

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

**Федеральное государственное образовательное бюджетное
учреждение высшего профессионального образования
«САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
им. проф. М. А. БОНЧ-БРУЕВИЧА»**

И. А. Липанова, В. А. Медведев, С. В. Хорошенко

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

**УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ
ПО ВЫПОЛНЕНИЮ КУРСОВЫХ РАБОТ**

СПбГУТ)))

**САНКТ-ПЕТЕРБУРГ
2015**

УДК 025.4.03; 002.6:004.65; 002.6:004.89

ББК 32.811я7

П16

Рецензент

доктор технических наук, профессор кафедры прикладной математики
и информационных технологий Санкт-Петербургского университета
Государственной противопожарной службы МЧС России *А. Ю. Иванов*

*Утверждено редакционно-издательским советом СПбГУТ
в качестве учебно-методического пособия*

Липанова И. А., Медведев В. А., Хорошенко С.В.

П16 Алгоритмизация и программирование: учебно-методическое пособие по выполнению курсовых работ/
И. А. Липанова, В. А. Медведев, С. В. Хорошенко – СПб. :
Издательство СПбГУТ, 2015. – 38 с.

В настоящем учебно-методическом пособии изложены цели, содержание и правила оформления курсовой работы по дисциплине «Алгоритмизация и программирование». Подробно рассмотрены основные этапы выполнения курсовой работы, требования к содержанию пояснительной записки, ее оформление и рекомендации по защите курсовой работы. Приведены примерные темы курсовой работы.

Предназначено для бакалавров, магистров направлений подготовки 022000, 090900, 100100, 200700, 210100, 210400, 210700, 210701, 220700, 221700, 230100, 230400, а также аспирантов и специалистов в области телекоммуникаций.

УДК 025.4.03; 002.6:004.65;

002.6:004.89

ББК 32.811я7

© Липанова И. А., Медведев В. А., Хорошенко С. В. 2015

© Федеральное государственное образовательное
бюджетное учреждение высшего профессионального
образования «Санкт-Петербургский государственный
университет телекоммуникаций
им. проф. М. А. Бонч-Бруевича», 2015

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 ХАРАКТЕРИСТИКА КУРСОВОЙ РАБОТЫ	5
2 СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ	13
3 ОФОРМЛЕНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ	15
3.1 Общие требования	15
3.2 Заголовки.....	15
3.3 Содержание.....	16
3.4 Иллюстрации	16
3.5 Таблицы	17
3.6 Формулы и уравнения	18
3.7 Перечисления.....	19
3.8 Схемы алгоритмов.....	19
3.9 Список использованных источников	21
3.10 Приложения	24
4 ЗАЩИТА КУРСОВОЙ РАБОТЫ.....	25
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	26
ПРИЛОЖЕНИЕ А. ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ	27
ПРИЛОЖЕНИЕ Б. ПРИМЕР ТИТУЛЬНОГО ЛИСТА.....	37

ВВЕДЕНИЕ

Курсовая работа по дисциплине «Алгоритмизация и программирование» имеет целью научить студентов применять полученные знания для самостоятельного решения на компьютере конкретных практических задач. Она направлена на закрепление и углубление теоретических знаний и практических навыков решения вычислительных задач на компьютере и получение навыков в оформлении курсовой работы.

Курсовая работа выполняется студентами, специализирующимися по следующим направлениям подготовки:

022000.62 Экология и природопользование;

090900.62 Информационная безопасность;

100100.62 Сервис;

200700.62 Фотоника и оптоинформатика;

210100.62 Электроника и наноэлектроника;

210400.62 Радиотехника;

210700.62 Инфокоммуникационные технологии и системы связи, профили - Защищенные системы и сети связи, Оптические системы и сети связи, Системы мобильной связи, Цифровое телерадиовещание, Инфокоммуникационные технологии в сервисах и услугах связи, Сети связи и системы коммутации;

220700.62 Автоматизация технологических процессов и производств;

221700.62 Стандартизация и метрология;

230100.62 Информатика и вычислительная техника;

230400.62 Информационные системы и технологии, профили -

Информационные системы и технологии, Информационные технологии в дизайне, Безопасность информационных систем.

Курсовая работа направлена на формирование следующих компетенций:

владеть культурой мышления, способностью к обобщению, анализу, восприятию информации, постановке цели и выбору путей ее достижения;

- владеть основными методами, способами и средствами получения, хранения, переработки информации, иметь навыки работы с компьютером как средством управления информацией;

- способность использовать инструментальные средства и системы программирования для решения профессиональных задач;

- готовность применять алгоритмическое мышление;

- осваивать методики использования программных средств для решения практических задач.

Полученные в ходе курсовой работы знания и навыки являются базовыми для выполнения последующих курсовых работ, а также для разработки дипломного проекта.

1 ХАРАКТЕРИСТИКА КУРСОВОЙ РАБОТЫ

Существо курсовой работы состоит в разработке алгоритма и программы на языке высокого уровня в соответствии с заданием (см. Приложение А).

Выполнение задания целесообразно разбить на ряд относительно самостоятельных этапов, которые вытекают из общей последовательности решения задачи на компьютере. К этим этапам относятся:

- уяснение задания курсовой работы;
- формализация (выбор способа решения) задачи и разработка алгоритма её решения;
- составление программы;
- тестирование и отладка программы;
- анализ полученных результатов.

