ФГБОУ ВПО «Воронежский государственный
технический университет»

Кафедра систем информационной безопасности

289-2015

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к курсовой работе по дисциплине
«Информатика»
для студентов специальностей
090301 «Компьютерная безопасность»,
090302 «Информационная безопасность
телекоммуникационных систем»,
090303 «Информационная безопасность
автоматизированных систем»
очной формы обучения

Воронеж 2015

Составитель д-р техн. наук О. Н. Чопоров

УДК 004

Методические указания к курсовой работе по дисциплине «Информатика» для студентов специальностей 090301 «Компьютерная безопасность», 090302 «Информационная безопасность телекоммуникационных систем», 090303 «Информационная безопасность автоматизированных систем» очной формы обучения / ФГБОУ ВПО «Воронежский государственный технический университет»; сост. О. Н. Чопоров. Воронеж, 2015. 49 с.

В данных методических указаниях приведены краткие теоретические сведения и задание для выполнения курсовой работы по дисциплине «Информатика». Методические указания предполагают углубленное изучение лекционного материала и приобретение навыков по разработке различных алгоритмических конструкций и их описанию с использованием псевдокода, блок-схем и языков программирования.

Методические указания подготовлены в электронном виде в текстовом редакторе MS Word 2007 и содержатся в файле Чопоров_КР_Информатика.pdf.

Табл. 7. Ил. 9. Библиогр.: 9 назв.

Рецензент д-р техн. наук, проф. А. Г. Остапенко

Ответственный за выпуск зав. кафедрой д-р техн. наук, проф. А. Г. Остапенко

Издается по решению редакционно-издательского совета Воронежского государственного технического университета

© ФГБОУ ВПО «Воронежский государственный технический университет», 2015

ВВЕДЕНИЕ

Алгоритмизация – это раздел информатики, изучающий методы и приемы построений алгоритма, а также их свойства. Она является основным, базовым компонентом компьютерной грамотности в современном компьютерном мире. Для достижения положительных результатов важную роль играет умение разрабатывать оптимальный алгоритм решения поставленной задачи, что требует от исполнителя наличия определенных навыков алгоритмизации и системного анализа, а также знания математики, физики, химии, экономики и других дисциплин.

Основу деятельности специалиста практически любой области составляет умение ставить задачи, разрабатывать алгоритмы, получать решения, производить анализ полученных данных и делать выводы. Поэтому в своей будущей профессиональной деятельности студенты должны уметь грамотно применять персональный компьютер (ПК) для решения научных и производственных задач.

Курсовая работа предполагает самостоятельную разработку алгоритма решения поставленной задачи (в соответствии с выбранным вариантом задания) и его описание в виде блок-схемы, а также псевдокода или программы на одном из языков программирования.

Данные методические указания по написанию курсовой работы содержат теоретические сведения, включая основные структуры алгоритмов и типовые приемы алгоритмизации, задание, требования к работе, а также сроки ее выполнения.

1. ЦЕЛИ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ

Целью курсовой работы является разработка алгоритма решения задачи (в соответствии с выбранным вариантом), описание его с помощью блок-схемы, а также с использованием псевдокода или в виде программы на одном из языков программирования.

При выполнении курсовой работы студенты должны изучить основные этапы создания компьютерной программы, понятие и свойства алгоритма, способы описания алгоритмов, основные алгоритмические конструкции, типовые приемы алгоритмизации.

Практическая часть курсовой работы ориентирована на разработку алгоритма решения задачи, его описание в виде блок-схемы и псевдокода или программы на языке программирования высокого уровня.

2. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И ОБЪЁМУ КУРСОВОЙ РАБОТЫ

Основные требования к курсовой работе (КР) установлены стандартом предприятия СТП ВГТУ 62-2007. КР состоит из расчетно-пояснительной записки (РПЗ) объёмом от 30 до 50 страниц печатного текста с иллюстративным графическим материалом, размещенным по разделам работы.

Пояснительная записка содержит следующие разделы:

а) титульный лист;

б) задание на курсовую работу;

в) лист «Замечания руководителя»;

г) содержание включает введение, наименование всех разделов, подразделов, пунктов (если они имеют наименование), заключение, список литературы, наименование приложений с указанием номеров страниц, с которых начинаются эти элементы пояснительной записки;

д) введение;

е) описание задания;

ж) теоретическую часть (подробное описание поставленной задачи, способы ее решения, теоретические вопросы алгоритмизации);

з) практическую часть (исследовательскую) содержащую:

- блок-схему алгоритма;
- описание алгоритма на псевдокоде (листинг программы);
- тестовые примеры;
- при разработке программ – описание программы (структура, интерфейс).

и) заключение;

к) список литературы;

л) приложения (при необходимости).

Также к КР прилагается диск с электронным вариантом курсовой работы.

2.1. График выполнения курсовой работы

Таблица 1

График выполнения курсовой работы

Срок выполнения	Содержание работы
1 – 2-я недели семестра	Выбор задания курсовой работы. Ознакомление с постановкой задачи
3 – 8-я недели семестра	Осмысление задания, изучение подхода к его выполнению, написание теоретической части работы.
9 – 12-я недели семестра	Разработка алгоритма решения задачи и его описание в виде блок-схемы и с использованием псевдокода (или языка программирования)
13 – 16-я недели семестра	Описание примеров решения поставленной задачи. Оформление пояснительной записки.
17 – 18-я недели семестра	Сдача пояснительной записки. Защита курсовой работы

2.2. Последовательность выполнения

Последовательность выполнения, рекомендации по выполнению разделов работы:

1. Получить вариант задания на курсовую работу.

2. Изучить постановку задачи и представленный в ней метод (алгоритм вычисления, игру и др.). Описать поставленную задачу (метод шифрации, вычислительный алгоритм, игру и др., в зависимости от варианта задания).

3. Привести общую формулировку задачи (в содержательных терминах с определением входных и выходных данных).

4. Привести математическую формулировку задачи (определить математические величины, которые будут описывать задачу, а также математические связи между ними, т.е. составить математическая модель)

5. Выбрать метод решения (исходя как из субъективных причин (знание тех или иных математических методов), так и объективных (имеющиеся ресурсы), из большого количества математических методов выбирается тот, который целесообразно использовать для решения поставленной задачи).

6. Составить алгоритм решения задачи и дать его описание в виде блок-схемы.

7. Описать алгоритм решения задачи в виде псевдокода (или составить и отладить программу на языке программирования высокого уровня).

8. Привести подробное описание контрольных примеров решения задачи с использованием разработанного алгоритма.

9. В случае разработки программы – привести ее структуру, описание интерфейса и тестовый пример выполнения.

10. Оформить отчет по курсовой работе.

2.3. Критерии оценки курсовой работы

Оценка за курсовую работу складывается из оценки за предоставленный отчет, полноту выполненной работы, защиту (ответы на вопросы по теме проекта) и составляет от 2 до 5 («неудовлетворительно», «удовлетворительно», «хорошо», «отлично»).

3. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

3.1. Понятие алгоритма

Понятие алгоритма относится к числу основных понятий современной вычислительной математики и информатики.

Алгоритм — описанная на некотором языке точная конечная система правил, определяющая содержание и порядок действий над некоторыми объектами, строгое выполнение которых дает решение поставленной задачи.

Разработать алгоритм решения означает разбить задачу на последовательно выполняемые этапы. Можно сказать, что алгоритм описывает процесс преобразования исходных данных в результаты, т. к. для решения любой задачи необходимо:

1) ввести исходные данные;

2) преобразовать исходные данные в результаты (выходные данные);

3) вывести результаты.

На этапе разработки алгоритма рекомендуется придерживаться следующих правил его составления:

1. Алгоритм должен быть максимально прост и понятен.

2. Алгоритм должен состоять из мелких шагов.

3. Сложная задача должна разбиваться на достаточно простые, легко воспринимаемые части (блоки).

4. Логика алгоритма должна опираться на минимальное число достаточно простых базовых управляющих структур.

В итоге процесс разработки алгоритма должен быть направлен на получение четкой структуры алгоритмических конструкций.

3.2. Способы описания алгоритмов

На любой стадии существования алгоритмы представляют с помощью конкретных изобразительных средств, состав и правила употребления которых образуют

конкретные способы или формы записи. К настоящему времени сложились пять наиболее употребительных способов записи:

словесное описание;

формульно-словесное описание;

псевдокод;

графический способ (блок-схема);

программа (способ описания с помощью языков программирования).

Словесное описание

Словесное описание представляет алгоритм – инструкцию о выполнении действий в определенной последовательности с помощью слов и предложений естественного языка. Форма изложения произвольна и устанавливается разработчиком.

Этот способ описания не имеет широкого распространения, т.к. строго не формализуем, допускает неоднозначность толкования при описании некоторых действий, страдает многословностью.

