

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Локтионова Оксана Геннадьевна  
Должность: проректор по учебной работе  
Дата подписания: 05.08.2023 10:28:01  
Уникальный программный ключ:  
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

**МИНОБРАЗОВАНИЯ РОССИИ**

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра информационных систем и технологий

УТВЕРЖДАЮ  
Проректор по учебной работе  
О.Г. Локтионова  
« 15 » 08 2019г.



ТЕОРИЯ СИСТЕМ И СИСТЕМНЫЙ АНАЛИЗ  
Методические указания  
по выполнению практических работ  
для студентов, обучающихся по направлениям 09.03.02, 09.03.03,  
02.03.03.

Курск 2019

УДК 681.3(075)

Составитель: Л.А. Лисицин

Рецензент

- Кандидат технических наук, доцент Халин Ю.А.

Теория систем и системный анализ [Текст]: методические указания по выполнению практических работ / Юго-Зап. гос. ун-т; сост.: Л.А. Лисицин. Курск, 2017. 34 с.: ил. 24. табл. 3. Библиогр. с. 34.

Содержат сведения по технологиям сбора, хранения, обработки и передачи информации. Материал ориентирован на практическую работу студентов в компьютерной среде.

Отражен порядок выполнения практических работ и правила оформления отчетов.

Методические указания предназначены для студентов, обучающихся по направлениям 09.03.02 «Информационные системы и технологии» и 09.03.03 «Прикладная информатика» и 02.03.03 «Математическое обеспечение и администрирование информационных систем». Могут использоваться также студентами, обучающимися по направлениям связанным с информационными системами.

Методические указания соответствуют требованиям программы, утвержденной учебно-методическим объединением по специальностям «Информационные системы».

Текст печатается в авторской редакции

Подписано в печать 15.03.19 Формат 60x84 1/16.

Усл.печ. л. 1,8. Уч.-изд. л. 1,6 Тираж 50 экз. Заказ 223

Бесплатно.

Юго-Западный государственный университет. 305040, г. Курск,  
ул. 50 лет Октября, 94.

## Оглавление

Практическая работа №1 .....	4
Тема: Структура системы (строение, расположение, порядок). Иерархия как упорядоченность компонентов по степени важности.....	4
Практическая работа №2 .....	11
Тема: Методы описания систем. Описание функционирования системы. Описательная модель «Бросание мячика в стенку».....	11
Практическая работа №3 .....	16
Тема: Построение формальной модели системы «Бросание мячика в стенку».....	16
Практическая работа №4 .....	18
Тема: Компьютерная модель движения физического тела .....	18
Практическая работа №5 .....	22
Тема: Компьютерный эксперимент. Анализ и корректировка исследуемой модели. ....	22
Практическая работа №6 .....	26
Тема: Вероятностные модели. Построение информационной модели с использованием метода Монте-Карло. ....	26
Практическая работа №7 .....	30
Тема: Модели логических устройств компьютера. Полусумматор.....	30
Практическая работа №8. ....	33
Тема: Построение и использование оптимизационной модели в электронных таблицах. ...	33
Практическая работа №9. ....	37
Тема: Графы и их исследование. Описание графа с помощью матрицы смежности. ....	37
Практическая работа № 10. ....	41
Тема: Построение остовного связного графа дерева. Вес графа. ....	41
Библиографический список.....	44

## Практическая работа №1

**Тема программы: Основные понятия теории систем.**

**Тема: Структура системы (строение, расположение, порядок).**

**Иерархия как упорядоченность компонентов по степени важности.**

**Цель:** изучение термина «система». Научиться системным представлениям не только как к удобному обобщающему понятию, но и как к средству постановки задач с большой неопределенностью.

### **Теоретические основы**

Обычно термин «система» универсален и используется, когда невозможно что-то продемонстрировать, изобразить, представить математическим выражением и нужно подчеркнуть, что это будет большим, сложным, не полностью сразу понятным (с неопределенностью) и целым, единым. Под **системой**, понимается объект свойства которого не сводятся без остатка к свойствам составляющих его дискретных элементов (неаддитивность свойств).

Любой элемент системы можно рассматривать как самостоятельную систему (математическую модель, описывающую какой - либо функциональный блок, или изучаемой аспект), как правило более низкого порядка. Каждый элемент системы описывается своей функцией. Под **функцией** понимается вещественно-энергетические и информационные отношения между входными и выходными процессами. Если такой элемент обладает внутренней структурой, то его называют подсистемой.

**Структура системы** — организация связей и отношений между подсистемами и элементами системы, а также, собственно, состав этих подсистем и элементов, каждому из которых обычно соответствует определенная функция.

Например, персональный компьютер можно считать системой, где есть следующие элементы 1. блок питания 2. материнская плата 3. оперативная память 4. центральный процессор 5. видеокарта 6. сетевая карта 7. проигрыватель оптических дисков 8. модем. Операционную систему компьютера можно считать системой. Самолет типа Boeing является системой. Ракета-носитель является системой, в которой присутствует система управления, система

навигации, разгонный блок, двигатель первой ступени, двигатели второй ступени, двигатели третьей ступени и др. элементы



Различаются структуры одноуровневые и многоуровневые. *Экономические системы*, характеризуются, как правило, многоуровневой *иерархической структурой*; им свойственна также полиструктурность, т.е. взаимопереплетение разнокачественных подсистем, образующих несколько связанных между собой иерархических структур (производственно-технологических, территориальных, институциональных, социальных и др.). Различают также системы с постоянной и переменной структурами, причем структура экономической системы обычно относится ко второму виду: она подвижна, формируется применительно к условиям функционирования.

Примеры описания систем с помощью формул:

O1. Система есть нечто целое:

$$S=A(1,0).$$

Это определение выражает факт существования и целостность. Двоичное суждение  $A(1,0)$  отображает наличие или отсутствие этих качеств.

O2. Система есть организованное множество (Темников Ф. Е.):

$$S=(\text{орг}, M),$$

где орг - оператор организации; M - множество.

О3. Система есть множество вещей, свойств и отношений (Уемов А. И.):

$$S = (\{t\}, \{n\}, \{r\}),$$

где  $t$  - вещи,  $n$  - свойства,  $r$  - отношения.

О4. Система есть множество входов, множество выходов, множество состояний, характеризуемых оператором переходов и оператором выходов:

$$S = (X, Y, Z, H, G),$$

где  $X$  - входы,  $Y$  - выходы,  $Z$  - состояния,  $H$  - оператор переходов,  $G$  - оператор выходов. Это определение учитывает все основные компоненты, рассматриваемые в автоматике.

Иерархичность можно выделить как закономерность построения всего мира и любой выделенной из него системы. *Иерархическая упорядоченность пронизывает все, начиная от атомно-молекулярного уровня и кончая человеческим обществом.* Иерархичность как закономерность заключается в том, что закономерность целостности проявляется на каждом уровне иерархии. Благодаря этому *на каждом уровне возникают новые свойства, которые не могут быть выведены как сумма свойств элементов.*

*Следствия иерархичности систем:*

1. с помощью иерархических представлений можно *отображать системы с неопределенностью.*

2. Построение иерархической структуры зависит от цели: для многоцелевых ситуаций можно построить несколько иерархических структур, соответствующих разным условиям, и при этом *в разных структурах могут принимать участие одни и те же компоненты.*

3. Даже при одной и той же цели, если поручить формирование иерархической структуры разным исследователям, то в зависимости от их предшествующего опыта, квалификации и знания *системы они могут получить разные иерархические структуры, т. е. по-разному разрешить качественные изменения на каждом уровне иерархии.*

**Порядок выполнения работы:**

1. Выбор задания.

Выберите систему, для которой будете составлять описание.

Примеры систем:

- технические устройства и комплексы (автомобили, компьютеры, телевизоры и т.д.);
- организации (предприятия, гостиницы, рестораны, магазины и т.д.);
- биологические системы (человек, животные, растения и т.д.);
- информационные системы (автоматизированные системы, программные комплексы, информационно-управляющие системы и т.д.);
- социальные, социально-экономические системы (система выборов, система безопасности, транспортная система региона и т.д.).

## 2. Классификация системы.

Классифицируйте систему по следующим признакам:

- происхождение (естественные, искусственные, смешанные);
- сложность (простые, сложные);
- изолированность (открытые, закрытые);
- характер функционирования (стабильные, самостабилизирующиеся, самоорганизующиеся);
- способ задания целей (цели задаются извне, цели формируются внутри);
- способ управления (самоуправляемые, управляемые извне, с комбинированным управлением).

## 3. Построение иерархии состава.

Выделите основные подсистемы исследуемой системы. В рамках каждой из них выделите более мелкие подсистемы и элементы. Представьте компоненты системы в виде иерархии. Пример иерархии состава компьютера представлен на рис. 1.1.

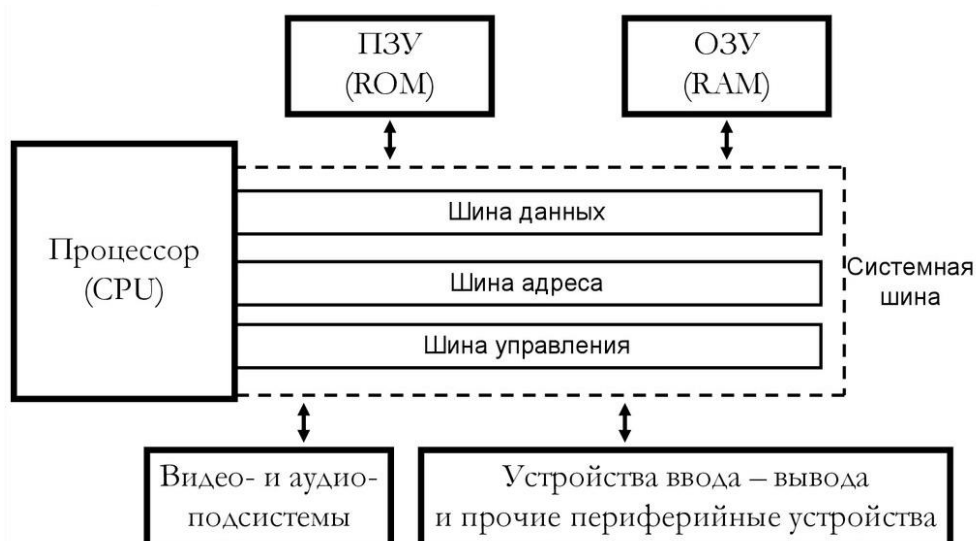


Рисунок 1.1 – Иерархия состава компьютера

Дайте краткое описание основных подсистем и элементов.