Начальным этапом курсовой работы является уяснение задания. Опыт показывает, что для того, чтобы минимизировать количество ошибок при разработке программы, необходимо вникнуть в содержание задачи путем уяснения её условия, четко представить себе основные функции, которые надо реализовать, исходные данные для работы программы, накладываемые ограничения и так далее. На основе этого составляется ориентировочный план работы.

Этап формализации (выбор способа решения) задачи и разработки алгоритма является довольно ответственным и включает в себя следующие виды работ:

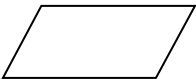


- изучение необходимой литературы;
- выбор способа (математического метода) решения задачи;
- выбор форм организации данных;
- выбор рациональной схемы вычислений в вычислительных задачах или подходящего способа поиска, упорядочения, сортировки в информационно-поисковых задачах;
- разработку алгоритма решения задачи.

После выбора способа (математического метода) решения задачи и форм организации данных, разработка алгоритма является важнейшей задачей, так как в алгоритме находят отражение основные идеи и порядок её решения. Алгоритм полностью определяет структуру программы, позволяет облегчить ее написание, предотвратить возникновение ошибок и, следовательно, сократить время отладки.



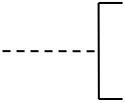

В основу разработки алгоритма целесообразно положить метод пошаговой детализации. Суть метода состоит в следующем. Разработка алгоритма производится по принципу «от общего к частному, сверху вниз». На первом шаге детализации обычно разрабатывается укрупненная схема алгоритма, в которой хорошо просматривается основная идея выбранного метода решения задачи. Она является как бы генеральным

планом и обычно имеет вид простой последовательности. На втором и последующих шагах раскрываются отдельные функциональные блоки путем замены их стандартными блоками (элементами). Основные элементы блок-схем алгоритмов представлены в таблице 1.

Таблица 1

Символ	Название	Описание
	Данные	Отображает данные, носитель которых не определен. Используется для ввода-вывода данных
	Процесс	Функция обработки данных любого вида (выполнение определенной операции или группы операций, приводящее к изменению значения, формы или размещения информации)
	Предопределенный процесс	Отображает предопределенный процесс, состоящий из одной или нескольких операций или шагов программы, которые определены в другом месте (в подпрограмме, модуле)
	Подготовка	Отображает модификацию команды или группы команд с целью воздействия на некоторую последующую функцию
	Решение	Символ отображает решение или функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, один и только один из которых может быть активирован после вычисления условий.
	Линия	Отображает поток данных или управления. При необходимости для повышения удобочитаемости могут быть добавлены стрелки-указатели

Продолжение таблицы 1

Символ	Название	Описание
	Соединитель	Выход в часть схемы и вход из другой части схемы и используется для обрыва линии и продолжения ее в другом месте
	Терминатор	Отображает выход во внешнюю среду и вход из внешней среды (начало или конец схемы программы, внешнее использование и источник или пункт назначения данных)
	Комментарий	Символ используют для добавления описательных комментариев или пояснительных записей в целях объяснения или примечаний. Пунктирные линии в символе комментария связаны с соответствующим символом или могут обводить группу символов. Текст комментариев или примечаний должен быть помещен около ограничивающей фигуры
	Пропуск	Символ (три точки) используют в схемах для отображения пропуска символа или группы символов, в которых не определены ни тип, ни число символов. Символ пропуска используют только в символах линий или между ними. Он применяется главным образом в схемах, изображающих общие решения с неизвестным числом повторений.

Степень детализации алгоритма зависит от опыта разработчика и языка программирования, на котором будет реализован алгоритм. При разработке алгоритма необходимо разделить его на логически законченные части и оформлять их в виде подпрограмм (функций).

Пример. Разработать алгоритм обработки целочисленного массива.
Условия:

- 1) ввести с клавиатуры 10-ти элементный целочисленный массив;
- 2) вычислить сумму значений элементов массива;
- 3) определить максимальный и минимальный элемент массива и записать их в файл.

Вариант декомпозиции (разделения на части) данной задачи может быть представлен набором четырёх функций, которые выполняют следующие операции:

- ввод массива с клавиатуры;
- вычисление суммы значений элементов;
- поиск максимального и минимального числа;
- запись максимального и минимального значения в файл.

В результате получился обобщенный алгоритм решения задачи (рисунок 1).



Рисунок 1

Затем каждая функция раскрывается детальным алгоритмом. Например, функция вычисления суммы значений элементов массива может быть представлена алгоритмом, изображенным на рисунке 2.

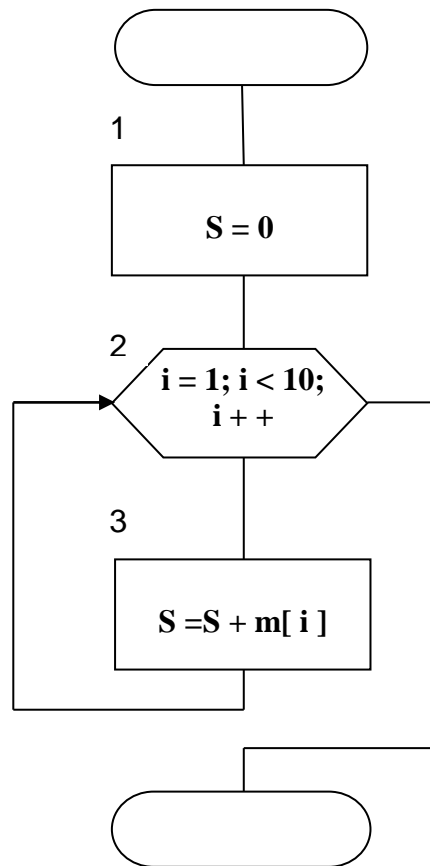


Рисунок 2

Чтобы облегчить переход от алгоритма к программе и обратно, желательно использовать в схеме алгоритма только те обозначения, которые допустимы в программе.