Пример 1

Алгоритм нахождения наибольшего общего делителя (НОД) двух натуральных чисел:

1) задать два числа;

2) если числа равны, то взять любое из них в качестве ответа и остановиться, в противном случае продолжить выполнение алгоритма;

3) определить большее из чисел;

4) заменить большее из чисел разностью большего и меньшего из чисел;

5) повторить алгоритм с п. 2.

Формульно-словесный способ

Формульно-словесный способ записи действий содержит формальные символы и выражения (формулы) в сочетании со словесными пояснениями, т.е. алгоритм

записывается в виде текста с формулами по пунктам, определяющим последовательность действий. Этот способ описания нагляден, лаконичен, но не является строго формальным.

П р и м е р 2

Алгоритм вычисления следующего выражения: y = 2a - (x + 6).

1) ввести значения ***a*** и ***x***;
2) найти сумму (***x*** + 6);
3) найти произведение (2**a*);
4) вычислить ***y*** как разность ***y*** = *2**a*** — (***x*** + 6);
5) вывести ***y*** как результат вычисления выражения.

Псевдокод

Псевдокод представляет собой описание структуры алгоритма на естественном, частично-формализованном языке, позволяющее выявить основные этапы решения задачи перед точной его записью на языке программирования.

В псевдокоде используются некоторые формальные конструкции и общепринятая математическая символика. Данный способ тесно связан со структурным подходом к программированию. Псевдокод занимает промежуточное положение между естественным языком и языком программирования. Его применяют преимущественно для того, чтобы подробнее объяснить работу программы, что облегчает проверку правильности программы.

П р и м е р 3

Алгоритм сложения двух чисел.

Псевдокод:

1) Ввод двух чисел a и b.
2) Вычисление суммы S = a + b.
3) Вывод S.
4) Конец.

П р и м е р 4

Алгоритм заполнения зачетной ведомости группы из 20 студентов (i - номер студента).

Псевдокод начало цикла

для i от 1 до 20 с шагом 1 **повторять:**

1.1. ввести фамилию студента

1.2. поставить оценку.

конец цикла (кц)

вывод ведомости

Графическая запись

Графическая запись, или **блок-схема**, - описание структуры алгоритма с помощью геометрических фигур с линиями связями, показывающими порядок выполнения отдельных инструкций. Описание алгоритмов с помощью схем - один из наиболее наглядных и компактных способов. Этот способ имеет ряд преимуществ перед остальными:

- наглядное отображение базовых конструкций алгоритма;
- концентрация внимания на структуре алгоритма;
- использование принципа блочности при коллективном решении сложной задачи;
- преобразование алгоритма методом укрупнения (сведения к единому блоку) или детализации (разбиения на ряд блоков);
- быстрая проверка разработанного алгоритма.

В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т. п.) соответствует геометрическая фигура, называемая блоком. Блочные символы соединяются линиями переходов (стрелками), определяющими очередность выполнения действий.

Условные графические изображения, используемые при построении схем, называются символами. Система символов и правила построения алгоритмов определены

соответствующими стандартами: блок-схема выстраивается в одном направлении: либо сверху вниз, либо слева направо, в порядке выполнения действий.

Для наглядности операции разного вида изображаются в схеме различными геометрическими фигурами (табл. 2).

Таблица 2

Средства графического изображения алгоритмов

Наименование	Символ	Выполняемые действия
Терминатор (пуск-остановка)		Обозначает начало и конец алгоритма
Данные (ввод-вывод)		Ввод и вывод данных
Процесс		Вычислительные действия
Решение		Наличие условия в алгоритме
Граница цикла		Наличие цикла в алгоритме
Предопределенный процесс		Наличие подпрограммы
Соединитель		Разрыв алгоритма (соединительные блоки)
Комментарий		Внесение комментариев

Терминатор (пуск-остановка). Элемент отображает вход из внешней среды или выход из нее (наиболее частое применение - начало и конец алгоритма). Внутри фигуры записывается соответствующее действие – начало/конец.

Данные (ввод-вывод). Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод). Данный символ не определяет типа носителя данных.

Процесс. Выполнение одной или нескольких операций, обработка данных любого вида (изменение значения данных, формы представления, расположения). Внутри фигуры записывают непосредственно сами операции. В этом блоке знак «=» означает не математическое равенство, а операцию присваивания.

Решение. Отображает решение или функцию переключательного типа с одним входом и двумя или более альтернативными выходами, из которых только один может быть выбран после вычисления условий, определенных внутри этого элемента. Вход в элемент обозначается линией, входящей обычно в верхнюю вершину элемента. Если выходов два или три, то обычно каждый выход обозначается линией, выходящей из оставшихся вершин (боковых и нижней). Если выходов больше трех, то их следует показывать одной линией, выходящей из вершины (чаще нижней) элемента, которая затем разветвляется. Соответствующие результаты вычислений могут записываться рядом с линиями, отображающими эти пути.

Граница цикла. Условия цикла и приращения записываются внутри символа цикла - в зависимости от типа организации цикла. Для изображения итерационного цикла на блок-схеме вместо данного символа используют символ решения, указывая в нем условие, а одну из линий выхода замыкают выше в блок-схеме (перед операциями цикла).

Предопределенный процесс. Символ отображает выполнение процесса, состоящего из одной или нескольких операций, который определен в другом месте программы (в подпрограмме, модуле). Внутри символа записывается название процесса и передаваемые в него данные.

Соединитель. Используется для обрыва линии и продолжения ее в другом месте (пример: разделение блок-схемы, не помещающейся на листе).

Соответствующие соединительные символы должны иметь одно (при том уникальное) обозначение.

Комментарий. Позволяет включать в схемы алгоритмов пояснения к функциональным блокам. Частое использование комментариев нежелательно, т. к. это усложняет (загромождает) схему, делает ее менее наглядной.

Составление блок-схемы алгоритма является важным и необходимым этапом решения задачи. Но, т.к. в таком виде алгоритм не может быть выполнен непосредственно ЭВМ, этот этап является промежуточным, значительно облегчающим процесс составления программы.

П р и м е р 4

Алгоритм сложения двух чисел (рис. 1).

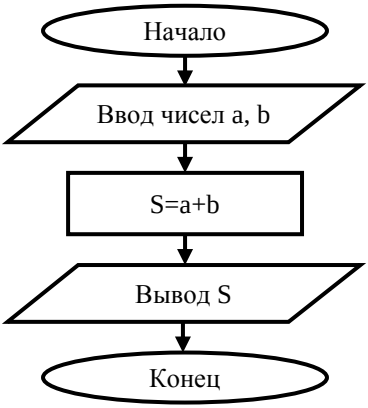


Рис. 1. Алгоритм сложения двух чисел

Программа

Программа – это алгоритм, записанный в виде последовательности команд, понятных ЭВМ (машинных команд). При записи алгоритмов в виде программ для ЭВМ используются языки программирования – системы

кодирования предписаний и правила их использования. Такие языки являются искусственными языками со строго определенными синтаксисом и пунктуацией. Они не допускают свободного толкования для своих конструкций, как это характерно для естественного языка. Существует большое количество языков программирования, предназначенных для решения прикладных задач.

Для записи алгоритмов в виде программ характерна высокая степень формализации. Перед составлением программ чаще всего используются словесно-формульный или графический способы.

П р и м е р 5

Алгоритм сложения двух чисел. Программа, записанная на языке Qbasic:

```
Rem Вычисление суммы двух чисел
Input «Введите число a»; a
Input «Введите число b»; b
Print «Сумма a+b=»; a+b
End
```

3.3. Основные алгоритмические конструкции (виды алгоритмов)

Алгоритмы можно представлять как некоторые структуры, состоящие из отдельных базовых (т.е. основных) элементов. Эти элементарные шаги объединяются в алгоритмические конструкции.

В зависимости от особенностей своего построения алгоритмы можно разделить на следующие группы:

1) линейные (последовательные);
2) разветвляющиеся;
3) циклические;
4) рекурсивные.

Линейным называется алгоритм, в котором все этапы решения задачи выполняются ровно один раз и строго последовательно.

П р и м е р 6

Составить алгоритм вычисления площади и периметра треугольника, если известны длины трех его сторон (рис. 2).

Входные данные: a, b, c (длины сторон треугольника);

Выходные данные: S, P (площадь и периметр треугольника).

Псевдокод:

1. Ввод чисел a, b, c
2. Вычисление полупериметра

$$r = \frac{a+b+c}{2}$$

3. Вычисление периметра

$$P = a + b + c$$

4. Вычисление площади

$$\mathbf{S} = \sqrt{\boldsymbol{r}(\boldsymbol{r}-\boldsymbol{a})(\boldsymbol{r}-\boldsymbol{b})(\boldsymbol{r}-\boldsymbol{c})}$$

5. Вывод S, P
6. Конец

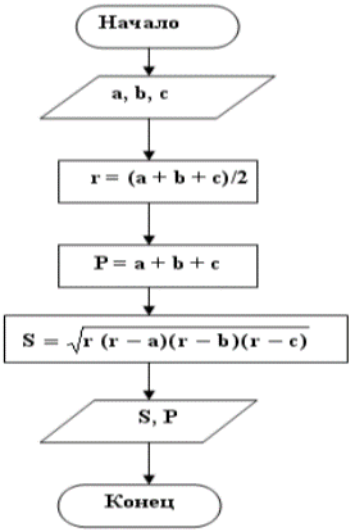


Рис. 2. Алгоритм вычисления площади и периметра треугольника

Разветвляющейся (ветвящейся) называется конструкция, которая обеспечивает выбор между двумя альтернативами в зависимости от значения входных данных.