#### 4. Описание сущностных свойств системы.

Опишите сущностное свойство системы и его внешнее проявление (явление). Определите, является ли данное свойство эмерджентным. Ответ обоснуйте.

Например, для ПК сущностное свойство – способность воспроизводить звук, закодированный в двоичном коде. Явление – звучание приемника, передающего радиопередачу. Данное свойство является эмерджентным, т.к. ни один из компонентов ПК по отдельности не обладает им: монитор, память и т.д.

#### 5. Описание структуры системы и ее взаимодействия с окружением.

Выделите объекты окружающей среды. Дайте им краткую характеристику. Опишите их связь с исследуемой системой.

Например, для радиоприемника основными объектами среды являются:

- передатчик радиостанции, передающий радиоволны, которые воспринимает приемник;
- пользователь, воспринимающий звучание радиоприемника и управляющий параметрами (частота радиоволн, громкость звучания) посредством механических воздействий на регуляторы).

Составьте схему взаимодействия компонент системы, а также схему взаимодействия со средой (это может быть одна общая схема).



Если система слишком большая и сложная, можете составить схему для некоторой подсистемы. Опишите внутренние и внешние связи.

Пример схемы взаимодействия компонент радиоприемника друг с другом и с окружающей средой представлен на рис. 1.2.

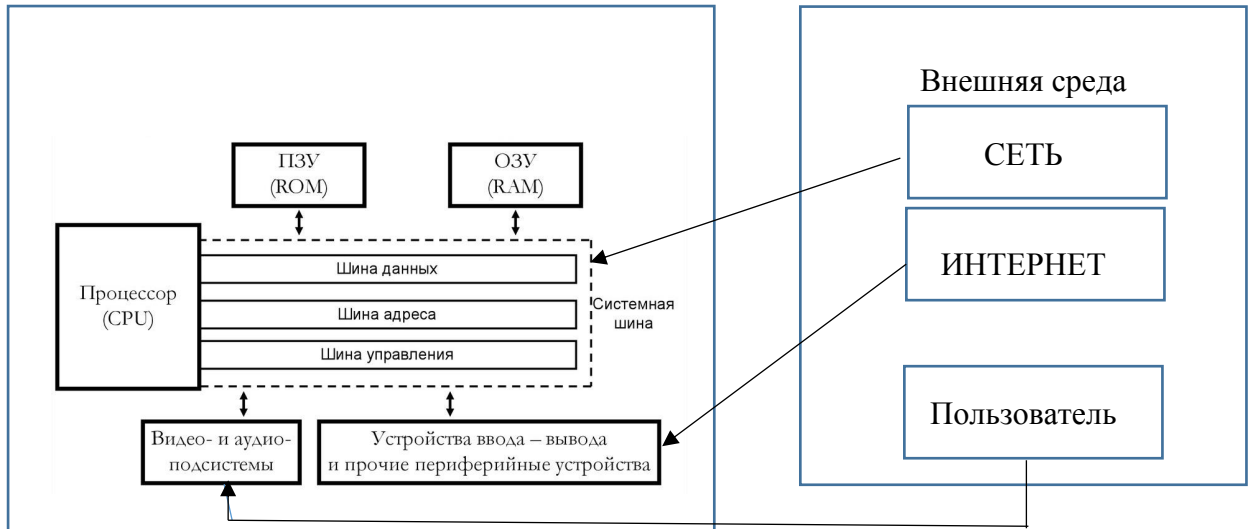


Рисунок 1.2 – Структура радиоприемника

#### 6. Составление отчета.

В отчет должны войти:

- наименование системы, ее классификация;
- иерархия состава системы (схема), описание подсистем и элементов;
- описание сущностных свойств системы; □ описание структуры системы и ее взаимодействия с окружением (схема).

#### Контрольные вопросы

1. Какие основные свойства сложных систем?
2. Охарактеризуйте уровни описания структуры системы.
3. Выделите основные фазы (поколения) эволюции информационных систем. Укажите основные достоинства систем «клиент-сервер».
4. Приведите классификацию базовых информационных процессов.
5. Какие виды иерархии характерны для информационных систем?
6. Укажите отличительные признаки большой системы.
7. Каковы основные свойства систем?

8. Что такое сложная система? Каковы отличия от простых систем?
9. Что такое модель структуры системы?
10. Как можно представить процесс функционирования любой системы?
11. Чем характеризуется устойчивость систем?
12. Каковы основные особенности управления сложными системами?

## Практическая работа №2

**Тема программы: Основные понятия теории систем.**

**Тема: Методы описания систем. Описание функционирования системы. Описательная модель «Бросание мячика в стенку»**

**Цель:** Получить практические навыки в описании функционирования системы во времени и управления системой

### **Теоретические основы**

При исследовании моделей на компьютере различают 5 этапов разработки.

Использование компьютера для исследования информационных моделей различных объектов и систем позволяет изучить их изменения в зависимости от значения тех или иных параметров. Процесс разработки моделей и их исследование на компьютере можно разделить на несколько основных этапов.

Описательная информационная модель. На первом этапе исследования объекта или процесса обычно строится описательная информационная модель на естественном языке. Такая модель выделяет существенные, с точки зрения целей проводимого исследования, свойства (параметры) объекта, а несущественными свойствами пренебрегает.

Формальная модель. На втором этапе создается формальная модель, т. е. описательная информационная модель записывается с помощью какого-либо формального языка. В такой модели с помощью формул, уравнений, неравенств и т. д. фиксируются формальные соотношения между исходными и искомыми величинами, а также накладываются ограничения на допустимые значения этих величин. Однако далеко не всегда удается найти формулы, явно выражающие искомые величины через исходные. В таких случаях используются приближенные математические методы, позволяющие получать результаты с заданной точностью.

Компьютерная модель. На третьем этапе необходимо формализованную информационную модель преобразовать в компьютерную модель, т. е. выразить ее на понятном для компьютера языке. Существуют два пути решения этой задачи: • создание проекта на одном из языков программирования; • построение компьютерной модели с использованием некоторого приложения, например электронных таблиц.

В процессе создания компьютерной модели полезно разработать удобный графический интерфейс, который позволит визуализировать формальную модель, а также реализовать интерактивный диалог человека с компьютером на этапе исследования модели.

Компьютерный эксперимент. Четвертый этап исследования информационной модели состоит в проведении компьютерного эксперимента. Если компьютерная модель существует в виде программы на одном из языков программирования, ее нужно запустить на выполнение и получить результаты. Если компьютерная модель исследуется в приложении, например в электронных таблицах, можно провести сортировку или поиск данных, построить диаграмму или график и т. д.

Анализ полученных результатов и корректировка исследуемой модели. Пятый этап представляет собой анализ полученных результатов и корректировку исследуемой модели. В случае отличия результатов, полученных при исследовании информационной модели, от измеренных параметров реальных объектов можно сделать вывод, что на предыдущих этапах построения модели были допущены ошибки или неточности. Например, при построении описательной качественной модели могут быть неправильно отобраны существенные свойства объектов, в процессе формализации могут быть допущены ошибки в формулах и т. д. В таких случаях необходимо провести корректировку модели, причем уточнение модели может проводиться многократно, пока анализ результатов не покажет их соответствие изучаемому объекту,

Визуализация формальных моделей. В процессе исследования формальных моделей часто производится их визуализация. Для визуализации алгоритмов используются блок-схемы, пространственных соотношений параметров объектов — чертежи, моделей электрических цепей — электрические схемы. При визуализации формальных моделей с помощью анимации может отображаться динамика процесса, производится построение графиков изменения величин и т. д. В настоящее время широкое распространение получили компьютерные интерактивные визуальные модели. В таких моделях исследователь может менять начальные условия и параметры протекания процессов и наблюдать изменения в поведении модели.

## Порядок выполнения работы:

1. Описание функционирования системы в пространстве состояний.

Выделите характеристики (параметры) системы. Параметры могут быть сгруппированы по типам: физические характеристики, технические характеристики, экономические показатели и т.д.

Например, для радиоприемника могут быть выделены следующие параметры:

- физические (размер, цвет, материал, дизайн корпуса);
- технические (напряжение питания, диапазон радиоволн);
- параметры производителя (компания-производитель, страна, марка, дата производства, гарантийный срок); и т.д.

Из множества параметров выделите те, которые могут характеризовать поведение (функционирование) системы, т.е. которые изменяются во времени. Опишите различные состояния системы, указав конкретные значения параметров. Пример описания состояний ПК приведен в таблице 2.1.

Таблица 2.1 – Описание состояний радиоприемника

Параметр	I состояние ПК в режиме эксплуатации	II состояние ПК в режиме ожидания	III состояние Дежурный режим
Потребление эл\энергии	200 Вт	50Вт	1 Вт
Свечение монитора	200Лм	0	0
Звук	Есть	Есть	нет

Укажите события, вызывающие переход из состояния в состояние. Например, для состояний, приведенных в таблице 2.1:

- переход из состояния I в II – настройка регулятора;
- переход из II в III – настройка уровня звука.

3. Описание жизненного цикла.