Этап составления программы. Программа составляется путем записи алгоритма решения задачи на языке программирования высокого уровня. При этом необходимо стремиться, чтобы программа получалась простая, наглядная и читаемая. Чем проще написана программа, чем лучше она оформлена, тем удобнее такая программа в изучении и эксплуатации. Удобочитаемость программы связана со стилем программирования. Поясним основные требования хорошего стиля программирования.

Первое требование. Программа должна содержать комментарии. Комментарии нужны как на стадиях проектирования и отладки программы, так и позже. Можно выделить комментарии вводные и пояснительные. Вводные комментарии должны помещаться в начале каждой программы или подпрограммы (функции) и пояснять их назначение. Пояснительные комментарии сопровождают наиболее трудно описываемые части программы. Средней нормой можно считать одну строку комментариев на десять строк программы.

Второе требование хорошего стиля программирования заключается в пропуске строк, что значительно повышает наглядность программы. Как в естественном языке мы пользуемся пропусками строк для отделения разделов, так и в программе с помощью пропуска строк можно разделять ее отдельные фрагменты. Например, пропуском одной строки можно отделять группу логически связанных операторов, пропуском двух строк – основные логически связанные фрагменты программы. Использование незаполненных строк облегчает поиск отдельных частей в программе.

Третье требование заключается в рациональном выборе имен переменных так, чтобы они наилучшим образом характеризовали те величины, которые они представляют. Все обозначения объектов в алгоритме и программе должны по возможности совпадать.

Четвертое требование касается размещения операторов. В большинстве языков программирования высокого уровня допускается размещение нескольких операторов на одной строке. Однако запись нескольких операторов на одной строке затрудняет чтение программы. Кроме того, по сообщению транслятора о синтаксической ошибке легче обнаружить эту ошибку при размещении в одной строке одного оператора.

Пятое требование. Для указания связи операторов, между ними делают одинаковые отступы от начала строки. Отступы, не оказывая влияния на логику программы, существенно улучшают ее читаемость. Правильно сделанные отступы хорошо подчеркивают структуру программы.

Далее приведён пример составления программы заполнения пятиэлементного целочисленного массива с клавиатуры и вывода на экран суммы его элементов.

// Программа вычисления суммы элементов массива

```
#include <iostream>
using namespace std;

int main()
{
    int m[5];
    int s,i;
    s=0;

    for(i=0; i<5; i++)    // Цикл ввода значений элементов массива
    {
        cout << "Insert number: ";
        cin >> m[i];
        s=s+m[i];        // Вычисление суммы
    }
}
```

```
    }  
  
    cout<<"Summa = " << s;  
  
    return 0;  
}
```

Этап тестирования и отладки программы. Этот этап является наиболее сложным этапом, так как общих и одновременно эффективных методов поиска ошибок не существует.

Все ошибки, которые возникают в процессе работы можно разделить на две большие группы: алгоритмические и программные. *Алгоритмические* ошибки приводят к неправильному или неэффективному решению задачи и обусловлены либо некорректной постановкой задачи, либо непониманием ее сути, либо неправильным выбором или применением математического аппарата. *Программные* ошибки возникают из-за недостаточного знания программистом языка программирования. Кроме алгоритмических и программных ошибок могут также возникать ошибки, вызываемые невнимательными действиями программиста в процессе набора программы.

При тестировании и отладке программы выполняются следующие виды работ:

а) Подготовка контрольных вариантов для проверки правильности решения поставленной задачи по программе. Контрольные варианты представляют собой наборы исходных данных и соответствующие им результаты решения задачи для наиболее характерных частных случаев выполнения программы. Контрольные варианты обычно рассчитывают вручную.

б) Планирование процесса отладки. Планирование состоит в выборе режима трансляции в интегрированной среде программирования, в определении контрольных точек для проверки ошибочных ситуаций, в формировании списка тех переменных, значения которых необходимо контролировать в этих точках в процессе выполнения программы или при других отладочных действиях.

в) Трансляция и исправление обнаруженных транслятором синтаксических ошибок. Исправление выявленных транслятором синтаксических ошибок представляет собой сложный этап отладки программы. Следует отметить, что на этапе трансляции выявляются не все ошибки. Ошибки логического характера транслятор обнаружить не может.

Из-за большого разнообразия синтаксических ошибок невозможно дать конкретные рекомендации по поводу способов их устранения. Однако можно указать следующий подход. Если в программе выявлено очень большое количество ошибок, следует просто исправить понятные и

очевидные ошибки и повторить трансляцию. Некоторые сообщения о синтаксических ошибках являются на самом деле ложными, так как вызваны другими ошибками. Поэтому не стоит тратить много времени на то, чтобы разобраться с каждым выданным сообщением, если характер ошибки не очевиден сразу. Ложных сообщений об ошибках особенно много тогда, когда в программе имеет место какое-то серьезное упущение, например, неправильное написание начала главной функции, отсутствие описания массива и так далее.