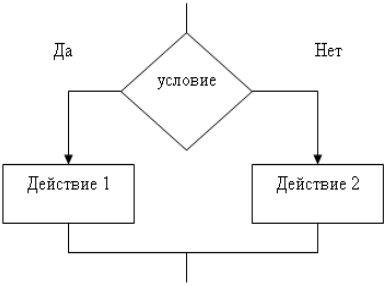
Структура ВЕТВЛЕНИЕ существует в двух основных вариантах: полное и неполное

Полное ветвление

Предполагает выполнение действий для обеих веток в алгоритме:

Если *[условие]*, ***то*** *[действие 1]*, ***иначе*** *[действие 2]*

Блок-схема



Псевдокод

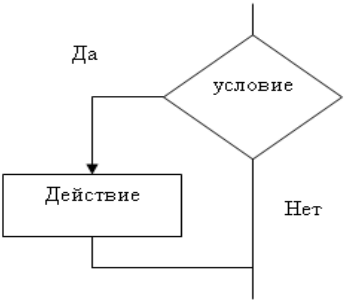
Если [условие]
  то [Действие 1]
    иначе [Действие 2]
конец если

Неполное ветвление

Предполагает выполнение действий только на одной ветви алгоритма (вторая отсутствует):

Если [условие], **то** [действие]

Блок-схема



Псевдокод

Если [условие]
  то [Действие]
конец если

Циклической называется конструкция, в которой некая, идущая подряд группа действий (шагов), может выполняться несколько раз, в зависимости от входных данных или условия задачи. Совокупность повторяющихся действий алгоритма называют **циклом**, т.е. **циклические алгоритмы** включают в себя циклы.

Группа повторяющихся действий на каждом шагу цикла называется **телом цикла**.

Циклический алгоритм включает в себя:

1. Подготовку цикла – действия, связанные с заданием исходных данных, используемых в цикле;

2. Тело цикла – повторяющиеся действия для вычисления искомых величин, а также подготовка значений, необходимых для повторного выполнения действий в теле цикла;

3. Условия продолжения цикла - действия, определяющие необходимость дальнейшего выполнения тела цикла.

Существует несколько видов циклических конструкций, с помощью которых можно организовать циклы. Их можно классифицировать следующим образом (рис. 3):

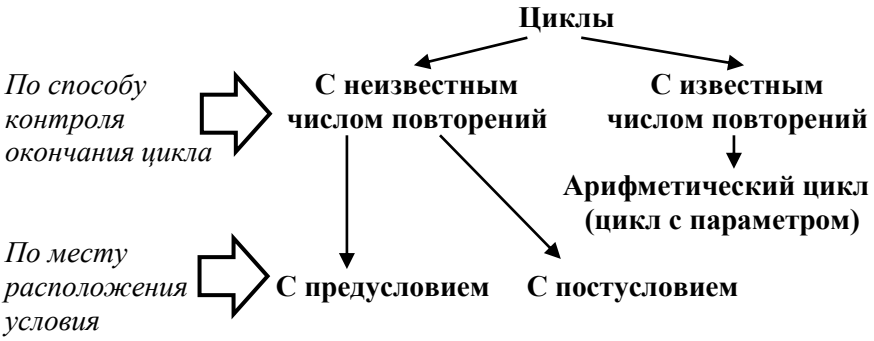


Рис. 3. Классификация циклических конструкций

Циклы, для которых число повторений определяется в ходе выполнения цикла, называются итерационными (циклы с неизвестным числом повторений). В этом случае одно

повторение цикла называется итерацией. Для лучшего понимания их основных отличий друг от друга рассмотрим каждый тип отдельно.

Арифметический цикл (цикл с параметром)

Предполагает, что число итераций заранее известно. Предписывает выполнять тело цикла для всех значений некоторой переменной (параметра цикла) в заданном диапазоне.

Цикл типа ***Для:***

Для всех [параметр цикла] *повторять* [действие]

[параметр цикла] - счетчик количества повторений;

[действие] – тело цикла (последовательность команд, действий).

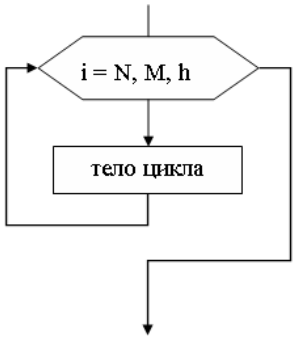
Количество повторений однозначно определяется правилом изменения параметра, которое задается с помощью начального и конечного значений параметра и шагом его изменения. На первом шаге цикла значение параметра равно N, на втором - N+h, на третьем - N+2h и т.д. На последнем шаге цикла значение параметра не больше M, но такое, что дальнейшее его изменение приведет к значению большему, чем M.

i - параметр цикла (является счетчиком количества повторений);

N - начальное значение параметра цикла;

M - конечное значение параметра цикла; h - шаг, с которым изменяется параметр цикла.

Блок-схема



Псевдокод

начало цикла(нц)
для
  i от M до N с шагом h
повторять:
    тело цикла
    (последовательность
     действий)
конец цикла (кц)

Цикл с предусловием

Предполагает, что число итераций заранее не определено и зависит от входных данных задачи. В данной циклической структуре сначала проверяется значение условия перед выполнением очередного шага цикла.

Цикл типа ***Пока:***

Пока <условие> *выполнять* [действие]

<условие> - некоторое проверяемое логическое условие;

[действие] - тело цикла (последовательность команд, действий).

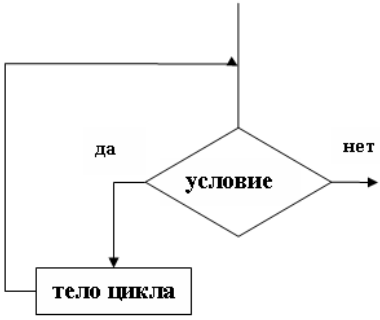
Условие записывается в виде логического выражения, в нем определяется необходимость дальнейшего выполнения повторяющихся действий.

Оператор цикла с предусловием выполняется следующим образом:

- сначала проверяется условие продолжения цикла;
- если это условие истинно, то выполняется тело цикла;
- затем снова проверяется условие продолжения цикла и т.д.;
- если условие продолжения цикла ложно, то происходит выход из цикла.

Особенностью цикла с предусловием является то, что если изначально условие ложно, то тело цикла не выполнится ни разу.

Блок-схема



Псевдокод

начало цикла(нц)
пока
<условие> истинно
выполнять:
тело цикла
(последовательность
действий)
конец цикла (кц)

Цикл с постусловием

Так же, как и в цикле с предусловием, число итераций заранее не определено и зависит от входных данных задачи. Но, в отличие от него, значение условия проверяется после выполнения очередного шага цикла.

Цикл типа ***До:***

Выполнять [действие] *до* тех пор пока <условие> ложно

<условие> - некоторое проверяемое логическое условие;

[действие] - тело цикла (последовательность команд, действий)

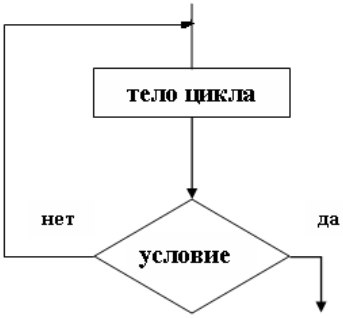
Условие записывается в виде логического выражения, в нем определяется необходимость дальнейшего выполнения повторяющихся действий.

Оператор цикла с постусловием выполняется следующим образом:

- сначала выполняется тело цикла;
- затем проверяется условие продолжения цикла;
- если это условие истинно, то снова выполняется тело цикла, затем снова проверяется условие и т. д.;
- если условие продолжения цикла ложно, то происходит выход из цикла.

Особенностью цикла с постусловием является то, что цикл выполнится хотя бы один раз до проверки условия.

Блок-схема



Псевдокод

начало цикла(нц)
выполнять:
тело цикла
(последовательность действий)
до тех пор пока истинно
<условие>
конец цикла (кц)

Рекурсивная алгоритмическая конструкция

Рекурсивным называется алгоритм, который в процессе выполнения на каком-либо шаге прямо или косвенно обращается сам к себе. Как правило, в основе такого алгоритма лежит рекурсивное определение какого-то понятия. Пример рекурсивного определения - определение факториала числа n:

n!=1*2*3*4* *n

или

$$\begin{cases} n! = 1, \text{при } n \leq 1 \\ n! = n * (n-1)!, \ \text{ при } n > 1 \end{cases}$$

Т.е. рекурсивный объект частично определяется через себя.

