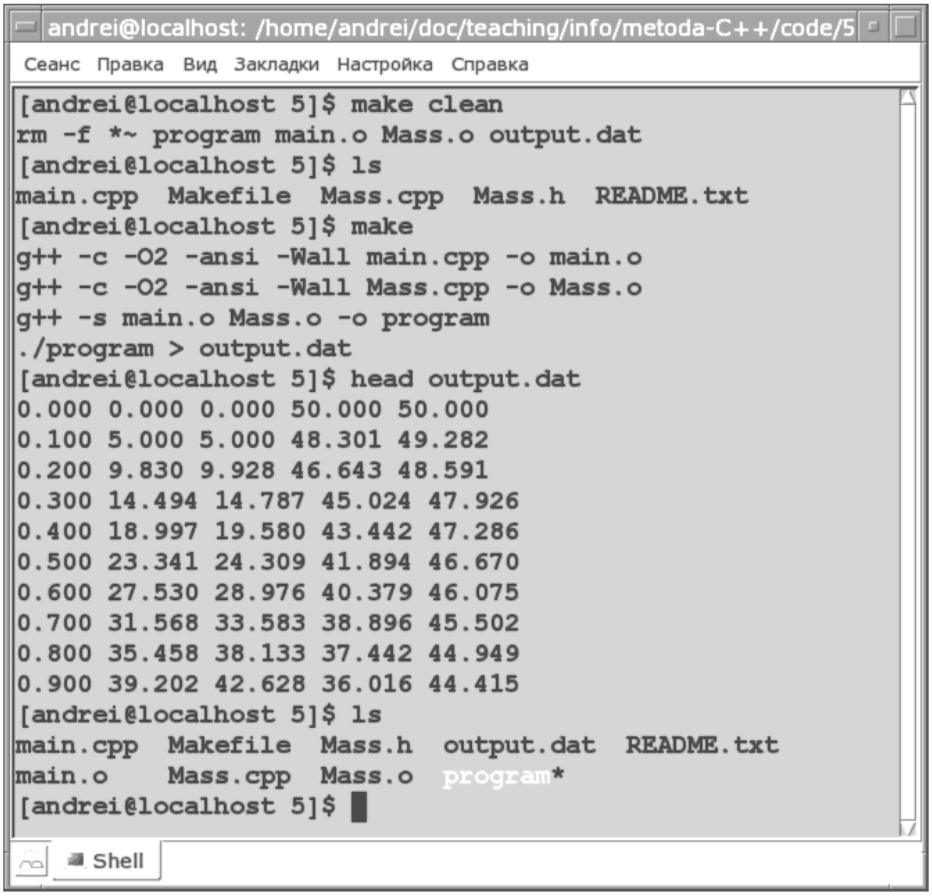

Рис. П1. Окно вывода программы

```
#ifndef _MASS_H_
#define _MASS_H_

// "Mass.h"
//
// Класс объекта МАССА (тело массой $m$, брошенное под углом 45^{0}
// к горизонту, движение которого рассчитывается численным решением
// системы из четырех ОДУ первого порядка динамики материальной точки (МТ)
// с учетом силы лобового сопротивления (по закону сохранения импульса).
//  Расчет по шагам идет методом Эйлера 1-го порядка, направление
// координатных осей: x - вправо, y - вверх.
//
// (C) Иванов И.И., гр. ДВ-999
//

class Mass {

 private: // закрытые данные класса

  // длина вектора конс. переменных ("размер" задачи)
  static const int ULEN = 4;

  // исходные данные
  double
    m_m,  // масса, кг
    m_d,  // диаметр, м
    m_Cx, // постоянный коэффициент сопротивления, б/р
    m_rho,// постоянная плотность воздуха, кг/м^{3}
    m_g,  // ускорение свободного падения $g_{x}$, м/с^{2}
    m_x0, // x-компонента начальной координаты, м
    m_y0, // y-компонента начальной координаты, м
    m_u0, // x-компонента начальной скорости, м/с
    m_v0; // y-компонента начальной скорости, м/с

  // первичные (рабочие) переменные, текущее решение
  double
    m_x, // x-компонента координаты, м
    m_y, // y-компонента координаты, м
    m_u, // x-компонента скорости, м/с
    m_v; // y-компонента скорости, м/с

  // вектор источников (правых частей уравнений - ОДУ)
  double *m_S;

  // вектор консервативных переменных
  double *m_U;

 public: // открытые методы класса

  Mass();               // конструктор; имитирует загрузку исх. данных
  ~Mass();              // деструктор; освобождает дин. память
```

```
  void init();          // метод инициализации по исходным данным
  void update(double); // метод обновления параметров на шаге
  void output(double); // метод вывода решения на шаге
};

#endif // _MASS_H_
```

```
#------------------------#
# Makefile для Примера 5 #
#------------------------#

# компилятор GNU C++
CXX = g++
# флаги компилятора (оптимизация, стандарт ANSI, все предупреждения)
CXXFLAGS = -O2 -ansi -Wall
# компоновщик (тот же GNU C++)
LINKER = $(CXX)
# флаги компоновщика
LFLAGS = -s
# список объектных файлов
OBJECTS = main.o Mass.o
# имя исполняемого файла программы
PROGRAM = program

# выполнение программы для обновления файла результатов расчета
test: output.dat
output.dat: $(PROGRAM)
  ./$(PROGRAM) > output.dat

# компоновка программы из объектных файлов
$(PROGRAM): $(OBJECTS)
  g++ $(LFLAGS) $(OBJECTS) -o $(PROGRAM)

# компилирование исходных текстов модулей на C++
main.o: main.cpp Mass.h
  g++ -c $(CXXFLAGS) main.cpp -o main.o
Mass.o: Mass.cpp Mass.h
  g++ -c $(CXXFLAGS) Mass.cpp -o Mass.o

# очистка рабочего каталога
clean:
  rm -f *~ $(PROGRAM) $(OBJECTS) output.dat

#== end of Makefile ==#
```

```
// "Mass.cpp"
//
// Класс объекта МАССА (тело массой $m$, брошенное под углом 45^{0}
// к горизонту, движение которого рассчитывается численным решением
// системы из четырех ОДУ первого порядка динамики материальной точки (МТ)
// с учетом силы лобового сопротивления (по закону сохранения импульса).