Так как вышерасположенные в программе ошибки влияют на синтаксис следующего за ними кода, их и надо исправлять в первую очередь. При этом после исправления каждой ошибки повторять трансляцию кода.

г) Выполнение программы и исправление логических ошибок. Программист приступает к этому виду работ только после устранения всех ошибок, обнаруженных на предыдущем этапе отладки программы. Выполнение программы производят для варианта, разработанного ранее в качестве контрольного. Если при этом получаются правильные результаты, то следует переходить к проверке программы на других вариантах. В случае если результаты не верны, производится обнаружение и исправление логических ошибок и повторная проверка программы по контрольным вариантам. После совпадения результатов выполнения программы со всеми контрольными вариантами отладка считается законченной.

Этап анализа полученных результатов. Это последний этап, в рамках которого производится использование уже разработанной программы для получения искомых результатов, осуществляется анализ результатов решения задачи и в случае необходимости - уточнение математической модели (с последующей корректировкой алгоритма и программы). Программы, имеющие большое практическое значение, используются длительное время. Иногда даже в процессе эксплуатации программы могут исправляться, дорабатываться.

2 СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

Пояснительная записка по своему содержанию должна соответствовать заданию на курсовую работу. В структурном плане пояснительная записка должна включать в себя следующие части:

- титульный лист;
- аннотацию;
- содержание (оглавление);
- основную часть;
- заключение;
- список использованной литературы;
- приложения.

Титульный лист выполняется стандартным образом.

В **аннотации** кратко излагается содержание проделанной работы и краткие выводы по полученным результатам. Объём аннотации - до одной страницы. Аннотация оформляется по правилам основной части, описание которой будет приведено ниже.

Содержание (оглавление) пояснительной записки включает номера и наименования разделов и подразделов основной части пояснительной записки с указанием номера листа, на котором размещается их начало.

Основная часть содержит материал, подлежащий обязательному представлению, т.е. последовательно излагает содержание работы, выполненной в соответствии с заданием:

- 1) Условие задачи.
- 2) Способ решения (формализация) задачи.
- 3) Схема алгоритма.
- 4) Текст программы.

Заключение должно содержать конкретные выводы по результатам работы. Примерный перечень пунктов выводов имеет следующий вид:

– какая решена задача (учебная, техническая, тактическая или какая-то другая);

– вид задачи (вычислительная или поисковая) и какой метод её решения выбран;

– особенности алгоритма решения задачи (структура укрупнённой схемы алгоритма, какие блоки и почему детализированы, как реализована многократность решения задачи т.д.);

– особенности программы решения задачи (как осуществлена проверка вводимых данных, какие операторы циклов и ветвлений использованы и почему, как реализован дружественный интерфейс пользователя, как применялись комментарии);

– какие разделы пояснительной записки оказались сложными и почему;

- техническая характеристика разработанной программы (язык программирования, версия, среда программирования, объём кода и т.д.);
- в какой области вашей профессиональной деятельности может возникнуть аналогичная задача.

Список использованной литературы. В пояснительной записке должны быть сделаны ссылки на использованную литературу. Список использованной литературы должен содержать только те источники, которые использовались при разработке курсовой работы и на которые есть ссылки в тексте пояснительной записки.

Приложения. Материал, дополняющий текст документа, допускается помещать в приложениях. Приложениями могут быть, например, экранные формы интерфейсов разработанной программы, графический материал, расчёты, образцы оформления, таблицы и т.д.

3 ОФОРМЛЕНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

3.1 Общие требования

Изложение текста и оформление пояснительной записки курсовой работы выполняют в соответствии с требованиями [1]. Страницы текста и включенные в пояснительную записку иллюстрации и таблицы должны соответствовать формату А4. Текст печатается на одной стороне листа белой бумаги через полтора интервала. Цвет шрифта - черный. Размер шрифта (кегель) - не менее 12 пт. Обычная практика - кегель 14 пт в основном тексте. В [1] не определяется тип шрифта, но обычно - Times New Roman.

Текст следует печатать без автоматических переносов, соблюдая следующие размеры полей: правое - не менее 10 мм, верхнее и нижнее - не менее 20 мм, левое - не менее 30 мм. Межстрочный интервал – полуторный; абзацный отступ (табуляция) – 1,25 см, выравнивание – по ширине.

Страницы работы нумеруются арабскими цифрами (нумерация сквозная по всему тексту). Номер страницы ставится в центре нижней части листа без точки. Титульный лист и аннотация или реферат включаются в общую нумерацию, но номера на них не ставятся. Иллюстрации и таблицы, расположенные на отдельных листах, включают в общую нумерацию страниц.

Качество напечатанного текста и оформления иллюстраций, таблиц должно удовлетворять требованию их четкого воспроизведения.

Опечатки, описки и графические неточности, обнаруженные в процессе подготовки пояснительной записки, допускается исправлять подчисткой или закрашиванием белой краской и нанесением на том же месте исправленного текста черными чернилами, пастой или тушью - рукописным способом.