Преимущество рекурсивного определения объекта заключается в том, что такое конечное определение теоретически способно описывать бесконечно большое число объектов.

В рекурсивном определении должно присутствовать ограничение, *граничное условие,* при выходе на которое рекурсивный возврат заканчивается. В случае с вычислением n! граничным условием является условие n ≤ 1.

Отличие рекурсивного алгоритма от циклического заключается в том, что в определенной точке алгоритма реализуется тот же самый алгоритм.

3.4. Типовые приемы алгоритмизации

При решении большинства задач используется определенный набор типовых приемов алгоритмизации. Среди них рассмотрим те, которые наиболее часто применяются при решении практических задач.

Комбинированные алгоритмы

На практике часто встречаются задачи, алгоритм решения которых распадается на части, фрагменты и каждый фрагмент представляет собой алгоритм одного из трех описанных типов. Алгоритм, который содержит несколько структур одновременно, называется комбинированным.

В теории алгоритмов доказано, что любой, сколь угодно сложный алгоритм может быть составлен из трех основных алгоритмических структур: линейной, ветвления и цикла. Алгоритмы решения сложных задач могут включать

Вложенные циклы

Частным случаем комбинированных алгоритмов являются алгоритмы, в которых необходимо использовать несколько циклов, изменяющихся одновременно. Например, внутри одного цикла могут находиться один или несколько других циклов. Такие комбинации называются **вложенными циклами**.

Охватывающий цикл называется **внешним**, а вложенные в него циклы - **внутренними** (вложенными), при этом область действия внутреннего цикла должна полностью находиться в области внешнего цикла, т.е. циклы не должны пересекаться.

Правила организации как внешнего, так и внутреннего циклов такие же, как и правила организации простого цикла. Параметры внешнего и внутреннего должны иметь разное имя и изменяться не одновременно, т.е. при одном значении параметра внешнего цикла параметр внутреннего цикла принимает по очереди все свои значения. Перед циклом необходимо задавать начальные значения параметров, а внутри – вычислять текущие.

Пример 7

Составить алгоритм заполнения таблицы данными о зарплате каждого сотрудника за первый квартал 2015 года (табл. 3).

Таблица 3

Данные о зарплате

ФИО сотрудника	Январь	Февраль	Март
Иванов	1000	1200	1300
Петров	1100	1000	1200
Сидоров	2000	2100	2200
Васечкин	3000	3100	3300

Обозначим вводимые данные как элементы двумерного массива {A (i;j)}, где i = 1, 2, 3, 4 (номер строки); j = 1, 2, 3 (номер столбца).

Таблица 4

Обозначение через массив

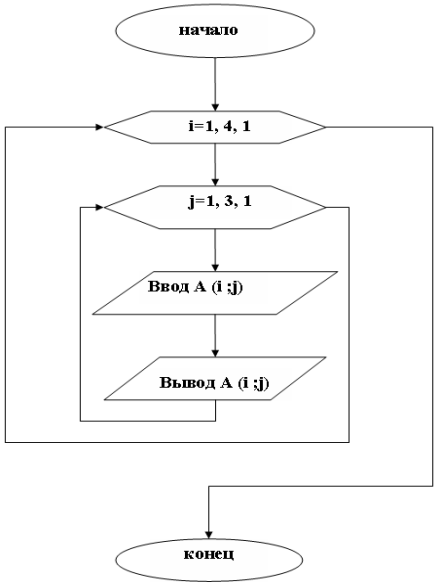
A(1;1)=1000	A(1;2)=1200	A(1;3)=1300
A(2;1)=1100	A(2;2)=1000	A(2;3)=1200
A(3;1)=2000	A(3;2)=2100	A(3;3)=2200
A(4;1)=3000	A(4;2)=3100	A(4;3)=3300

Задача сводится к заполнению двумерного массива.

Входные данные: i, j, A(i;j).

Выходные данные: A(i;j).

Для построения алгоритма решения таких задач используется цикл с параметром, т.к. число шагов цикла известно.



Псевдокод

Начало цикла
Для i=1, 4, 1 повторить:
Начало цикла
Для j=1,3, 1
повторить:
а) ввод A(i;j);
б) вывод A(i;j);
Конец цикла
Конец цикла
Конец

Суммирование

$S=a_1+a_2+a_3+a_4+\ldots\ldots$

При вычислении суммы используется прием накопления.

$S_1=S+a_1$ S – начальное значение суммы

$S_2=S+a_1+a_2=S_1+a_2$ S_i – значение суммы на каждом

$S_3=a_1+a_2+a_3=S_2+a_3$ шаге вычисления

и т. д. a_i – очередное слагаемое

Вычисление суммы сводится к ее накоплению в виде значения переменной в цикле, в котором вычисляются соответствующие слагаемые. При этом вновь вычисленное слагаемое прибавляется к сумме предыдущих слагаемых, т.е. в цикле последовательно вычисляются все промежуточные суммы.

После первого выполнения цикла первая промежуточная сумма должна быть равна значению первого слагаемого. Следовательно, начальное значение S должно быть равно нулю.

Правило суммирования:

начальное значение суммы S=0

в теле циклической конструкции выполняется команда

S = S + <слагаемое>

Формула для накопления суммы

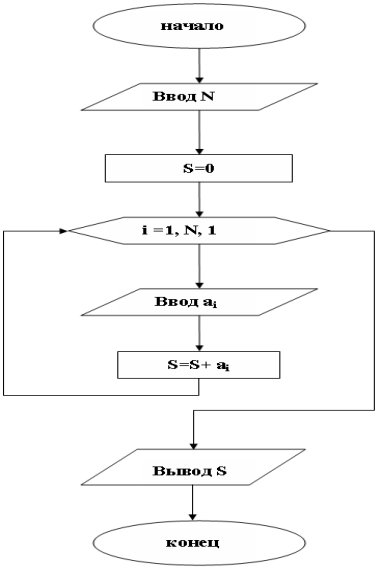
S = S + a, где a - очередное слагаемое; S - промежуточная сумма.

П р и м е р 8

Составить алгоритм вычисления общей суммы выплаченной премии всем сотрудникам фирмы.

Входные данные: N (количество сотрудников фирмы); a_i (размер премии каждого сотрудника); i (параметр).

Выходные данные: S (общая сумма выплат).



Псевдокод

1. Ввод N
2. Начальное значение суммы S=0
3. **Начало цикла**
   Для i=1,N, 1 повторить:
   а) ввод a*
   б) вычисление текущей суммы S=S+ a_i
   Конец цикла
4. Вывод S
5. Конец

Произведение

$P=a_1*a_2*a_3*a_4*$.....

При вычислении произведения используется тот же прием накопления.

$P_1=P*a_1$ P – начальное значение

$P_2=P*a_1*a_2=P_1*a_2$ произведения

$P_3= P*a_1*a_2*a_3=P_2*a_3$ P_i – значение произведения

и т.д. на каждом шаге вычисления

a_i – очередной множитель

Вычисление произведения сводится к его накоплению в цикле в виде значения переменной, при этом в цикле вычисляются последовательно все промежуточные произведения.

После первого выполнения цикла первая промежуточная величина должна быть равна значению первого множителя. Следовательно, начальное значение P, которое задается перед циклом, должно быть равно единице.

Правило умножения:

начальное значение произведения Р=1

в теле циклической конструкции выполняется команда

Р = Р * <множитель>

Формула для накопления произведения

Р = Р * а,

где а – очередной сомножитель; Р – промежуточное произведение.

Подсчет количества элементов

При вычислении количества элементов используется прием накопления, как и при вычислении суммы и произведения. Подсчет количества элементов сводится к вычислению количества натуральных чисел в последовательности 1, 2, 3, 4 и т.д. **Счетчик** искомых элементов обозначается **К.**

Процесс является циклическим, так как каждый раз совершается одно и тоже действие: увеличение предыдущего натурального числа на 1.

При первом подсчете должно получиться значение К, равное единице. Следовательно, начальное значение счетчика должно быть равно нулю.

Правило счетчика:

начальное значение счетчика К=0;

в теле циклической конструкции выполняется команда

К = К+1;

если не задано дополнительных условий

Формула для накопления счетчика:

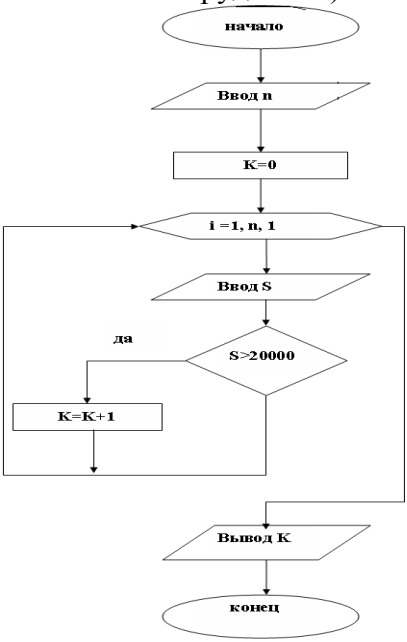
К=К+1

где К – промежуточное значение счетчика.