Выделите этапы жизненного цикла исследуемой системы, характеризующиеся определенными предсказуемыми состояниями. Опишите каждый этап. Пример описания этапов жизненного цикла ПК приведен в таблице 2.2.

Таблица 2.2 – Описание этапов жизненного цикла радиоприемника

№	Этап ЖЦ	Характеристика
I	До эксплуатации	Эксплуатируется редко (только в тестовом режиме), все элементы в хорошем состоянии
II	Активная эксплуатация	Активно эксплуатируется, время от времени состояние некоторых элементов может ухудшаться (повреждения корпуса, ...)
III	Утилизация	Не эксплуатируется ввиду морального или физического износа (сломан)

#### 4. Описание управления системой.

Определите основную цель системы. Цель может быть имманентной (внутренне присущей) или задаваемой извне, субъективной или объективной. Если система является неживым объектом, цель, как правило, определяется пользователем. Например, для ПК цель, задаваемая владельцем, – получить качественное изображение выбранной ТВ передачи.

Определите, кто (что) и как управляет системой, с помощью каких управляющих воздействий осуществляется управление, используется ли в процессе управления обратная связь и если используется, то каким образом.

Например, управление ПК осуществляет пользователь (внешнее управление) посредством изменения мышки. Управляющие воздействия: экранный интерфейс. Обратная связь – определение на слух, наличия помех и уровня громкости звука.

**Описательная модель.** Сначала построим описательную модель процесса движения тела с использованием объектов, понятий и законов физики, т. е. в данном случае идеализированную модель

движения объекта. Из условия задачи можно сформулировать следующие основные допущения: • мячик мал по сравнению с Землей, поэтому его можно считать материальной точкой; • изменение высоты мячика мало, поэтому ускорение свободного падения можно считать постоянной величиной  $g$  —  $9,8 \text{ м/с}^2$ , следовательно, движение по оси  $Y$  можно считать равноускоренным; • скорость бросания мячика мала, поэтому сопротивлением воздуха можно пренебречь, следовательно, движение по оси  $X$  можно считать равномерным.

### 5. Составление отчета.

В отчет должны войти:

- наименование системы;
  - описание функционирования системы в пространстве состояний (характеристики, состояния, события);
  - описание этапов жизненного цикла системы;
- описание управления системой (цель, управляющие воздействия, обратная связь).

Контрольные вопросы

1. Охарактеризуйте уровни описания структуры системы.
2. Каковы основные свойства систем?
3. Что такое и для чего нужна модель «черный ящик»?
4. Что такое модель состава системы?
5. Что такое модель структуры системы?

## Практическая работа №3

Тема программы: Основные понятия теории систем.

Тема: Построение формальной модели системы «Бросание мячика в стенку».

**Цель:** Получить практические навыки в формировании базовых моделей («черного ящика», состава, структуры) системы и описании их на формальном языке.

### Построение и исследование физических моделей

1. Построение формальной модели движения тела, брошенного под углом к горизонту

Рассмотрим процесс построения и исследования модели на конкретном примере движения тела, брошенного под углом к горизонту.

Содержательная постановка задачи «Бросание мячика в стенку». В процессе тренировок теннисистов используются автоматы по бросанию мячика. Необходимо задать автомату необходимую скорость и угол бросания мячика для попадания в стенку определенной высоты, находящуюся на известном расстоянии.

Формальная модель. Для формализации модели обозначим величины: • начальную скорость мячика —  $V_0$ , • угол бросания мячика —  $\alpha$ ; • высоту стенки —  $h$ ; • расстояние до стенки —  $s$ .

Изобразим график движения мячика (рис. 1.1).

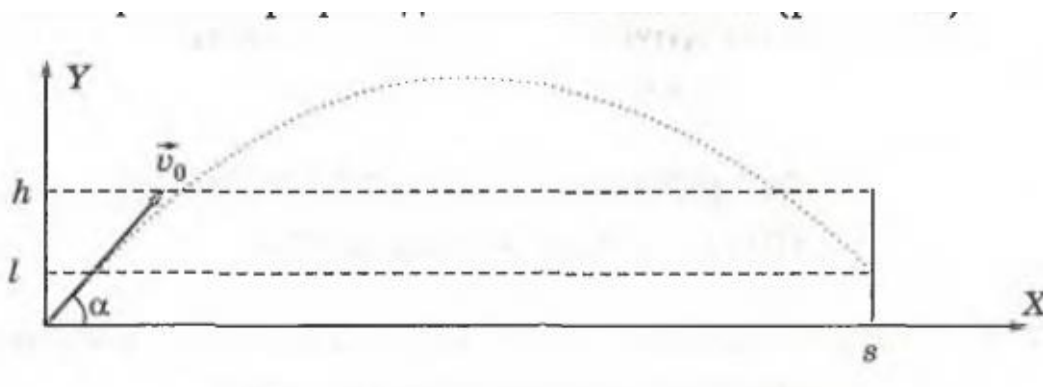


Рис. 3.1. Бросание мячика в стенку

Используем известные из курса физики формулы равномерного и равноускоренного движения для определения координат мячика. Дальность  $x$  и высоту  $y$  при заданной начальной скорости  $V_0$  и угле



бросания а для любого момента времени  $t$  можно вычислить по формулам:  $x = V_0 \cdot \cos \alpha \cdot t$ ,  $y = V_0 \cdot \sin \alpha \cdot t - g \cdot t^2 / 2$ . Чтобы определить, попадет ли мячик в стенку, необходимо вычислить его координату  $y$  в момент времени, когда он будет находиться на расстоянии  $s$ . Из первой формулы выражаем время, которое понадобится мячику, чтобы преодолеть расстояние  $s$ :  $t = s / v_0 \cdot \cos \alpha$ . Подставляем это значение времени  $t$  в формулу для вычисления координаты  $y$ . Получаем  $I$  — высоту мячика над землей на расстоянии  $s$  (см. рис. 1.1):

$$I = s \cdot \operatorname{tg} \alpha - g \cdot s^2 / 2 \cdot V_0^2 \cdot \cos^2 \alpha. \quad (1.2)$$

Формализуем теперь условие попадание мячика в мишень. Попадание произойдет, если значение высоты мячика  $I$  будет удовлетворять условию в форме неравенства:  $0 < I \leq h$ . Если  $I < 0$ , то это означает «недолет», а если  $I > h$ , то это означает «перелет».

. Компьютерная модель движения тела на языке Visual Basic

На основе формальной модели, описывающей движение тела, брошенного под углом к горизонту, создадим компьютерную модель с использованием системы программирования Visual Basic.

### 3.2 Составление отчета.

В отчет должны войти:

- наименование системы;
- описание функционирования системы в пространстве состояний (характеристики, состояния, события);
- построить формальную модель, описывающую движение тела, для построения компьютерной модели.

### Контрольные вопросы

1. Охарактеризуйте уровни описания структуры системы.
2. Каковы основные свойства систем?
3. Что такое модель состава системы?
4. Что такое модель структуры системы?

## Практическая работа №4

**Тема программы: Основные понятия теории систем.**

**Тема: Компьютерная модель движения физического тела**

**Цель:** Построение компьютерной программы модели движения физического тела.

Формальная модель. Для формализации модели обозначим величины: • начальную скорость мячика —  $V_0$ , • угол бросания мячика —  $\alpha$ ; • высоту стенки —  $H$ ; • расстояние до стенки —  $s$ .

Создадим сначала графический интерфейс проекта «Бросание мячика в стенку» (рис. 1.2).

1. Разместить на форме:

- четыре текстовых поля для ввода значений: `TextBox1` — начальной скорости, `TextBox2` — угла бросания мячика, `TextBox3` — расстояния до стенки и `TextBox4` — высоты стенки;
- надпись `Label 1` для вывода высоты мячика на заданном расстоянии;
- надпись `Label 2` для вывода текстового сообщения о результатах броска;
- десять надписей для вывода имен переменных и единиц измерения;
- кнопку `Button 1` для запуска событийной процедуры вычисления результатов бросания мячика;
- кнопку `Button2` для демонстрации траектории движения мячика.

Обработчик события — вычисление результатов бросания мячика

2. Создать программный код обработчика события, который определяют попадание мячика в стенку. В этом коде:

Проект «Бросание мячика в стенку» на языке Visual Basic

- объявить вещественные константы одинарной точности  $G$  (ускорение свободного падения  $g$ ) и  $Pi$  (число  $\pi$ );
- объявить вещественные переменные одинарной точности  $V_0$  (начальная скорость  $V_0$ ),  $\alpha$  (угол бросания  $\alpha$ ),  $S$  (расстояние до стенки  $s$ ),  $H$  (высота стенки  $h$ ) и  $L$  (высота мячика  $Z$ );

- присвоить переменным  $V_0$ ,  $A$ ,  $S$ ,  $H$  значения, введенные в текстовые поля, с использованием функции преобразования строки в вещественное число  $Val()$ ;

- вычислить высоту мячика  $L$  на заданном расстоянии по формуле (1.2);

- вывести высоту мячика  $L$  на надпись  $Label\ 1$ ;

- вывести текстовое сообщение о результатах броска на надпись  $Label\ 2$  с использованием оператора  $Select\ Case$ , в котором в качестве условия проверяется значение переменной  $L$ .