Фамилии, названия организаций, фирм, названия программных продуктов должны приводиться на языке оригинала. Допускается транслитерировать имена собственные и приводить названия организаций в переводе на русский язык с добавлением (при первом упоминании) оригинального названия.

3.2 Заголовки

К структурным элементам пояснительной записки относятся "Аннотация", "Содержание", "Заключение", "Список использованных источников", "Приложение". Заголовки структурных элементов располагают в середине строки без точки в конце и печатают прописными (заглавными) буквами, не подчеркивая. Каждый структурный элемент следует начинать с новой страницы.

Основную часть отчета пояснительной записки курсовой работы делят на разделы, подразделы и пункты. Пункты, при необходимости, могут делиться на подпункты.

Разделы, подразделы, пункты и подпункты нумеруются арабскими цифрами и записываются с абзацного отступа. Их обычно нумеруют в пределах всего текста, кроме приложений, следующим образом: разделы – 1, 2, 3 и т.д., подразделы – 1.1, 1.2, 1.3 и т.д., пункты – 1.1.1, 1.1.2, 1.1.3, подпункты – 1.1.1.1, 1.1.1.2, 1.1.1.3 и т.д. После номера раздела, подраздела, пункта и подпункта в тексте точку не ставят.

Разделы и подразделы должны иметь заголовки. Пункты, как правило, заголовков не имеют. Заголовки должны четко и кратко отражать содержание разделов, подразделов. Заголовки разделов и подразделов следует печатать с абзацного отступа с прописной буквы без точки в конце, не подчеркивая.

Пример

1 Математическое моделирование

1.1 Постановка задачи

Если заголовок состоит из двух предложений, их разделяют точкой. Переносы слов в заголовках не допускаются.

Расстояние между заголовком и текстом должно быть равно 3 или 4 интервалам (15 мм). Если пояснительная записка напечатана интервалом 1,5, то это значит, что расстояние между заголовком и текстом равно одной пустой строке. Расстояние между заголовками раздела и подраздела - 2 интервала (8 мм).

3.3 Содержание

Заголовок СОДЕРЖАНИЕ пишется заглавными буквами посередине строки.

Содержание включает введение (если оно есть), наименование всех разделов, подразделов, пунктов, заключение, список использованных источников и наименование приложений с указанием номеров страниц, с которых начинаются эти элементы пояснительной записки.

3.4 Иллюстрации

На все иллюстрации (чертежи, графики, схемы, компьютерные распечатки, диаграммы, фотоснимки) в тексте должны быть даны ссылки. Иллюстрации должны располагаться непосредственно после текста, в котором они упоминаются впервые, или на следующей странице.

Иллюстрации, за исключением иллюстрации приложений, нумеруются арабскими цифрами, при этом нумерация сквозная, но допускается нумеровать и в пределах раздела. В последнем случае номер рисунка состоит из номера раздела и порядкового номера иллюстрации, разделенных точкой (например: Рисунок 1.1).

Подпись к иллюстрации располагается под ней посередине строки. Слово «Рисунок» пишется полностью. Можно ограничиться только номером (т.е. оставить, например, подпись: Рисунок 2), но при необходимости, иллюстрации могут иметь наименование и пояснительные данные (подрисуночный текст). В этом случае подпись должна выглядеть так: Рисунок 2 – Алгоритм подпрограммы. Точка в конце подрисуночного текста не ставится.

Если в работе есть приложения, то иллюстрации каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением впереди обозначения приложения (например: Рисунок А.3).

При ссылках на иллюстрации следует писать "... в соответствии с рисунком 2" при сквозной нумерации и "... в соответствии с рисунком 1.2" при нумерации в пределах раздела

3.5 Таблицы

Таблицы применяют для лучшей наглядности и удобства сравнения показателей. На все таблицы в тексте должны быть ссылки. При ссылке следует писать слово "таблица" с указанием ее номера. Таблица должна располагаться непосредственно после текста, в котором она упоминается впервые, или на следующей странице.

Все таблицы нумеруются (нумерация сквозная, либо в пределах раздела - в последнем случае номер таблицы состоит из номера раздела и порядкового номера внутри раздела, разделенных точкой (например: Таблица 1.2). Таблицы каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением впереди обозначения приложения (например: Таблица В.2). Слово «Таблица» пишется полностью.

Если в пояснительной записке одна таблица, то она должна быть обозначена "Таблица 1" или "Таблица В.1", если она приведена в приложении В.

Наличие у таблицы собственного названия не обязательно, но при необходимости, название таблицы следует помещать над таблицей слева, без абзачного отступа в одну строку с ее номером через тире (например: Таблица 3 - Доходы фирмы). Точка в конце названия не ставится. Название таблицы, при его наличии, должно отражать ее содержание, быть точным, кратким.

Таблицу с большим числом строк допускается переносить на другой лист (страницу). При переносе части таблицы на другой лист (страницу) слово "Таблица", ее номер и наименование указывают один раз слева над первой частью таблицы, а над другими частями также слева пишут слова "Продолжение таблицы" и указывают номер таблицы.

Таблицу с большим количеством столбцов допускается делить на части и помещать одну часть под другой в пределах одной страницы.

Если строки и столбцы таблицы выходят за формат страницы, то в первом случае в каждой части таблицы повторяется головка, во втором случае - боковик.