Пример 9

Составить алгоритм подсчета количества сотрудников фирмы, зарплата которых превышает 20 тыс. руб. (Составить алгоритм подсчета количества чисел, больших 20, среди n заданных).

Входные данные: n (число сотрудников); S (размер зарплаты); i (параметр); Выходные данные: счетчик K (количество сотрудников).



Псевдокод

1. Ввод n
2. Начальное значение счетчика K=0
3. **Начало цикла**
   Для i=1,n, 1 повторить:
   а) ввод S
   б) проверка условия:
   Если S > 20000
   то K=K+1
   Конец если
   Конец цикла
4. Вывод K
5. Конец

***Табулирование функции, заданной одной формулой* Y=f(X)**

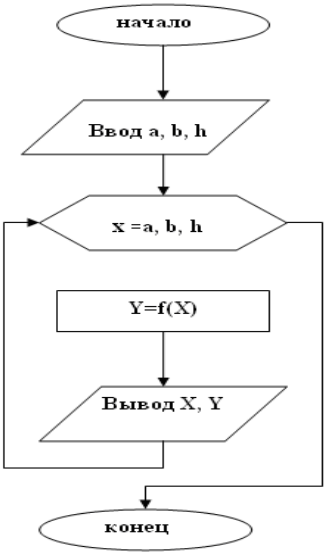
Функция, как правило, определена на заданном участке. Требуется n раз произвести вычисления одного выражения при разных значениях X.

Алгоритм решения такой задачи состоит из следующих основных этапов:

1. Ввод исходных данных a, b, h;
2. Организация цикла по X от начального значения a до конечного значения b с шагом h;
3. Вычисление Y = f(x);
4. Печать X, Y;
5. Конец цикла.

В зависимости от условий конкретной задачи, к перечисленным этапам могут добавляться другие вычисления.

Блок-схему и псевдокод такой задачи в общем случае можно представить в следующем виде:



Псевдокод

1. Ввод a, b, h
2. Начало цикла
Для x = a, b, h
повторить:
а) вычислить *y=f(x)*
б) Вывод x, y
Конец цикла
3. Конец

4. ЗАДАНИЕ КУРСОВОЙ РАБОТЫ

4.1. Общая часть

В соответствии с полученным индивидуальным вариантом задания, дать его содержательное описание, разработать алгоритм решения поставленной задачи и представить его в виде блок-схемы и описания на псевдокоде (программы на языке программирования высокого уровня), привести тестовые примеры работы алгоритма.

4.2. Индивидуальные варианты заданий

Для выполнения курсовой работы необходимо выбрать объект из предложенного ниже списка.

1. Шифр Цезаря. Чтобы зашифровать текст, записанный с помощью русских букв и знаков препинания, его можно переписать, заменив каждую букву непосредственно следующей за ней буквой по алфавиту (буква Я заменяется на А). Обобщив этот способ шифровки, можно производить сдвиг не на одну букву, а на N букв (N – натуральное число). Создать программу, которая А) зашифрует введенный текст, Б) расшифрует данный текст.

2. Шифровка последовательностей нулей и единиц. Способ шифровки последовательностей нулей и единиц (или, например, точек и тире) состоит в следующем. Пусть $a_1, a_2, \ldots a_N$ – такая последовательность. То, что предлагается в качестве ее шифра - это последовательность $b_1, \ldots b_N$, образованная по следующему правилу: $b_1=a_1$, $b_i = 1$ при $a_i=a_{i-1}$ либо $b_i = 0$ иначе (для $i = 2, \ldots N$) Пользуясь изложенным способом: А) зашифровать данную последовательность Б) расшифровать данную последовательность.

3. «Табличная шифровка». Один из простейших способов шифровки текста состоит в табличной замене каждого символа другим символом – его шифром. Выбрать некоторую таблицу, разработать способ ее представления,

затем: А) зашифровать данный текст Б) расшифровать данный текст.

4. «Матричная шифровка». Чтобы зашифровать текст из 121 буквы, его можно записать в квадратную матрицу порядка 11 по строкам, а затем прочитать по спирали, начиная с центра (т.е. с элемента, имеющего индексы 6,6). Такой способ можно обобщить и для произвольной длины текста, подбирая нужный размер матрицы. А) зашифровать данный текст Б) расшифровать данный текст

5. «Шифровка решеткой» Шифровка текста с помощью решетки заключается в следующем. Решетка, т.е. квадрат из клетчатой бумаги 10x10 клеток, некоторые клетки в котором вырезаны, совмещается с целым квадратом 10x10 клеток и через прорези на бумагу наносятся первые буквы текста. Затем решетка поворачивается на 90 градусов и через прорези записываются следующие буквы. Это повторяется еще дважды. Таким образом, на бумагу наносится 100 букв текста. Решетку можно изобразить квадратной матрицей порядка 10 из нулей и единиц (нуль изображает прорезь). Доказано что матрица $[A_{ij}]$, i=1,…10, j=1,…10 может служить ключом шифра, если из элементов A_{ij}, $A_{10-i+1\ j}$, A i 10-j+1, A10-i+1 10 –j+1 в точности один равен нулю. Дана последовательность из 100 букв и матрица-ключ. А) Зашифровать данную последовательность Б) Расшифровать данную последовательность. Обобщить на случай последовательности произвольной длины.

6. «Шифровка зафиксированной перестановкой». Зафиксируем натуральное k и перестановку чисел 1,… k (ее можно задать с помощью последовательности натуральных чисел p1,… pk, в которую входит каждое из чисел 1,… k). При шифровке в исходном тексте к каждой из последовательных групп по k символов применяется зафиксированная перестановка. Пусть k=4 и перестановка есть 3, 2, 4, 1. Тогда группа символов s1,s2,s3,s4 заменяется на s3, s2, s4, s1. Если в последней группе меньше четырех символов, то к ней добавляются пробелы. Пользуясь изложенным способом: А) зашифровать данный текст Б) расшифровать данный текст

7. Шифр Гронсфельда. Этот шифр сложной замены, называемый шифром Гронсфельда, представляет собой модификацию шифра Цезаря числовым ключом. Для этого под буквами исходного сообщения записывают цифры числового ключа. Если ключ короче сообщения, то его запись циклически повторяют. Шифртекст получают примерно, как в шифре Цезаря, но отсчитывают по алфавиту не третью букву (как это делается в шифре Цезаря), а выбирают ту букву, которая смещена по алфавиту на соответствующую цифру ключа. Например, применяя в качестве ключа группу из четырех начальных цифр числа е (основания натуральных логарифмов), а именно 2718, получаем для исходного сообщения ВОСТОЧНЫЙ ЭКСПРЕСС следующий шифртекст (табл. 5):

Таблица 5

Шифртекст

Сообщение		В	О	С	Т	О	Ч	Н	Ы	Й		Э	К	С	П	Р	Е	С	С
Ключ		2	7	1	8	2	7	1	8	2		7	1	8	2	7	1	8	2
Шифртекст		Д	Х	Т	Ь	Р	Ю	О	Г	Л		Д	Л	Щ	С	Ч	Ж	Щ	У

Чтобы зашифровать первую букву сообщения В, используя первую цифру ключа 2, нужно отсчитать вторую по порядку букву от В в алфавите (табл. 6)/

Таблица 6

Пример зашифрования

В	Г	Д
	1	2

получается первая буква шифртекста Д. Пользуясь изложенным способом: А) зашифровать данный текст Б) расшифровать данный текст

8. Шифровка с помощью квадрата Полибия. В Древней Греции (II в. до н.э.) был известен шифр, называемый "квадрат

Полибия". Шифровальная таблица представляла собой квадрат с пятью столбцами и пятью строками, которые нумеровались цифрами от 1 до 5. В каждую клетку такого квадрата записывалась одна буква. В результате каждой букве соответствовала пара чисел, и шифрование сводилось к замене буквы парой чисел. Для латинского алфавита квадрат Полибия имеет вид (табл. 7):

Таблица 7

Квадрат Полибия

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I, J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Пользуясь изложенным способом: А) зашифровать данный текст Б) расшифровать данный текст