//  Расчет по шагам идет методом Эйлера 1-го порядка, направление
// координатных осей: x - вправо, y - вверх.
//
// (C) Иванов И.И., гр. ДВ-999
//

using namespace std;

#include <iostream>
#include <cmath>

#include "Mass.h"

// конструктор
Mass::Mass()
{
  // имитируем загрузку исходных данных
  m_m   = 1.;    // масса, кг
  m_d   = 0.1;   // диаметр, м
  m_Cx  = 0.4;   // постоянный коэффициент сопротивления, б/р
  m_rho = 1.293; // постоянная плотность воздуха, кг/м^{3}
  m_g   = -9.81; // ускорение свободного падения $g_{x}$, м/с^{2}
  m_x0  = 0.;    // x-компонента начальной координаты, м
  m_y0  = 0.;    // y-компонента начальной координаты, м
  m_u0  = 50.;   // x-компонента начальной скорости, м/с
  m_v0  = 50.;   // y-компонента начальной скорости, м/с

  // выделяем динамическую память (не проверяя результата!)
  m_S = new double [ULEN];
  m_U = new double [ULEN];
}

// деструктор: высвобождает динамическую память
Mass::~Mass()
{
  delete [] m_S;
  delete [] m_U;
}

// метод инициализации (нач. условий) задачи по исходным данным
void Mass::init()
{
  // первичные переменные на старте, (0)-м временнОм слое
  m_x = m_x0;
  m_y = m_y0;
```

```
  m_u = m_u0;
  m_v = m_v0;

  // вектор консервативных переменных на старте (условно и координаты)
  m_U[0] = m_x;
  m_U[1] = m_y;
  m_U[2] = m_m * m_u; // x-компонента количества движения, кг * м/с
  m_U[3] = m_m * m_v; // y-компонента...
}

// метод обновления параметров на шаге
void Mass::update(double dt)
{
  // рассчитаем силы сопротивления движению (для уравнений импульса)
  static double V, F;
  V = sqrt(m_u * m_u + m_v * m_v);
  F = m_Cx * (M_PI_4 * m_d * m_d) * m_rho * V * V / 2.;

  // вычислим правые части на старом слое
  m_S[0] = m_u;
  m_S[1] = m_v;
  m_S[2] = m_m * m_g - F * m_u / V; // в проекциях
  m_S[3] = 0.        - F * m_v / V;

  // обновление консервативных переменных (м. Эйлера 1-го порядка)
  for (int k = 0; k < ULEN; k++) { m_U[k] = m_U[k] + dt * m_S[k]; }

  // 'раскодируем' вектор консервативных переменных до первичных
  m_x = m_U[0];
  m_y = m_U[1];
  m_u = m_U[2] / m_m;
  m_v = m_U[3] / m_m;
}

// метод вывода решения на шаге
void Mass::output(double t)
{
  // установим режим вывода с тремя знаками после запятой
  streamsize precnOld = cout.precision(3);

  // выводим в установленном режиме, вывод всех 3 знаков ,->
  cout << fixed <<   t << ' '
       << fixed << m_x << ' ' << fixed << m_y << ' '
       << fixed << m_u << ' ' << fixed << m_v << endl;

  // восстановим старый режим
  cout.precision(precnOld);
}

// end of "Mass.cpp"
```

Составители: ВАХИТОВ Юрий Рашитович
ЗАГАЙКО Сергей Андреевич

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ В САПР

Лабораторный практикум
по дисциплине
«Основы систем автоматизированного проектирования»

Подписано в печать 02.04.2007. Формат 60x84 1/16.
Бумага офсетная. Печать плоская. Гарнитура Times New Roman Cyr.
Усл. печ. л. 2,4. Усл. кр.-отт. 2,4. Уч.-изд. л. 2,3.
Тираж 100 экз. Заказ № ___.
ГОУ ВПО Уфимский государственный авиационный технический университет
Центр оперативной полиграфии УГАТУ
450000, Уфа-центр, ул. К. Маркса, 12