При делении таблицы на части допускается ее головку или боковик заменять соответственно номером столбцов и строк. При этом нумеруют арабскими цифрами столбцы и (или) строки первой части таблицы.

Заголовки столбцов и строк таблицы следует писать с прописной буквы в единственном числе, а подзаголовки столбцов - со строчной буквы, если они составляют одно предложение с заголовком, или с прописной буквы, если они имеют самостоятельное значение. В конце заголовков и подзаголовков столбцов и строк таблиц точки не ставят.

Заголовки столбцов, как правило, записывают параллельно строкам таблицы, но при необходимости допускается их перпендикулярное расположение.

Таблицы слева, справа и снизу, как правило, ограничивают линиями. Допускается применять размер шрифта в таблице меньший, чем в тексте.

Разделять заголовки и подзаголовки боковика и столбцов диагональными линиями не допускается.

Горизонтальные и вертикальные линии, разграничивающие строки таблицы, допускается не проводить, если их отсутствие не затрудняет пользование таблицей. Но головка таблицы должна быть отделена линией от остальной части таблицы.

Цифровой материал, как правило, оформляют в виде таблиц. Если цифровые или иные данные в какой-либо строке таблицы не приводят, то в ней ставят прочерк.

3.6 Формулы и уравнения

Формулы и уравнения следует выделять из текста в отдельную строку. Над и под каждой формулой или уравнением нужно оставить по пустой строке. Если уравнение не умещается в одну строку, то оно должно быть перенесено после знака равенства (=) или после знаков плюс (+), минус (-), умножения (x), деления (:), или других математических знаков, причем этот знак в начале следующей строки повторяют.

Если нужны пояснения к символам и коэффициентам, то они приводятся сразу под формулой или уравнением в той же последовательности, в которой они идут в формуле (уравнении).

Все формулы (уравнения) нумеруются. Обычно нумерация сквозная. Номер проставляется арабскими цифрами в круглых скобках в крайнем правом положении на строке.

Пример

$$A = a : (b + c)$$

(1)

$$B = c : (b - c)$$

(2)

Допускается нумерация формул и уравнений в пределах раздела. В этом случае номер формулы (уравнения) состоит из номера раздела и порядкового номера внутри раздела, разделенных точкой, например: (1.4).

Если в тексте пояснительной записки только одна формула, то ее обозначают (1).

Формулы и уравнения в приложениях имеют отдельную нумерацию в пределах каждого приложения с добавлением впереди обозначения приложения, например: (В.2).

Ссылки в тексте на порядковые номера формул дают в скобках, например: в формуле (1).

3.7 Перечисления

Перед каждым перечислением следует ставить дефис или, при необходимости ссылки в тексте на одно из перечислений, строчную букву (за исключением ё, з, й, о, ч, ь, ы, ь).

Для дальнейшей детализации перечислений необходимо использовать арабские цифры, после которых ставится скобка, а запись производится с абзацного отступа.

- а) _____
- б) _____
 - 1) _____
 - 2) _____
- в) _____

3.8 Схемы алгоритмов

Схема алгоритма состоит из символов, краткого пояснительного текста и соединяющих линий.

Символы предназначены для графического обозначения отдельных операций, суть которых выражается текстом внутри символов. Символы должны быть по возможности одного размера и располагаться в схеме равномерно, в любой ориентации, но предпочтительным является их горизонтальное расположение.

Внутри символа помещается минимальное количество текста, необходимого для понимания функции данного символа. Для пояснения работы алгоритма следует использовать символ "комментарий". Пунктирная линия символа "комментарий" связывается с соответствующим символом.

Символы в схеме соединяются линиями, которые указывают потоки управления. Линии должны быть направлены к центру символа. Следует избегать пересечения линий, если потоки в данном месте не входят друг в друга. При необходимости линии в схемах следует разрывать во избежание лишних пересечений или слишком длинных линий, а также, если схема состоит из нескольких страниц. Соединитель в начале разрыва является

внешним, а в конце разрыва - внутренним. В комментариях к соединителям могут быть приведены ссылки к страницам (рисунок 3)

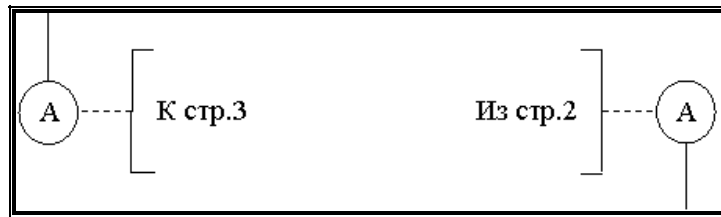


Рисунок 3

Символ "решение" является *логическим*. Каждый выход из символа "решение" должен сопровождаться значением условия, приведенного внутри (рисунок 4).