9. Шифр Хилла (с длиной блока = 2). Криптосистема, основанная Хиллом, базируется на линейной алгебре. Пространства исходных сообщений и криптотекстов совпадают: латинский алфавит. Перенумеруем буквы в порядке их следования в алфавите: **A** получает номер 0, **B** - номер 1, ... и **Z** - номер 25. Все арифметические операции выполняются по модулю 26 (длина алфавита), то есть 26 отождествляется с 0, 27 - с единицей и т.д. Выберем целое число **D** <= 2. Оно указывает размерность используемых матриц. В процедуре шифрования наборы из **D** букв шифруются вместе. Возьмем **D** = 2. Пусть ключ **M** - квадратная матрица порядка **D**, элементами которой являются числа 0 .. 25. Эта матрица должна удовлетворять требованию невырожденности, т.е. для нее должна существовать матрица M^{-1}, например:

$$M = \begin{vmatrix} 3 & 3 \\ 2 & 5 \end{vmatrix}, \text{ и } M^{-1} = \begin{vmatrix} 15 & 17 \\ 20 & 9 \end{vmatrix}$$

(вся арифметика ведется по модулю 26)

Шифрование осуществляется с помощью уравнения MP = C, где **P** и **C** - вектор столбцы длиной **D**. То есть, каждый набор из **D** букв исходного сообщения определяет вектор **P**, компонентами которого являются номера букв. В свою очередь, полученный вектор C также интерпретируется как набор из D букв. Например: исходное сообщение: HELP определяет 2 вектора (по 2 буквы в каждом):

$$P_1 = \begin{vmatrix} H \\ E \end{vmatrix} = \begin{vmatrix} 7 \\ 4 \end{vmatrix} \text{ и } P_2 = \begin{vmatrix} L \\ P \end{vmatrix} = \begin{vmatrix} 11 \\ 15 \end{vmatrix}$$

Из уравнений

$$M*P_1 = \begin{vmatrix} 7 \\ 8 \end{vmatrix} = C1 \text{ и } M*P_2 = \begin{vmatrix} 0 \\ 19 \end{vmatrix} = C2$$

получаем зашифрованный текст HIAT...

Для дешифровки сообщения используем матрицу M^{-1} [mod 26] и для шифротекста **C** вычисляем $P = M^{-1} *C$ [mod 26]. Пользуясь изложенным способом: А) зашифровать данный текст Б) расшифровать данный текст

<u>10. Шифр Плейфера</u>. Шифр Плейфера использует матрицу 5x5 (для латинского алфавита, для русского алфавита необходимо увеличить размер матрицы до 6x6), содержащую ключевое слово или фразу. Для создания матрицы и использования шифра достаточно запомнить ключевое слово и четыре простых правила. Чтобы составить ключевую матрицу, в первую очередь нужно заполнить пустые ячейки матрицы буквами ключевого слова (не записывая повторяющиеся символы), по-

том заполнить оставшиеся ячейки матрицы символами алфавита, не встречающимися в ключевом слове, по порядку (в английских текстах обычно опускается символ «Q», чтобы уменьшить алфавит, в других версиях «I» и «J» объединяются в одну ячейку). Ключевое слово может быть записано в верхней строке матрицы слева направо, либо по спирали из левого верхнего угла к центру. Ключевое слово, дополненное алфавитом составляет матрицу 5x5 и является ключом шифра. Для того, чтобы зашифровать сообщение необходимо разбить его на биграммы (группы из двух символов), например «Hello World» становится «HE LL OW OR LD», и отыскать эти биграммы в таблице. Два символа биграммы соответствуют углам прямоугольника в ключевой матрице. Определяем положения углов этого прямоугольника относительно друг друга. Затем руководствуясь следующими 4 правилами зашифровываем пары символов исходного текста:

1. Если два символа биграммы совпадают, добавляем после первого символа «X», зашифровываем новую пару символов и продолжаем. В некоторых вариантах шифра Плейфера вместо «X» используется «Q».

2. Если символы биграммы исходного текста встречаются в одной строке, то эти символы замещаются на символы, расположенные в ближайших столбцах справа от соответствующих символов. Если символ является последним в строке, то он заменяется на первый символ этой же строки.

3. Если символы биграммы исходного текста встречаются в одном столбце, то они преобразуются в символы того же столбца, находящимися непосредственно под ними. Если символ является нижним в столбце, то он заменяется на первый символ этого же столбца.

4. Если символы биграммы исходного текста находятся в разных столбцах и разных строках, то они заменяются на символы, находящиеся в тех же строках, но соответствующие другим углам прямоугольника. Для расшифровки необходимо использовать инверсию этих четырёх правил, откидывая символы «X» (или «Q»), если они не несут смысла в исходном сообщении.

Пользуясь изложенным способом: А) зашифровать данный текст Б) расшифровать данный текст

11. Шифр Вижинера (для латинских букв). Квадрат Виженера или таблица Виженера, так же известная как tabula recta, может быть использована для заширования и расшифрования.

В шифре Цезаря каждая буква алфавита сдвигается на несколько позиций; например в шифре Цезаря при сдвиге +3, A стало бы D, B стало бы E и так далее. Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифрования может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря (табл. 8). На разных этапах кодировки шифр Виженера использует различные алфавиты из этой таблицы. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова.

Например, предположим, что исходный текст имеет вид:

ATTACKATDAWN

Человек, посылающий сообщение, записывает ключевое слово(«LEMON») циклически до тех пор, пока его длина не будет соответствовать длине исходного текста: LEMONLEMONLE

Таблица 8

Таблицы шифра Цезаря

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	Z	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	X	A	B	S
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	S	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	V	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	S	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	S	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	G	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Первый символ исходного текста A зашифрован последовательностью L, которая является первым символом ключа. Первый символ L шифрованного текста находится на пересечении строки L и столбца A в таблице Виженера. Точно так же для второго символа исходного текста используется второй символ ключа; то есть второй символ шифрованного текста X получается на пересечении строки E и столбца T. Остальная часть исходного текста шифруется подобным способом.

Расшифрование производится следующим образом: находим в таблице Виженера строку, соответствующую первому символу ключевого слова; в данной строке находим первый символ зашифрованного текста. Столбец, в котором находится данный символ, соответствует первому символу

исходного текста. Следующие символы зашифрованного текста расшифровываются подобным образом.

$$C_i = P_i + K_i \pmod{26}$$

Из наблюдения за частотой совпадения следует:

$$P_i = C_i - K_i \pmod{26}$$

Пользуясь изложенным способом: А) зашифровать данный текст Б) расшифровать данный текст

12. Шифр с использованием кодового слова. Шифр с использованием кодового слова является одним из самых простых как в реализации так и в расшифровывании. Идея заключается в том что выбирается кодовое слово которое пишется впереди, затем выписываются остальные буквы алфавита в своем порядке. Шифр с использованием кодового слова WORD. Исходный алфавит: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Алфавит замены: W O R D A B C E F G H I J K L M N P Q S T U V X Y Z Как мы видим при использовании короткого кодового слова мы получаем очень и очень простую замену. Так же мы не можем использовать в качестве кодового слова слова с повторяющимися буквами, так как это приведет к неоднозначности расшифровки, то есть двум различным буквам исходного алфавита будет соответствовать одна и та же буква шифрованного текста. Пользуясь изложенным способом: А) зашифровать данный текст Б) расшифровать данный текст

13. Шифр перестановки "скитала" Известно, что в V веке до нашей эры правители Спарты, наиболее воинственного из греческих государств, имели хорошо отработанную систему секретной военной связи и шифровали свои послания с помощью скитала, первого простейшего криптографического устройства, реализующего метод простой перестановки. Шифрование выполнялось следующим образом. На стержень цилиндрической формы, который назывался скитала,

наматывали спиралью (виток к витку) полоску пергамента и писали на ней вдоль стержня несколько строк текста сообщения. Затем снимали со стержня полоску пергамента с написанным текстом. Буквы на этой полоске оказывались расположенными хаотично. Такой же результат можно получить, если буквы сообщения писать по кольцу не подряд, а через определенное число позиций до тех пор, пока не будет исчерпан весь текст (рис. 4).

Рис. 4. Пример шифра скитала

Сообщение НАСТУПАЙТЕ при размещении его по окружности стержня по три буквы дает шифртекст

НУТАПЕСА_ТЙ

Для расшифрования такого шифртекста нужно не только знать правило шифрования, но и обладать ключом в виде стержня определенного диаметра. Зная только вид шифра, но не имея ключа, расшифровать сообщение было непросто. Пользуясь изложенным способом: А) зашифровать данный текст Б) расшифровать данный текст

14. Простая табличная перестановка. Одним из самых примитивных табличных шифров перестановки является простая перестановка, для которой ключом служит размер таблицы. Этот метод шифрования сходен с шифром скитала. На- пример, сообщение

ТЕРМИНАТОР ПРИБЫВАЕТ СЕДЬМОГО В ПОЛНОЧЬ

записывается в таблицу поочередно по столбцам. Результат заполнения таблицы из 5 строк и 7 столбцов показан на рис. 5.

После заполнения таблицы текстом сообщения по столбцам для формирования шифртекста считывают содержимое таблицы по строкам.