В языке программирования  $Visual\ Basic$  аргументы тригонометрических функций  $Sin()$ ,  $Cos()$  и  $Tan()$  задаются в радианах, а угол бросания мячика мы будем вводить в градусах. Поэтому необходимо преобразовать значения углов из градусов в радианы с использованием константы  $Pi$ .

```
Const G As Single = 9.81
```

```
Const Pi As Single = 3.14
```

```
Dim VO, A, S, L, H As Single
```

```
Private Sub Button1_Click (...)
```

```
' Ввод начальных значений
```

```
VO=Val(TextBox1.Text)
```

```
A=Val(TextBox2.Text)
```

```
S=Val(TextBox3.Text)
```

```
H=Val(TextBox4.Text)
```

```
' Попадание в стенку
```

```
L=S*Math.Tan(A*Pi/180)- (G*SA2 )/ (2*VOA2*  
Math.Cos(A*Pi/180)^2)
```

```
L a b e ll.Text=L
```

```
Select Case L
```

```
Case <0
```

```
Label2.Text="Недолет"
```

```
Case Is >H
```

```
Label2.Text="перелет"
```

```
Case Else Label2.
```

```
Tex^"Попадание"
```

```
End Select End Sub
```

Обработчик события — демонстрация траектории движения мячика

Для визуализации компьютерной модели построим в графическом поле траекторию движения мячика.

3. Поместить дополнительно на форму графическое поле PictureBox1. С помощью диалогового окна Свойства установить с использованием свойства Size размер поля, например 400;220.

В обработчике события осуществим преобразование компьютерной системы координат графического поля в математическую систему координат, удобную для построения траектории движения. Нарисуем оси координат и нанесем на них шкалы.

Проект «Система координат» Информатика и ИКТ-9 | §Г

4. В математической системе координаты находятся в диапазонах  $0 < X < 400$  и  $-20 < Y < 200$ . Траектория движения мячика, скорее всего, будет в диапазоне координат  $0 < X < 40$  м и  $0 < Y < 20$  м. Следовательно, необходимо увеличить масштаб графика в 10 раз:

- координаты точек графика необходимо умножить на 10;
- значения шкал осей разделить на 10.

Построение траектории осуществить в цикле со счетчиком (координата X) с использованием метода рисования точки Draw Ellipse(Pen1, X\*10, Y\*10, 1, 1), в котором координатами точки являются координаты мячика.

```
Dim Graph1 As Graphics
Dim Pen1 As New Pen (Color.Black, 4)
Dim drawBrush As New SolidBrush(Color.Black)
Dim drawFont As New Font("Arial", 10)
Dim X, Y, T As Single Private Sub Button2_Click (...)
Graph1=Me.PictureBox1.CreateGraphics() Graph1.Clear(Color.White)
'Печать шкал математической системы координат 'в
компьютерной системе координат For X=0 To 400 Step 50
Graph1. DrawString(X/10, drawFont, drawBrush, X, 200)
Next X
For Y=20 To 220 Step 50
Graph1. DrawString( (Y-20)/10, drawFont, drawBrush, 0, 220-Y)
Next Y
'Преобразование компьютерной системы координат 'в
математическую систему координат
Graph1. ScaleTransform(1, -1)
'Поворот оси Y
Graph1.TranslateTransform (0, -200)
'Сдвиг по оси Y 'Рисование осей математической системы
координат Graph1.DrawLine(Pen1, 0, 0, 400, 0)
```

```
' Ось X
Graphl.DrawLine (Pen1, 0, -20, 0, 200)
'Ось Y ' Стенка
Graphl.DrawLine (Pen1, S*10, 0, S*10, H*10)
'Построение траектории движения мячика
For T=0 To 10 Step 0.1
Y=V0*Math.Sin(A*Pi/180)*T-G*T*T/2
X=V0 *Math.Cos(A*Pi/180) *T
Graphl.DrawEllipse(Pen1, X*10, Y*10, 1, 1)
Next T
End Sub
```

### 5. Составление отчета.

В отчет должны войти: рабочая программа на одном из языков программирования со скриншотами.

### Контрольные вопросы

1. Охарактеризуйте интерфейс программы.
2. Почему удобно визуальное программирование?

## Практическая работа №5

**Тема программы: Основные понятия теории систем.**

**Тема: Компьютерный эксперимент. Анализ и корректировка исследуемой модели.**

**Цель:** Получить практические навыки компьютерного эксперимента систем.

Компьютерный эксперимент. Введем произвольные значения начальной скорости и угла бросания мячика. Скорее всего, попадания в стенку не будет. Меняя один из параметров, например угол, произведем пристрелку, используя известный артиллерийский прием «взятие в вилку», в котором применяется эффективный метод «деление пополам». Сначала найдем угол, при котором мячик перелетит стенку, затем угол, при котором мячик не долетит до стенки. Вычислим среднее значение углов, составляющих «вилку», и проверим, попадет ли при этом значении мячик в стенку. Если он попадет в стенку, то задача выполнена, если не попадет, то рассмотрим новую «вилку» и т. д.

5. Запустить проект и ввести значения начальной скорости, угла бросания, расстояния до стенки и ее высоты. Щелкнуть по кнопкам Вычислишь и Показать. На надписи будут выведены результаты, а в графическом поле появится траектория движения тела (см. рис. 1.2). Подобрать значения начальной скорости и угла бросания мячика, обеспечивающие его попадание в стенку.

Построение и исследование информационных моделей 21

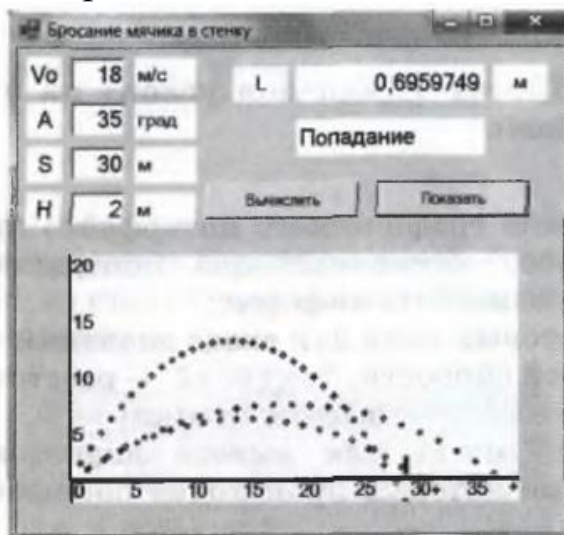


Рис. 1.2. Компьютерный эксперимент по бросанию мячика в стенку

Например, при скорости бросания мячика  $V_0 = 18$  м/с и угле бросания  $\alpha = 35^\circ$  мячик попадет в стенку высотой  $h = 2$  м и находящуюся на расстоянии  $s = 30$  м на высоте  $I = 0,6959749$  м.

### 1.3.4. Приближенное решение уравнений в электронных таблицах

Возможности электронных таблиц не ограничиваются вычислениями по формулам и построением диаграмм и графиков. С помощью метода Подбор параметра можно приближенно с заданной точностью решать уравнения.

Задача. Найти корень уравнения  $x^3 - \cos x = 0$  приближенными методами (графическим и с помощью метода Подбор параметра).

Представим функцию в табличной форме, построим ее график, который позволит определить корень уравнения грубо приближенно. *Л h t e i*

1. Представить заданное уравнение в табличной форме (рис. 1.16).

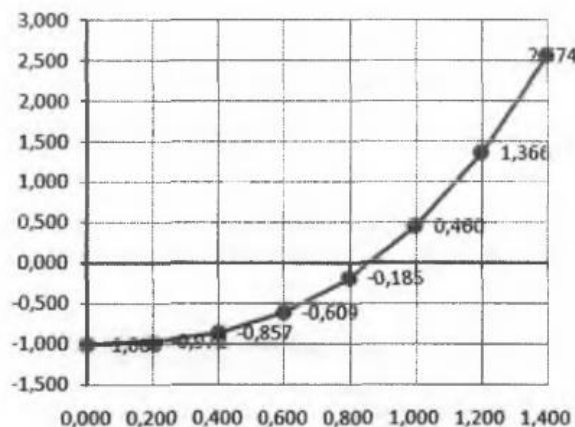
	A	B	C	D	E	F	G	H	I
1	x	0,000	0,200	0,400	0,600	0,800	1,000	1,200	1,400
2	$y = x^3 - \cos x$	-1,000	-0,972	-0,857	-0,609	-0,185	0,460	1,366	2,574

Рис. 1.16. Табличное представление уравнения

Проект «Приближенное решение уравнения» в электронных таблицах

Построение и исследование информационных моделей 45

2. Для грубо приближенного определения корня построить диаграмму типа График. По графику грубо приближенно можно определить, что  $x \approx 0,8$  (рис. 1.17).



- 3.

Рис. 1.17. График функции

Для поиска решения с заданной точностью используем метод Подбор параметра. Точность подбора зависит от заданной точности представления чисел в ячейках таблицы (например, до трех знаков после запятой). Методом подбора параметра необходимо определить значение аргумента  $x$  (ячейка F1), при котором значение функции  $y$  (ячейка F2) будет равно нулю.

3. Выделить ячейку со значением функции F2 и ввести команду [Сервис-Подбор параметра...] в Microsoft Excel 2003 и OpenOffice.org Calc или команду [Анализ "что-если"- Подбор параметра...] в Microsoft Excel 2007.

4. В диалоговом окне Подбор параметра (рис. 1.18):

- в поле Установить в ячейке: ввести адрес ячейки \$F\$2;
- в поле Значение: ввести требуемое значение функции (в данном случае 0);
- в поле Изменяя значение ячейки: ввести адрес ячейки \$F\$1, в которой будет производиться подбор значения аргумента.

Щелкнуть по кнопке ОК.

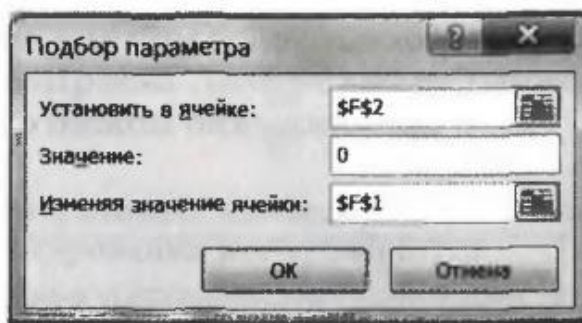


Рис. 1.18. Диалоговое окно надстройки Подбор параметра

5. На панели Результат подбора параметра будет выведена информация о величине значения в ячейке F2 (рис. 1.19).