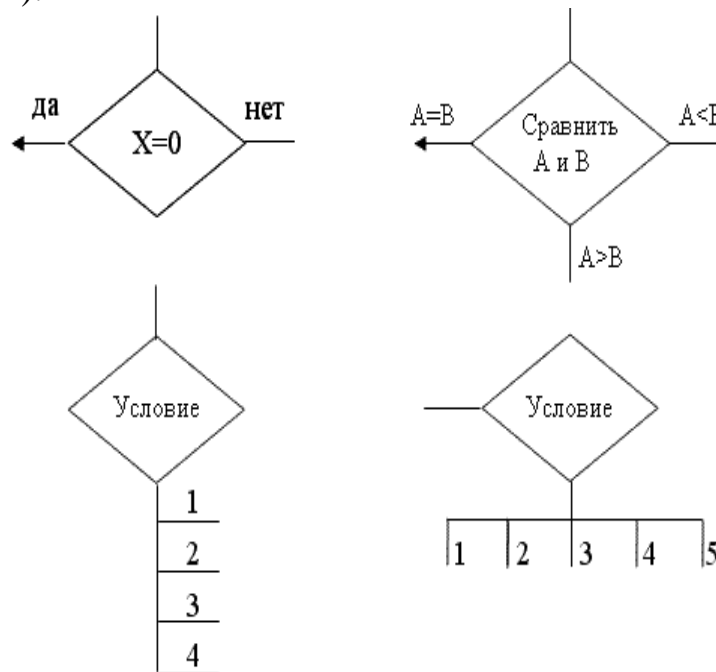


Рисунок 4

Представление алгоритма решения задачи в виде схемы является наиболее наглядным, позволяет проследить процесс прохождения данных, связи между отдельными участками программы. Однако, схема должна быть удобочитаемой, т.е. не должна быть чересчур мелкой, подробной, "перегруженной", чтобы не потерять своей наглядности.

В случае описания решения очень большой, сложной задачи рекомендуется выполнять схему с несколькими *уровнями детализации*, число которых зависит от размеров и сложности задачи. Уровень детализации должен быть таким, чтобы различные части и взаимосвязь между ними были понятны в целом. При этом весь алгоритм разбивается на смысловые фрагменты, связь между которыми следует указать на более

крупной схеме. Сами же фрагменты удобнее выполнять на отдельных страницах, желательно каждый фрагмент размещать на одной странице, не перегружая чрезмерно ссылками и комментариями.

Расстояние между параллельными линиями потоков должно быть не менее 3 мм, между остальными элементами схемы – не менее 5 мм.

Microsoft Word позволяет создавать блок-схему. Следует перейти на вкладку «Вставка» и в блоке «Иллюстрации» выбрать строку «Фигуры». Открывается меню всевозможно доступных элементов блок-схемы (рисунок 5).

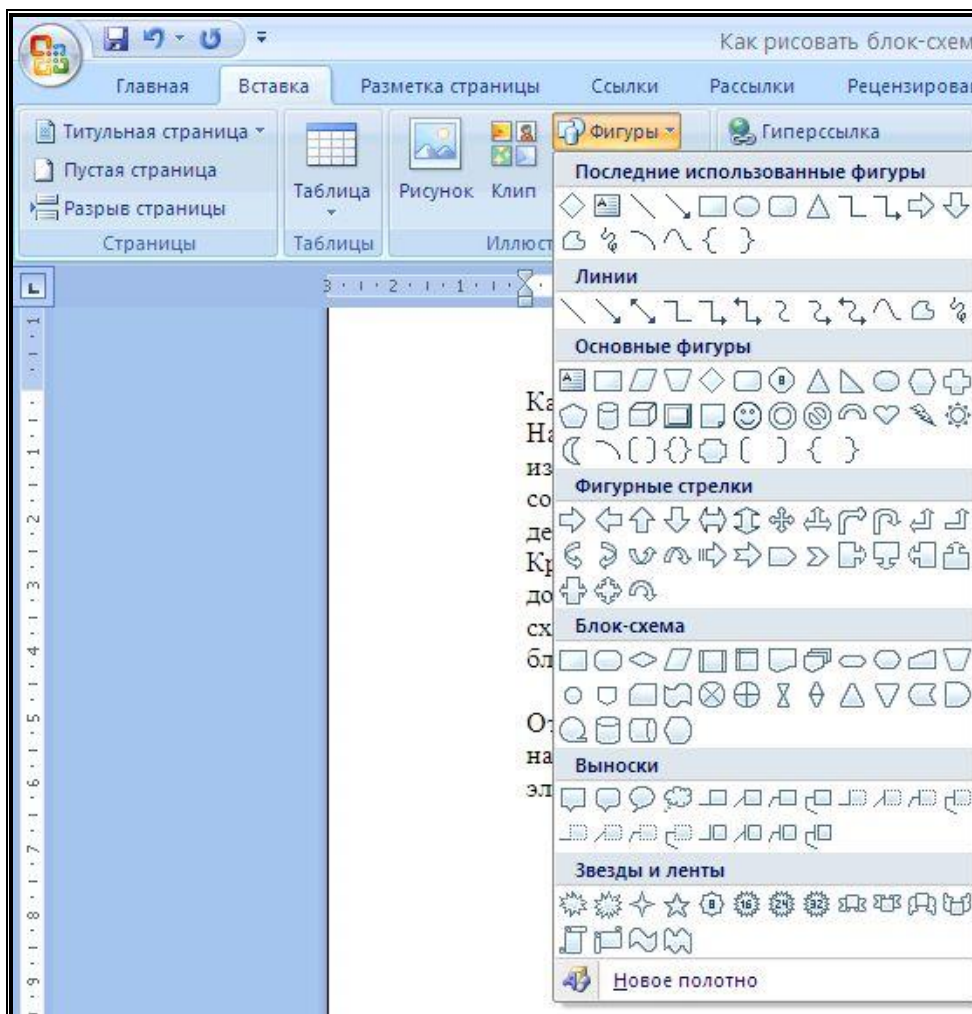


Рисунок 5

3.9 Список использованных источников

В пояснительной записке должны быть сделаны ссылки на использованную литературу. Ссылаться следует на документ в целом или его разделы и приложения. Ссылки на подразделы, пункты, таблицы и иллюстрации не допускаются, за исключением подразделов, таблиц и иллюстраций данного документа. Ссылки на использованные источники следует приводить в квадратных скобках. Ссылки на номер и страницу