Т	Н	П	В	Е	Г	Л
Е	А	Р	А	Д	О	Н
Р	Т	И	Е	Ь	В	О
М	О	Б	Т	М	П	Ч
И	Р	Ы	С	О	О	Ь

Рис. .. Пример табличной перестановки

Если шифртекст записывать группами по пять букв, получается такое шифрованное сообщение:

ТНПВЕ ГЛЕАР АДОНР ТИЕЬВ ОМОБТ МПЧИР ЫСООЬ

Естественно, отправитель и получатель сообщения должны заранее условиться об общем ключе в виде размера таблицы. Следует заметить, что объединение букв шифртекста в 5-буквенные группы не входит в ключ шифра и осуществляется для удобства записи несмыслового текста. При расшифровании действия выполняют в обратном порядке. Пользуясь изложенным способом: А) зашифровать данный текст Б) расшифровать данный текст

15. Табличная шифровка с ключевым словом. Несколько большей стойкостью к раскрытию обладает метод шифрования, называемый одиночной перестановкой по ключу. Этот метод отличается от предыдущего тем, что столбцы таблицы переставляются по ключевому слову, фразе или

набору чисел длиной в строку таблицы (рис. 6). Применим в качестве ключа, например, слово ПЕЛИКАН, а текст сообщения возьмем из предыдущего примера.

П	Е	Л	И	К	А	Н
7	2	5	3	4	1	6
Т	Н	П	В	Е	Г	Л
Е	А	Р	А	Д	0	Н
Р	Т	И	Е	Ь	В	О
М	о	Б	Т	М	П	Ч
И	Р	Ы	С	О	О	Ь

До перестановки

А	Е	И	К	Л	Н	П
1	2	3	4	5	6	7
Г	Н	В	Е	П	Л	Т
0	А	А	Д	Р	Н	Е
В	Т	Е	Ь	И	0	Р
П	0	Т	М	Б	Ч	М
О	Р	С	О	Ы	Ь	И

После перестановки

Рис. 6. Табличная перестановка с ключевым словом

На рис. 6 показаны две таблицы, заполненные текстом сообщения и ключевым словом, при этом левая таблица соответствует заполнению до перестановки, а правая таблица - заполнению после перестановки. В верхней строке левой таблицы записан ключ, а номера под буквами ключа определены в соответствии с естественным порядком соответствующих букв ключа в алфавите. Если бы в ключе встретились одинаковые буквы, они бы были понумерованы слева направо. В правой таблице столбцы переставлены в соответствии с упорядоченными номерами букв ключа. При считывании содержимого правой таблицы по строкам и записи шифртекста группами по пять букв получим шифрованное сообщение: ГНВЕП ЛТООА ДРНЕВ ТЕЬИО РПОТМ БЧМОР СОЬЬИ Пользуясь изложенным способом: А) зашифровать данный текст Б) расшифровать данный текст

16. Двойная табличная перестановка. Для обеспечения дополнительной скрытности можно повторно зашифровать сообщение, которое уже прошло шифрование. Такой метод шифрования называется двойной перестановкой. В случае двойной перестановки столбцов и строк таблицы перестановки определяются отдельно для столбцов и отдельно для строк. Сначала в таблицу записывается текст сообщения, а потом

поочередно переставляются столбцы, а затем строки. При расшифровании порядок перестановок должен быть обратным.

Пример выполнения шифрования методом двойной перестановки показан на рис. 7. Если считывать шифртекст из правой таблицы построчно блоками по четыре буквы, то получится следующее:

ТЮАЕ ООГМ РЛИП ОЬСВ

	4	1	3	2
3	П	Р	И	Л
1	Е	Т	А	Ю
4	В	О	С	Ь
2	М	О	Г	О

Исходная таблица

	1	2	3	4
3	Р	Л	И	П
1	Т	Ю	А	Е
4	О	Ь	С	В
2	О	О	Г	М

Перестановка столбцов

	1	2	3	4
1	Т	Ю	А	Е
2	О	О	Г	М
3	Р	Л	И	П
4	О	Ь	С	В

Перестановка строк

Рис. 7. Метод двойной перестановки

Ключом к шифру двойной перестановки служит последовательность номеров столбцов и номеров строк исходной таблицы (в нашем примере последовательности 4132 и 3142 соответственно).

Пользуясь изложенным способом: А) зашифровать данный текст Б) расшифровать данный текст

17. Обучение переводу из 10-й системы счисления в двоичную. Составить программу для обучения переводу чисел из десятичной системы счисления в двоичную и обратно. Программа должна предлагать десятичное (двоичное) число, выбранное с помощью датчика случайных чисел, обучающийся – назвать это число в двоичной (десятичной) системе счисления. Причем, должен быть контроль за временем на размышление.

18. Обучение переводу из 10-й системы счисления в 16-ричную. Составить программу для обучения переводу чисел из десятичной системы счисления в 16-ричную и обратно. Программа должна предлагать десятичное (16-ричное) число, выбранное с помощью датчика случайных чисел, обучающийся –

назвать это число в 16-ричной (десятичной) системе счисления. Причем, должен быть контроль за временем на размышление.

19. Шифровка с помощью магического квадрата. В средние века для шифрования перестановкой применялись и магические квадраты. Магическими квадратами называют квадратные таблицы с вписанными в их клетки последовательными натуральными числами, начиная от 1, которые дают в сумме по каждому столбцу, каждой строке и каждой диагонали одно и то же число. Шифруемый текст вписывали в магические квадраты в соответствии с нумерацией их клеток. Если затем выписать содержимое такой таблицы по строкам, то получится шифртекст, сформированный благодаря перестановке букв исходного сообщения. В те времена считалось, что созданные с помощью магических квадратов шифртексты охраняет не только ключ, но и магическая сила. Пример магического квадрата и его заполнения сообщением

ПРИЛЕТАЮ ВОСЬМОГО

показан ниже (рис. 8)

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

О	И	Р	М
Е	О	С	Ю
В	Т	А	Ь
Л	Г	О	П

Рис. 8. Магический квадрат

Шифртекст, получаемый при считывании содержимого правой таблицы по строкам, имеет вполне загадочный вид:

ОИРМ ЕОСЮ ВТАЬ ЛГОП

Число магических квадратов быстро возрастает с увеличением размера квадрата. Существует только один магический квадрат размером 3x3 (если не учитывать его повороты). Количество магических квадратов 4x4 составляет

уже 880, а количество магических квадратов 5x5 - около 250000. Пользуясь изложенным способом: А) зашифровать данный текст Б) расшифровать данный текст

20. Шифровка «тарабарская грамота». Суть шифровки, которая использовалась в 15-16 веках на Руси, в следующем. Все согласные буквы русской азбуки записывались 20 в два ряда; одна половина букв вверху, другая половина — внизу, причем в обратном порядке (одна буква под другой):

Б В Г Д Ж З К Л М Н
Щ Ш Ч Ц Х Ф Т С Р П

При зашифровке слов согласные взаимно заменялись, а гласные, Й и буквы Ъ, Ь вписывались без изменений. Слова записывались без промежутков между ними, как вообще писался любой текст до 16 века. и это еще больше затрудняло разгадывание. Пользуясь изложенным способом: А) зашифровать данный текст Б) расшифровать данный текст

21. Шифровка «тарабарская грамота» с гласными буквами. Добавим к правилам шифровки согласных (см. предыдущую тему) правило замены гласных при шифровке по правилу:

А Е Ё И О
Я Ю Э Ы У

Пользуясь изложенным способом: А) зашифровать данный текст Б) расшифровать данный текст

22. Шифровка «тарабарская грамота» (весь алфавит). Пусть первые 10 согласных букв русского алфавита заменяются на соотвествующие гласные, а остальные в соответствии с таблицей

Б В Г Д Ж З К Л М Н П Р С Т Ф
А Е Ё И О У Ы Э Ю Я Щ Ш Ч Ц Х

Пользуясь изложенным способом: А) зашифровать данный текст Б) расшифровать данный текст

23. «100 спичек». Из кучки, первоначально содержащей 100 спичек, двое играющих поочередно берут по несколько спичек: не менее одной и не более десяти. Проигрывает взявший последнюю спичку .

24. «Угадай число». Один из играющих задумывает число от 1 до 1000, другой пытается угадать его за десять вопросов вида: верно ли, что задуманное число больше такого-то числа. Написать программу, играющую за отгадчика.

25. «Ним». Имеется три кучки спичек. Двое играющих по очереди делают ходы. Каждый ход заключается в том, что из одной какой-то кучки берется произвольное ненулевое число спичек. Выигрывает взявший последнюю спичку.

26. «Цзяньшидзы». Имеются две кучки камней. Двое играющих по очереди делают ходы. Каждый ход может состоять в одном из двух:

1) берется произвольное ненулевое число камней из какой- то одной кучки;

2) берется одновременно по одинаковому ненулевому числу камней из обеих кучек.