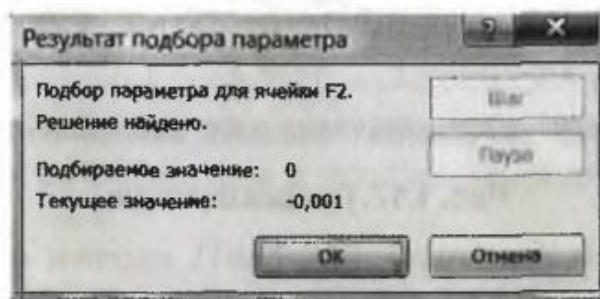


Рис. 1.19. Информационная панель надстройки Подбор параметра

3. В ячейке аргумента F1 появится подобранное значение 0,855. Таким образом, корень уравнения  $x \ll 0,855$  найден с заданной точностью



**Содержание отчета:**

- Отчет должен содержать скриншоты для разных  $R$  и  $N$  (6 вариантов).

**Контрольные вопросы.**

1. От чего зависит точность вычислений в электронных таблицах.
2. До какой точности целесообразно округлить полученное значение корня уравнения?

## Практическая работа №6

**Тема программы: Основные понятия теории систем.**

**Тема: Вероятностные модели. Построение информационной модели с использованием метода Монте-Карло.**

**Цель:** Получить практические навыки вероятностного оценивания систем.

### Построение информационной модели с использованием метода Монте-Карло

Вероятностные модели базируются на использовании больших серий испытаний со случайными параметрами, причем точность полученных результатов зависит от количества проведенных опытов. Построим вероятностную модель, позволяющую приближенно вычислять площади геометрических фигур. Эта модель будет основана на методе Монте-Карло.

Описательная модель вычисления площадей геометрических фигур с использованием метода Монте-Карло. Сначала построим описательную вероятностную модель данного метода: • поместим геометрическую фигуру полностью внутри квадрата; • будем случайным образом «бросать» точки в этот квадрат, т. е. с помощью генератора случайных чисел задавать координаты точек внутри квадрата; • будем считать, что отношение числа точек, попавших внутрь фигуры, к общему числу точек, попавших в квадрат, приблизительно равно отношению площади фигуры к площади квадрата, причем это отношение тем точнее, чем больше количество точек.

**Формальная модель «Определение площади круга методом Монте-Карло».** Построим формальную модель для вычисления площади круга радиуса  $r$ , центр которого совпадает с началом координат. Круг вписан в квадрат со стороной  $2 \cdot r$ , тогда площадь квадрата можно вычислить по формуле:  $S_1 = 4 \cdot r^2$ .

Пусть  $N$  — количество точек, которые случайным образом генерируются внутри квадрата.

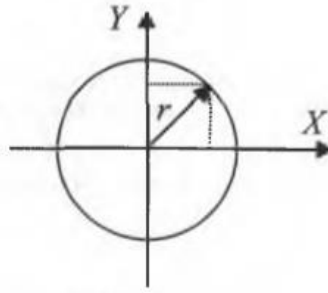


Рис. 6.1. Круг, вписанный в квадрат

Случайный выбор координат точек, которые попадают внутрь квадрата ( $N$  точек), должен производиться так, чтобы координаты точек  $x$  и  $y$  удовлетворяли условиям:

-  $-r < X < r$  и  $-r < y < r$ . Пусть  $M$  — количество точек, попавших внутрь круга, их координаты удовлетворяют условию:

$x^2 + y^2 < r^2$ . Предположим, что отношение площадей круга  $S_2$  и квадрата  $S_1$  равно отношению количества точек  $M$ , попавших внутрь круга, к количеству точек  $N$ , попавших внутрь квадрата, тогда получим формулу:

$S_2 / S_1 = M/N$  Тогда площадь круга можно вычислить в единицах измерения радиуса по формуле:  $S_2 = S_1 \cdot M/N = 4r^2 \cdot M/N$ . Более того, таким способом можно вычислить значение числа  $\pi$ . Подставим в формулу значение площади круга  $S_2 = \pi r^2$  и получим формулу для вычисления числа  $\pi$ :

$$\pi \cdot r^2 = 4 r^2 \cdot M/N; \pi = 4 \cdot M/N.$$

1.4.2. Компьютерные модели, построенные с использованием метода Монте-Карло, на языке Visual Basic

Разработаем на языке Visual Basic компьютерную модель, позволяющую методом Монте-Карло определять площадь круга и число  $\pi$ .

Проект «Метод Монте-Карло» на языке Visual Basic

1. Поместить на форму (рис. 6.2):

- графическое поле PictureBox1, в котором будет отображаться процесс случайной генерации точек;
- два текстовых поля: TextBox1 — для ввода количества генерируемых точек и TextBox2 — для ввода радиуса окружности;
- две надписи для вывода значения площади круга и числа  $\pi$ ;
- кнопку Button 1 для запуска обработчика события.

2. Установить размер графического поля: с помощью окна Свойства присвоить свойству Size значение 200;200.

3. Создать обработчик события, который обеспечивает:

- стирание результатов предыдущего опыта;
- ввод количества генерируемых точек и присваивание его переменной N;
- ввод значения радиуса окружности и присваивание его переменной R;
- генерацию случайных координат точек;
- рисование в графическом поле квадрата со стороной  $2 \cdot R$  и окружности радиуса R;
- подсчет в переменной M количества точек, попавших внутрь круга;
- вычисление и вывод значения площади круга и числа k на надписи.

```
Dim Graph1 As Graphics
```

```
Dim Pen1 As New Pen (Color.Black, 1)
```

```
Dim I, N, M, X, Y, R As Long
```

```
Private Sub Button1_Click (...) Graph1=Me.PictureBox1.CreateGraphics  
(
```

```
'Стирание Graph1.Clear (Color.White)
```

```
'Ввод данных
```

```
N=Val(TextBox1.Text)
```

```
R=Val(TextBox2.Text)
```

```
' Сдвиг по осям X и Y
```

```
Graph1.TranslateTransform (100, 100)
```

```
'Рисование круга и квадрата
```

```
Graph1.DrawEllipse (Pen2, -R, -R, 2*R, 2*R)
```

```
Graph1.DrawRectangle(Pen2, -R, -R, 2*R, 2*R)
```

```
For I=1 To N
```

```
'Генерация случайных координат точек и рисование точек в  
графическом поле
```

```
X=Int(Rnd() *2*R)-R Y=Int(Rnd() *2*R)-R
```

```
Graph1.DrawEllipse(Pen1, X, Y, 1, 1)
```

```
'Подсчет точек, попавших внутрь круга
```

```
If X2+Y2<=R2 Then M=M+1 Next I
```

```
'Вывод площади круга и числа
```

```
Pi Label1.Text=4*(M/N)* (R2)
```

```
Label2.Text=4*(M/N) End Sub
```

4. Ввести количество генерируемых точек и радиус окружности (от 1 до 100). Щелкнуть по кнопке Выполнить, в графическом поле

будет отображен процесс генерации случайных точек, а на надписи будут выведены значение площади круга и число  $\pi$  (см. рис. 1.21).



Рис. 6.2. Проект «Метод Монте-Карло» на языке Visual Basic

Компьютерный эксперимент. С помощью метода Монте-Карло можно определить с необходимой точностью значение числа  $\pi$  (при увеличении количества генерируемых точек будет увеличиваться точность вычисления числа  $\pi$ ).

**Содержание отчета:**

- Отчет должен содержать скриншоты для разных  $R$  и  $N$ .

**Контрольные вопросы.**

1. До какой точности целесообразно округлить полученное значение числа  $\pi$ ?

## Практическая работа №7

**Тема программы: Основные понятия теории систем.**

**Тема: Модели логических устройств компьютера.**

### Полусумматор

**Цель:** Получить практические навыки моделирования логических устройств ПК.

В электронных таблицах имеются логические функции, реализующие базовые логические операции. Аргументами и значениями логических функций являются логические значения ИСТИНА и ЛОЖЬ. Логическое значение может быть получено как результат определения значения логического выражения. Например, для логического выражения  $10 > 5$  результатом будет логическое значение ИСТИНА, а для логического выражения  $A1 < A2$  (где в ячейке A1 хранится число 10, а в ячейке A2 — число 5) — значение ЛОЖЬ.

Логическая функция И имеет в качестве аргументов логические значения, которые могут быть истинными или ложными, и задается формулой  $=\text{И}(\text{лог\_знач1}; \text{лог\_знач2}; \dots)$ . Принимает значение ИСТИНА тогда и только тогда, когда все аргументы имеют значение ИСТИНА. Например, значение функции  $=\text{И}(10 > 5; 10 < 5)$  - ЛОЖЬ.

Логическая функция ИЛИ имеет в качестве аргументов логические значения и задается формулой  $=\text{ИЛИ}(\text{лог\_знач1}; \text{лог\_знач2}; \dots)$ . Принимает значение ИСТИНА тогда и только тогда, когда хотя бы один из аргументов имеет значение ИСТИНА. Например, значение функции  $=\text{ИЛИ}(10 > 5; 10 < 5)$  — ИСТИНА.

Логическая функция НЕ меняет на противоположное значение своего аргумента и задается формулой  $=\text{НЕ}(\text{лог\_знач})$ . Принимает значение ИСТИНА, если аргумент имеет значение ЛОЖЬ, и наоборот. Например, значение функции  $=\text{НЕ}(10 > 5)$  — ЛОЖЬ.

Получим таблицу истинности операции логического умножения, значением которой в трех случаях является ЛОЖЬ и только в одном — ИСТИНА.