Выигрывает взявший последний камень. Пара (a,b) , где a и b – количество камней в кучках при a<b, является проигрышной, если число a оканчивается в «фибоначчиевой» системе четным числом нулей, а число b получается из a приписыванием еще одного нуля в конце.

27. Магический квадрат. Магическим квадратом порядка N называется квадратная таблица размера NxN , составленная из чисел 1, 2, … N^2 так, что суммы по каждому столбцу, каждой строке и каждой из двух диагоналей равны между собой. Составить программу для построения магического квадрата для заданного N.

28. Латинский квадрат. Латинским квадратом порядка N называется квадратная таблица размером NxN , каждая строка и каждый столбец которой содержит числа 1,2, … N.

Составить программу для построения латинского квадрата для заданного N.

29. Перевод десятичного числа в двоичное. Написать программу, которая преобразует введенное пользователем десятичное число (возможно, с дробью) в двоичное.

30. Проверка знания таблицы умножения. Написать программу проверки знания таблицы умножения. Программа должна вывести 10 примеров и выставить оценку: за 10 правильных ответов – "отлично", за 9 и 8 – "хорошо", за 7 и 6 – "удовлетворительно", за 6 и менее – "плохо".

31. Проверка умения складывать и вычитать числа. Написать программу проверки умения складывать и вычитать числа в пределах 100. Программа должна вывести 10 примеров, причем в каждом примере уменьшаемое должно быть больше или равно вычитаемому, т. е. не допускается предлагать испытуемому примеры с отрицательным результатом. Оценка выставляется по следующему правилу: за 10 правильных ответов – «отлично», за 9 и 8 – «хорошо», за 7 и 6 – «удовлетворительно», за 6 и менее – «плохо».

32. Преобразование десятичного целого числа в указанную систему счисления. Написать программу, которая преобразует введенное пользователем десятичное целое число в число в указанной системе счисления (от 2 до 10).

33. Преобразование десятичного числа в шестнадцатиричное. Написать программу, которая преобразует введенное пользователем десятичное число (в том числе и дробное) в шестнадцатеричное.

34. Вычисление значения выражения, заданного строкой символов. Написать программу, которая вычисляет значение выражения $N_0O_1N_1O_2\ldots O_kN_k$, где N_i – целое одноразрядное число, O_i – один из двух знаков простейших арифметических действий: сложения (+) или вычитания.

35. Итоги Олимпийских игр. Написать программу, которая подводит итоги Олимпийских игр. Программа должна получить от пользователя число медалей разного достоинства, завоеванное каждой командой-участницей, вычислить общее

количество медалей и соответствующее ему число очков, и после этого упорядочить список в соответствии с набранным количеством очков. Количество очков вычисляется по следующему правилу: за золотую медаль команда получает семь очков, за серебряную – шесть, за бронзовую – пять очков.

36. Игра «Угадай число». Написать программу, реализующую игру «угадай число». Правила игры следующие. Играют двое. Один задумывает число, второй — угадывает. На каждом шаге угадывающий делает предположение, а задумавший число – говорит, сколько цифр числа угаданы и сколько из угаданных цифр занимают правильные позиции в числе. Например, если задумано число 725 и выдвинуто предположение, что задумано число 523, то угаданы две цифры (5 и 2) и одна из них (2) занимает верную позицию.

37. Программа-телеграф. Напишите программу-телеграф, которая принимает от пользователя сообщение и выводит его на экран в виде последовательности точек и тире. Вывод точек и тире можно сопровождать звуковым сигналом соответствующей длительности. Азбука Морзе для букв русского алфавита приведена на рис. 9.

А	.-	Б	-...	В	.---	Г	--.
Д	-..	Е	.	Ж	...-	З	--..
И	..	Й	.---	К	-.-	Л	.-..
М	--	Н	-.	О	---	П	.--.
Р	.-.	С	…	Т	-	У	..-
Ф	..-.	Х	….	Ц	-.-.	ч	---.
Ш	----	Щ	--.-	ъ	-..-	ы	-.--
ь	-..-	Э	..-.	Ю	..—	Я	.-.-

Рис. 9. Азбука Морзе

4.3. Контрольные вопросы

1. Что такое алгоритм?

2. Перечислите основные правила составления алгоритма.

3. Перечислите основные свойства алгоритма.

4. Что такое алгоритмизация?

5. Перечислите основные способы описания алгоритма.

6. Что представляет собой словесное описание алгоритма? Приведите пример.

7. Что представляет собой формульно-словесное описание алгоритма? Приведите пример.

8. Что такое псевдокод? Приведите пример.

9. Что такое блок-схема алгоритма? Приведите пример.

10. Какими геометрическими фигурами изображаются в блок-схеме операции разного вида?

11. Что такое программа? Приведите пример.

12. Перечислите основные алгоритмические конструкции.

13. Что представляет собой линейная алгоритмическая конструкция? Приведите пример.

14. Что представляет собой разветвляющаяся алгоритмическая конструкция? Приведите пример.

15. Что представляет собой циклическая алгоритмическая конструкция? Приведите пример.

16. Что представляет собой арифметический цикл (цикл с параметром)? Приведите пример.

17. Что представляет собой цикл с предусловием? Приведите пример.

18. Что представляет собой цикл с постусловием? Приведите пример.

19. Что представляет собой рекурсивная алгоритмическая конструкция? Приведите пример.

20. Что представляет собой вложенные циклы? Приведите пример.

21. Приведите алгоритмы суммирования, формулу для накопленной суммы.

22. Приведите алгоритмы вычисления произведения, формулу для накопленного произведения.

23. Приведите алгоритмы подсчета количества элементов.

24. Приведите алгоритм табулирования функции заданной одной формулой и заданной несколькими формулами.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Аляев, Ю. А. Алгоритмизация и языки программирования Pascal, C++, Visual Basic [Текст]: учеб.-справ. пособие для вузов / Ю. А. Аляев. – М.: Финансы и статистика, 2004. – 320 с.

2. Аляев, Ю. А. Практикум по алгоритмизации и программированию на языке Паскаль [Текст]: учеб. пособие / Ю. А. Аляев, В. П. Гладков, О. А. Козлов. – М.: Финансы и статистика, 2004. – 528 с.

3. Бочарова, Т. А. Основы алгоритмизации [Текст]: учеб. пособие / Т. А. Бочарова, Н. О. Бегункова. – Хабаровск: Изд-во Тихоокеан. гос. ун-та, 2011. – 64 с.

4. Златопольский, Д. М. Сборник задач по программированию [Текст] / Д. М. Златопольский. – 2-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2007. – 240 с.

5. Информатика [Текст]: учебник / Б. В. Соболь и др. – изд. 3-е, доп. и перераб. – Ростов-на-Дону: Феникс, 2007. – 446 с.

6. Колдаев, В. Д. Основы алгоритмизации и программирования [Текст]: учеб. пособие / В. Д. Колдаев; под ред. проф. Л. Г. Гагариной. – М.: ФОРУМ: ИНФРА – М., 2006. – 416 с.

7. Кормен, Т. Алгоритмы: построение и анализ [Текст]/ Т. Кормен, Ч. Лейзерсон, Р. Ривест. – М.: МЦНМО, 2002. – 955 с

8. Новожилов, О. П. Информатика [Текст]: учеб. пособие / О. П. Новожилов. – М.: Юрайт, 2011. – 594 с.

9. Пентюхов, В. В. Информатика. Теоретические основы [Текст]: учеб. пособие / В. В. Пентюхов, Г. А. Кащенко, С. И. Лавлинский. – Воронеж: Изд-во ВГТУ, 2000. – 168 с.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 1
1. ЦЕЛИ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ 2
2. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И ОБЪЁМУ КУРСОВОЙ РАБОТЫ 3
2.1. График выполнения курсовой работы 4
2.2. Последовательность выполнения 4
2.3. Критерии оценки курсовой работы 5
3. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ 6
3.1. Понятие алгоритма 6
3.2. Способы описания алгоритмов 6
3.3. Основные алгоритмические конструкции (виды алгоритмов) 13
3.4. Типовые приемы алгоритмизации 20
4. ЗАДАНИЕ КУРСОВОЙ РАБОТЫ 28
4.1. Общая часть 28
4.2. Индивидуальные варианты заданий 28
4.3. Контрольные вопросы 46
БИБЛИОГРАФИЧЕСКИЙ СПИСОК 48

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к курсовой работе по дисциплине
«Информатика»
для студентов специальностей
090301 «Компьютерная безопасность»,
090302 «Информационная безопасность
телекоммуникационных систем»,
090303 «Информационная безопасность
автоматизированных систем»
очной формы обучения

Составитель
Чопоров Олег Николаевич

В авторской редакции

Подписано к изданию 01.06.2015.
Уч. - изд. л. 3,0.

ФГБОУ ВПО «Воронежский государственный
технический университет»
394026 Воронеж, Московский просп., 14