1. В пары ячеек (A2, B2), (A3, B3), (A4, B4)(A5, B5) ввести пары значений аргументов логической операции (ЛОЖЬ, ЛОЖЬ), (ИСТИНА, ЛОЖЬ), (ЛОЖЬ, ИСТИНА) и (ИСТИНА, ИСТИНА).
2. В ячейку C2 ввести формулу логической функции И:  $=\text{И}(A2;B2)$
3. Скопировать формулу в ячейки C3, C4 и C5.

Получим таблицу истинности операции логического сложения, значением которой в одном случае является ЛОЖЬ, в трех других — ИСТИНА.

4. В пары ячеек (E2, F2), (E3, F3), (E4, F4) (E5, F5) ввести пары значений аргументов логической операции (ЛОЖЬ, ЛОЖЬ), (ИСТИНА, ЛОЖЬ), (ЛОЖЬ, ИСТИНА) и (ИСТИНА, ИСТИНА). 5. В ячейку G2 ввести формулу логической функции ИЛИ: =ИЛИ(E2;F2). 6. Скопировать формулу в ячейки G3, G4 и G5.

Получим таблицу истинности операции логического отрицания, значением которой в одном случае является ИСТИНА, а в другом — ЛОЖЬ.

7. В ячейку I2 ввести значение аргумента логической операции ЛОЖЬ, а ячейку J2 — ИСТИНА. 8. В ячейку J2 ввести формулу логической функции НЕ: =НЕ(I2). 9. Скопировать формулу в ячейку J3. Получим таблицу, показанную на рис. 7.1.

	A	B	C	D	E	F	G	H	I	J
1	Логическое функция "И"			Логическая функция "ИЛИ"			Логическая функция "НЕ"			
2	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	ИСТИНА
3	ИСТИНА	ЛОЖЬ	ЛОЖЬ	ИСТИНА	ЛОЖЬ	ИСТИНА	ИСТИНА	ИСТИНА	ИСТИНА	ЛОЖЬ
4	ЛОЖЬ	ИСТИНА	ЛОЖЬ	ЛОЖЬ	ИСТИНА	ИСТИНА	ИСТИНА			
5	ИСТИНА	ИСТИНА	ИСТИНА	ИСТИНА	ИСТИНА	ИСТИНА	ИСТИНА			

Рис. 7.1. Таблицы истинности базовых логических операций  
Проект «Полусумматор» в электронных таблицах Microsoft Excel

1. На листе Полусумматор предусмотреть ввод логических значений аргументов в ячейки B1 и B2.

В ячейку B3 ввести формулу =И(B1;B2).

В ячейку B4 ввести формулу =НЕ(B3).

В ячейку B5 ввести формулу =ИЛИ(B1;B2).

В ячейку B6 ввести формулу =И(B5;B4).

2. Для создания графического интерфейса проекта запустить систему объектно-ориентированного программирования Visual Basic командой [Сервис-Макрос-Редактор Visual Basic].

3. В окне системы программирования добавить форму командой [Insert-UserForm].

4. Поместить на форму (рис. 7.3):

- кнопку CommandButton1 для запуска событийной процедуры;
- четыре надписи для отображения базовых логических элементов;
- два текстовых поля TextBoxA и TextBoxB для ввода логических значений на входе полусумматора;

- четыре надписи для вывода промежуточных логических значений, а также итоговых значений суммы и переноса.

5. Создать событийную процедуру:

```
Private Sub CommandButton1_Click()
    Cells(1, 2) = TextBoxA.Text
    Cells(2, 2) = TextBoxB.Text
    Label1.Caption = Cells(3, 2)
    Label3.Caption = Cells(4, 2)
    Label2.Caption = Cells(5, 2)
    Label4.Caption = Cells(6, 2)
End Sub
```

	А	В
1	A=	ЛОЖЬ
2	B=	ИСТИНА
3	P=	=И(B1;B2)
4	Not P =	=НЕ(B3)
5	Or =	=ИЛИ(B1;B2)
6	S=	=И(B5;B4)

Рис. 7.2. Формулы для модели полусумматора

6. Запустить проект, ввести значения аргументов и щелкнуть по кнопке Перенос и сумма.

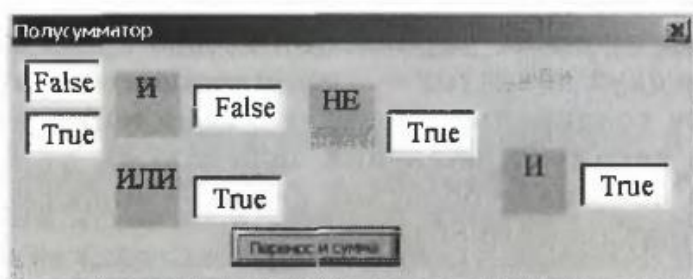


Рис. 7.3. Модель полусумматора в электронных таблицах



## Практическая работа №8.

**Тема программы: Основные понятия теории систем.**

**Тема: Построение и использование оптимизационной модели в электронных таблицах.**

**Цель:** Получить практические навыки исследования оптимизационных моделей в электронных таблицах.

### Построение и исследование оптимизационной модели в электронных таблицах

Надстройка Поиск решения. Возможности электронных таблиц Microsoft Excel не ограничиваются вычислениями по формулам и построением диаграмм и графиков. Задачи оптимизационного моделирования можно решать с помощью надстройки электронных таблиц Поиск решения, которая требует дополнительной установки. Процедура поиска решения позволяет найти оптимальное значение формулы, содержащейся в ячейке, которая называется целевой. Эта процедура работает с группой ячеек, прямо или косвенно связанных с формулой в целевой ячейке. Чтобы получить по формуле, содержащейся в целевой ячейке, заданный результат, процедура изменяет значения во влияющих ячейках. Чтобы сузить множество значений, используемых в модели, применяются ограничения. Эти ограничения могут ссылаться на другие влияющие ячейки.

Установим надстройку Поиск решения в Microsoft Excel 2003.

1. Ввести команду [Сервис- Надстройки]. В диалоговом окне Надстройки в списке Доступные надстройки: выбрать настройку Поиск решения путем установки флажка (рис. 8.1.). Щелкнуть по кнопке ОК.

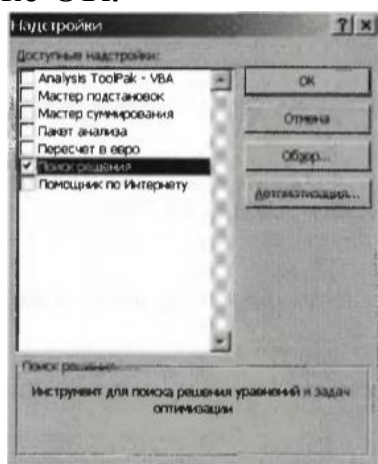


Рис. 8.1. Диалоговое окно Надстройки в Microsoft Excel 2003

Установим надстройку Поиск решения в Microsoft Excel 007.

1. Щелкнуть по значку Кнопка Microsoft Office & . В появившемся диалоговом окне щелкнуть по кнопке Параметры Excel. В появившемся диалоговом окне выбрать пункт Надстройки. В списке Управление выбрать пункт Надстройки Excel и нажать кнопку Перейти. В диалоговом окне Надстройки в списке Доступные надстройки: выбрать настройку Поиск решения путем установки флажка (рис. 1.31). Щелкнуть по кнопке ОК.

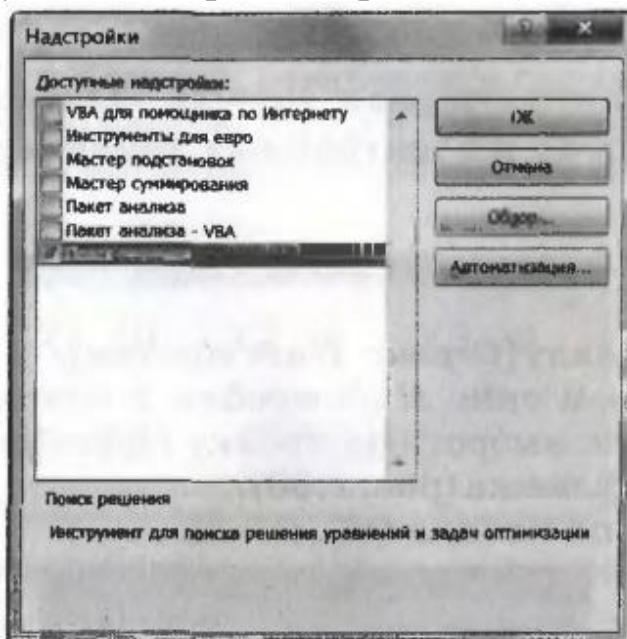


Рис. 8.2. Диалоговое окно Надстройки в Microsoft Excel 2007

2. Ячейки B2, C2 и D2 выделить для хранения значений параметров X1, X2 и X3 (рис. 1.32). В ячейку B4 ввести формулу вычисления целевой функции:  $=B2+C2+D2$ . В ячейку B7 ввести формулу вычисления количества заготовок типа А:  $=10*B2+3*C2 + 8*D2$ . В ячейку B8 ввести формулу вычисления количества заготовок типа Б:  $=3*B2+6*C2 + 4*D2$ .

А В С 1 X1 X2 x 3 2 Параметры: 0 0 0 3 4 Целевая функция: 0 ..5 ~ г  
Ограничения X Кол-во заготовок А: 0 8 Кол-во заготовок Б: 0

	А	В	С	Д
1		X1	X2	X3
2	Параметры:	0	0	0
3				
4	Целевая функция:	0		
5				
6	<b>Ограничения</b>			
7	Кол-во заготовок А:	0		
8	Кол-во заготовок Б:	0		

### Рис. 8.3. Ввод формул оптимизационной модели

Для поиска оптимального набора значений параметров, который соответствует минимальному значению целевой функции, воспользуемся надстройкой электронных таблиц Поиск решения.

### Построение и исследование информационных моделей 77

2. В электронных таблицах Microsoft Excel 2003 ввести команду {Сервис-Поиск решения...}. В электронных таблицах Microsoft Excel 2007 ввести команду [Данные-Поиск решения...]. В появившемся диалоговом окне Поиск решения (рис. 1.33) установить:

- адрес целевой ячейки;
- вариант оптимизации значения целевой ячейки (в нашем случае минимизацию);
- адреса ячеек, значения которых изменяются в процессе поиска решения (в которых хранятся значения параметров);
- ограничения (типа равно для ячеек, хранящих количество заготовок, и типа больше равно для параметров).

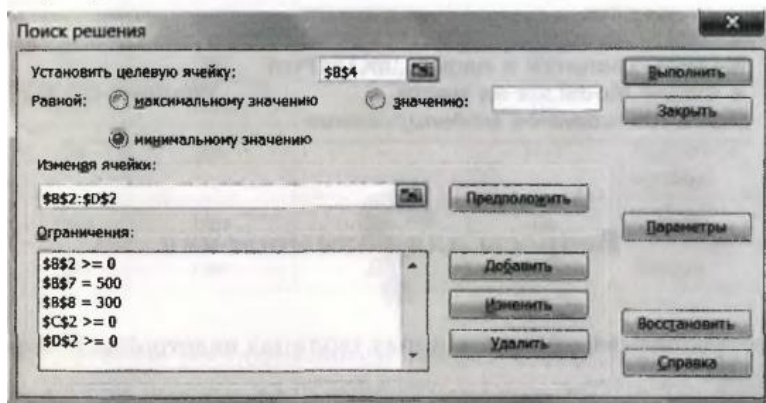


Рис. 8.4. Диалоговое окно Поиск решения

4. Щелкнуть по кнопке Выполнить. В диалоговом окне Текущее состояние поиска решения (рис. 8. 4) нажимать кнопку Продолжить до тех пор, пока в ячейке целевой функции не появится минимальное значение (рис. 8.5).

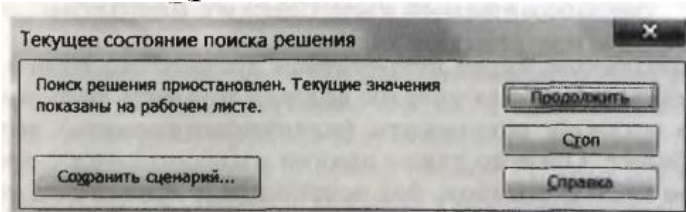


Рис. 8.5. Процесс поиска решения

5. Таким образом, для изготовления 500 деталей А и 300 деталей Б требуется 70 листов материала, при этом 20 листов необходимо

раскроить по первому, 20 листов — по второму и 30 листов — по третьему варианту (см. рис. 8.6).

	A	B	C	D
1		X1	X2	X3
2	<b>Параметры:</b>	20	20	30
3				
4	<b>Целевая функция:</b>	70		
5				
6	<b>Ограничения</b>			
7	<b>Кол-во заготовок А:</b>	500		
8	<b>Кол-во заготовок Б:</b>	300		

Рис. 8.6. Результат оптимизации

## Практическая работа №9.

**Тема программы: Основные понятия теории систем.**

**Тема: Графы и их исследование. Описание графа с помощью матрицы смежности.**

**Цель:** Описать граф с помощью матрицы смежности.

### Введение в теорию графов

При проектировании компьютерных сетей, телефонных линий, трубопроводов и строительстве дорог необходимо минимизировать затраты на прокладку коммуникаций. Прежде всего, целесообразно выбрать минимальный по длине маршрут прокладки коммуникаций. Например, необходимо соединить телефонным и/или компьютерным кабелем шесть зданий, расстояния между которыми различны (рис. 9.1.).

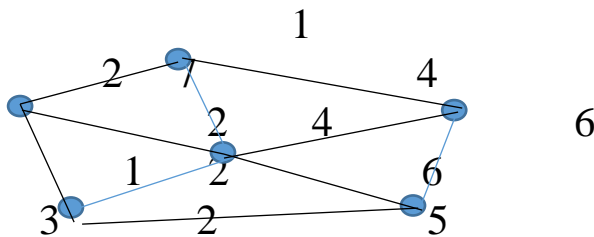


Рис. 9.1. Задача прокладки коммуникаций

Возникает задача определения маршрута прокладки кабеля минимальной длины, но при этом подходящего к каждому зданию. Для решения таких задач часто используют теорию графов.

**Основные понятия теории графов.** Граф  $G$  задается с помощью пары множеств  $G = (V, R)$ , где  $V$  — множество (совокупность) вершин,  $R$  — множество ребер, соединяющих пары вершин. Множество ребер может быть пустым, если ни одна из вершин не соединена с другими вершинами. Обычно граф представляют с помощью схемы, на которой некоторые вершины соединены линиями (ребрами) (рис. 9.2).

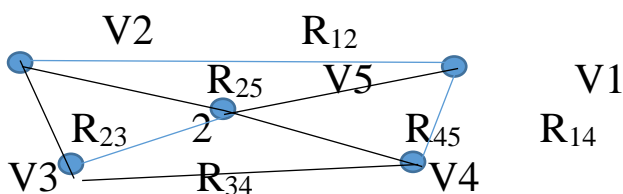


Рис. 9.2. Граф  $G$  в форме схемы

Вершинами могут служить объекты любой природы: будь то населенные пункты, компьютеры сети, элементы блок-схем алгоритмов и т. д. Под ребрами могут подразумеваться дороги между двумя соседними городами, стороны геометрических фигур, линии связи между компьютерами. Любую систему улиц в городе можно представить в виде графа. Здесь вершины — это перекрестки. Вершины называются смежными, если их соединяет ребро. Например, на рис. 9.2 смежны вершины  $V_1$  и  $V_2$ , так как их соединяет ребро  $R_{12}$ . Множества  $V$  и  $R$  являются конечными — мы можем перечислить все вершины и ребра графа. Количество вершин и количество ребер графа определяют мощности множеств  $V$  и  $R$ . Так, количество вершин графа  $G$  равно 5, а количество ребер равно 8. Ребро и любая из его двух вершин называются инцидентными. Под степенью вершины подразумевается количество инцидентных ей ребер. Так, степень вершины  $V_1$  равна 3, а степень вершины  $V_5$  равна 4.

**Маршрут графа** — это последовательность чередующихся вершин и ребер. В графах можно выделить различные маршруты. Маршрут является замкнутым (циклом), если его начальная и конечная вершины совпадают. Например, в графе  $G$  можно выделить несколько циклических маршрутов, например  $V_1R_{12}V_2R_{23}V_3R_{34}V_4R_{14}V_1$  и  $V_2R_{23}V_3R_{35}V_5R_{25}V_2$ . Маршрут называется простой цепью, если все его вершины и ребра различны. Длина маршрута равна количеству ребер, входящих в него. Одна вершина достижима из другой, если между ними проложен маршрут. Граф считается связным, если каждая его вершина достижима из любой другой. Например, граф  $G$  является связным, так как между любыми двумя его вершинами можно проложить маршрут. Например, маршрут между вершинами  $V_1$  и  $V_4$  может быть следующим:  $V_1R_{12}V_2R_{23}V_3R_{34}V_4$ . Вершины, которые не имеют инцидентных ребер, называются изолированными вершинами. Можно также сказать, что степень таких вершин нулевая. Из всего этого следует, что изолированные вершины недостижимы из любых других вершин.

**Ориентированные графы.** Сообразительный читатель может задать вопрос: имеет ли значение направление, ориентация ребра? Ведь в системе улиц можно найти дороги как с односторонним, так и с двусторонним движением. Именно поэтому в теории графов

вводится понятие орграфа — ориентированного графа. В орграфе каждое ребро имеет одно направление. Такие ребра называются дугами. Другими словами, ребро — это неупорядоченная пара вершин, а дуга — упорядоченная. Очевидным является тот факт, что дуги  $R_{12}$  и  $R_{21}$  не совпадают в орграфе. Для орграфа вводятся такие понятия, как входящая и исходящая степени вершины. Это соответственно число входящих в вершину дуг и число исходящих из нее дуг.

**Взвешенные графы.** Очень часто в практических применениях желательно приписывать ребрам графа веса для того, чтобы моделировать такие величины, как расстояние, время перехода или стоимость доставки между двумя точками. Взвешенный граф (сеть) — это такой граф, ребрам или дугам которого поставлены в соответствие числовые величины. Вес сети равен сумме весов ее ребер.

**Описание графа с помощью матрицы смежности.** Для наглядного представления графа используются схемы (см. рис. 9.2), а для математических расчетов удобнее использовать представление графа в форме матрицы смежности (рис. 9.3, а). Матрицу смежности можно представить в виде таблицы, строки и столбцы которой соответствуют номерам вершин графа. Если вершины смежны, то элемент матрицы смежности равен 1, если вершины не смежны, то элемент матрицы равен 0. Диагональные элементы матрицы равны 0, так как вершины сами с собой не смежны (их не соединяет ребро). Пусть в матрице смежности графа  $G$  строки нумеруются индексом  $p$ , а столбцы — индексом  $k$ , тогда элементы матрицы смежности обозначаются  $R_{nk}$ . Для смежных вершин элементы матрицы смежности ( $R_{12}, R_{14}, R_{15}, R_{21}, R_{23}, R_{25}, R_{32}, R_{34}, R_{35}, R_{41}, R_{43}, R_{45}, R_{51}, R_{52}, R_{53}, R_{54}$ ) равны 1, а для не смежных вершин элементы матрицы ( $R_{13}, R_{24}, R_{42}$ ) равны 0. Диагональные элементы матрицы ( $R_{11}, R_{22}, R_{33}, R_{44}, R_{55}$ ) равны 0.

Преобразуем граф  $G$  в сеть, т. е. присвоим ребрам, соединяющим смежные вершины, числовые значения ( $R_{12} = 50, R_{14} = 25, R_{15} = 10, R_{21} = 50, R_{23} = 25, R_{25} = 30, R_{32} = 25, R_{34} = 50, R_{35} = 35, R_{41} = 25, R_{43} = 50, R_{45} = 15, R_{51} = 10, R_{52} = 30, R_{53} = 35, R_{54} = 15$ ). Отообразим сеть  $G$  в форме матрицы смежности (рис. 9.3, б).

	1	2	3	4	5			1	2	3	4	5
1	0	1	0	1	1		1	0	50	0	25	10
2	1	0	1	0	1		2	50	0	25	0	30
3	0	1	0	1	1		3	0	25	0	50	35
4	1	0	1	0	1		4	25	0	50	0	15
5	1	1	1	1	0		5	10	30	35	15	0

а

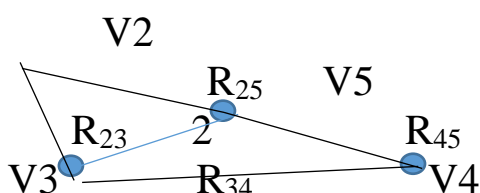
б

Рис. 9.3. Граф и сеть G в форме матрицы смежности

## 5. Составление отчета.

В отчет должны войти:

1. Постройте матрицу смежности для графа.
2. Опишите маршрут графа.



## Контрольные вопросы

1. Какие графы являются деревьями?
2. В какой форме можно представить граф?
3. Раскройте содержание прикладного уровня информационных технологий.
4. Охарактеризуйте понятия «системы и управления» с позиций системного подхода.
5. Что такое граф? Для чего применяются графы?



## Практическая работа № 10.

Тема программы: Основные понятия теории систем.

Тема: Построение остовного связного графа дерева. Вес графа.

**Цель:** Построить остовной связный граф дерева. Определить вес графа.

**Подграфы и деревья.** Подграфом графа  $G$  называется граф, у которого все вершины и ребра принадлежат графу  $G$  (рис. 1.60, а).

Остовной связный подграф — это подграф графа  $G$ , который содержит все его вершины и каждая его вершина достижима из любой другой (рис. 1.60, б).

**Дерево** — это граф, в котором нет циклов, т. е. граф, в котором нельзя из некоторой вершины пройти по нескольким различным ребрам и вернуться в ту же вершину. Остовным связным деревом называется подграф, включающий все вершины исходного графа  $G$ , каждая вершина которого достижима из любой другой, и при этом не содержащий циклов (рис. 1.60, в).

Рис. 1.60. а — подграф графа  $G$ , б — остовной связный подграф графа  $G$ , в — остовное связное дерево

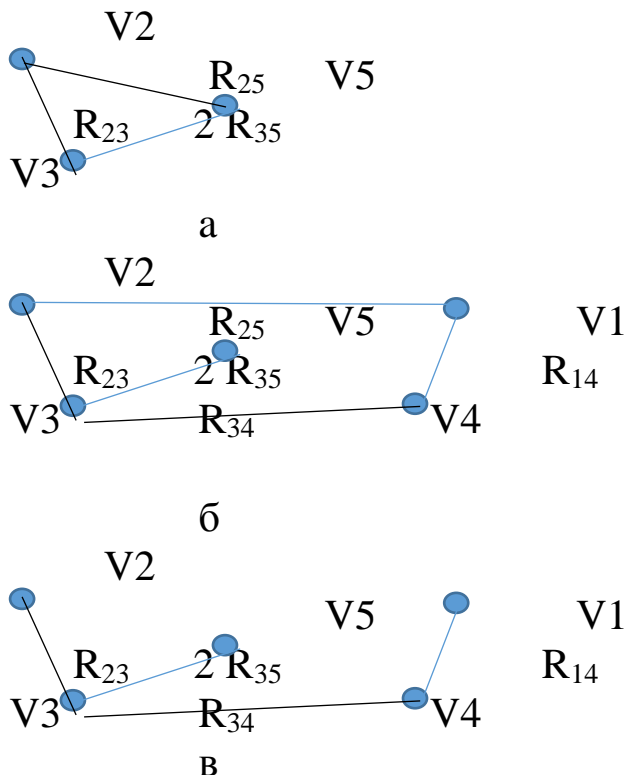


Рис. 10.1. а-подграф графа  $G$ , б-остовной связный подграф графа  $G$ , в – остовное связное дерево.

**Преобразование графа в остовное связное дерево минимального веса.** Пусть  $G = (V, R)$  — связный взвешенный неориентированный граф, где  $V$  — множество вершин, а  $R$  — множество ребер (см. рис. 9.2). Ребра графа не ориентированы, т. е. ребра  $R_{nk}$  и  $R_{kn}$  считаются одним и тем же ребром, поэтому в матрице смежности не учитываются дублирующие друг друга ребра. В результате граф  $G$  можно представить с помощью матрицы смежности, содержащей значения весов десяти ребер (на рис. 10.2 выделены жирным шрифтом).

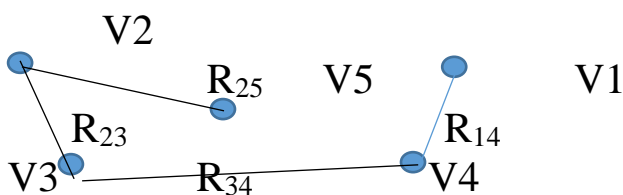
	1	2	3	4	5
1	0	50	0	25	10
2	50	0	25	0	30
3	0	25	0	50	25
4	25	0	50	0	15
5	10	30	35	15	0

Рис.10.2. Матрица смежности связного взвешенного неориентированного графа  $G$

Введем понятие цикломатического числа  $u$ , показывающего, сколько ребер графа нужно удалить, чтобы в нем не осталось ни одного цикла. Цикломатическое число  $u$  равно

увеличенной на единицу разности между количеством ребер и количеством вершин графа:  $u = m - n + 1$ , где  $m$  — количество ребер,  $n$  — количество вершин. Например, для графа, изображенного на рис. 9.2, цикломатическое число равно:  $u = m - n + 1 = 8 - 5 + 1 = 4$ . Это значит, что для получения остовного связного дерева в графе  $G$  необходимо убрать четыре ребра, и тогда в нем не останется ни одного цикла.

Для каждого графа обычно существует несколько остовных связных деревьев, которые обладают различными весами. На рис. 10.1, в представлено остовное связное дерево графа  $G$ , вес которого равен 135, а на рис. 10.3 представлены три остовных связных дерева графа  $G$ , веса которых равны 130, 100, 135.



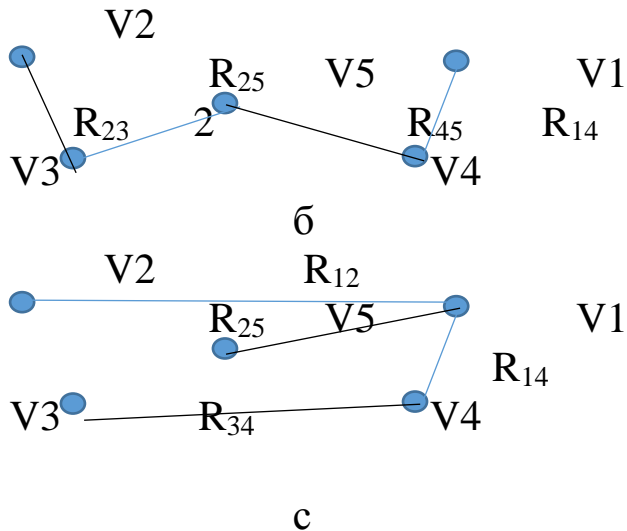


Рис. 10.3. Остовные связные деревья графа  $G$

Для построения остовного связного дерева минимального веса используется алгоритм Крускала.

1. Первоначально из графа удаляются все ребра, получается остовной подграф, где все вершины изолированы. Каждая вершина такого графа помещается в одноэлементное подмножество.

2. Ребра сортируются по возрастанию весов.

3. Ребра последовательно, по возрастанию их весов, включаются в остовное дерево. Существуют четыре случая: а) обе вершины включаемого ребра принадлежат одноэлементным подмножествам, тогда они объединяются в новое, связное подмножество; б) одна из вершин принадлежит связному подмножеству, а другая нет, тогда включаем вторую в подмножество, которому принадлежит первая; в) обе вершины принадлежат разным связным подмножествам, тогда объединяем подмножества; г) обе вершины принадлежат одному связному подмножеству, тогда исключаем данное ребро.

4. Алгоритм заканчивает работу, когда все вершины будут объединены в одно множество, при этом оставшиеся ребра не включаются в остовное дерево.

Задание. Постройте остовное дерево минимального веса для графа  $G$ , применяя данный алгоритм.

б. Составление отчета.

В отчет должны войти:

- Последовательность построения остовного дерева минимального веса для графа  $G$  Данные построения внести в таблицу ( поэтапно, как по алгоритму).

### Контрольные вопросы

1. Укажите основные категории системного подхода.
2. Укажите основные этапы логического анализа информационных систем.
3. Что называю остовным графом?
4. Что такое сложная система? Каковы отличия от простых систем?
5. Что такое и для чего нужна модель «черный ящик»?
6. Что такое модель структуры системы?
7. Как можно представить процесс функционирования любой системы?

### Библиографический список

1. Информатика. Базовый курс / под ред. С.В. Симоновича. 3-е изд. СПб.: Питер, 2012. 640 с.
2. Вдовин, В.М., Суркова, Валентинов, ВТА- Теория систем и системный анализ [Электронный ресурс]: Учебник для бакалавров, Издательство; ДАШКОВ и К, 2014 г, 644 с,.
3. Лисицин, Л.А. Теоретические основы и методы исследования информационных процессов и систем :[ Текст] : учебное пособие /Халин Ю.А., Катыхин, Ю.А. Курск:ЮЗГУ, 2017.-126 с