источника также заключается в квадратные скобки, например: [15, с.124 - 128].

Список литературы должен называться «Список использованных источников», и заголовок данного структурного элемента печатается прописными буквами в центре строки.

Сведения об использованных в пояснительной записке источниках следует располагать в порядке появления ссылок на источники в тексте и нумеровать арабскими цифрами без точки и печатать с абзацного отступа. При этом впереди обычно идут нормативные акты.

Независимо от вида носителя, на котором размещён документ, в его библиографическую запись включается следующий перечень полей.

3.9.1 Область заглавия и сведений об ответственности

Эта область включает заголовок библиографической записи, который представляет собой фамилию первого автора с инициалами или их расшифровкой. Если авторов более трёх, то книгу следует описывать под её заглавием.

После заголовка следует заглавие, которое приводится без сокращения.

Элемент «Общее обозначение материала» используют в тех случаях, когда из имеющихся элементов записи не видна характеристика физического носителя документа (например, описание базы данных, расположенной на локальном компьютере и др.). Элемент указывают в квадратных скобках сразу после заглавия (*например: [Электронный ресурс], [Аудиозапись] и т.д.*). Если из других элементов библиографической записи понятно, о каком физическом носителе идет речь, допустимо опустить этот элемент.

Сведения, относящиеся к заглавию приводятся после двоеточия и представляют собой сведения, дополняющие или разъясняющие заглавие и расположенные на титульном листе документа.

Сведения об ответственности приводятся после косой черты и включают в себя фамилии авторов, указанных на титульном листе книги, начиная с первого и до четвертого, если больше четырех – первые три и слова «и др.»; сведения о редакторах, художниках, переводчиках и т.д. с указанием их роли.

3.9.2 Область издания

Сведения об издании приводятся в том случае, когда присутствует переиздание. *Например: 3-е изд., доп. и перераб.* Для электронного ресурса характерен выпуск версии издания, о чём и дается информация. *Например: – версия 5.61.*

3.9.3 Область выходных данных

Элемент «Выходные сведения» присутствует тогда, когда описывается документ целиком. В этом случае указываются место издания, издательство, год издания (*например: М. : Проспект, 2013*). При указании

места издания указывается название города полностью - Кострома, сокращаются только Москва (М.) и Санкт-Петербург (СПб.).

3.9.4 Область физической характеристики

Количественные характеристики содержат для описания документа на бумажном носителе информацию о количестве страниц в издании.

В процессе написания пояснительной записки курсовой работы могут быть использованы статьи, помещенные в сборниках или статьи из журналов, а также электронные ресурсы.

Если материал помещен в журнале, то сначала указываются фамилия и инициалы автора материала, название материала, затем ставится знак «//», после которого помещаются сведения об издании, в котором помещен данный материал: указывается название журнала и все его «выходные данные», помещенные на оборотной стороне титульного листа (серия, год, номер и т.д.). Важно помнить о том, что в конце сведений о материале указывается нумерация страниц, на которых начинается и заканчивается публикация.

Среди электронных ресурсов выделяют два вида: имеющие физическую основу (CD-ROM; DVD и т.д.) и не имеющие физической основы, то есть доступные удалённо (сайты, базы данных по подписке и т.д.).

В ситуации с электронным ресурсом основным источником информации является экранная титульная страница сайта, а также весь ресурс в целом.

Для электронных ресурсов, имеющих физическую основу (CD-ROM; DVD и т.д.) в качестве титульного листа выступает этикетка компакт-диска.

Для удалённых электронных ресурсов приводятся сведения о режиме доступа. В этом случае требуется указать адрес (URL) и условия доступа (чаще рекомендуется указывать ограничения доступа). Например: . – Режим доступа: <http://www.spbbt.ru/> или же UPL: <http://www.nlr.ru/>. Также указываются сведения об истории удаленного ресурса - указывают дату последнего посещения ресурса. Эти сведения приводятся в скобках и не отделяются от адреса какими-либо знаками. Например: (22.03.2013).

Пример выполнения списка литературы имеет следующий вид:

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Павловская Т. А. С/С++. Программирование на языке высокого уровня. – СПб.: Питер, 2014. – 461 с.

2 Керниган Б. Язык программирования Си / Керниган Б., Ритчи Д. – СПб, 2012. – 352 с.

3 Верещагин, Н. К. Колмогоровская сложность и алгоритмическая случайность / Н. К. Верещагин, В. А. Успенский, А. Шень. – М. : МЦНМО, 2013. – 575 с.

4 Информатика. Базовый курс : учебник для вузов / Под ред. С. В. Симоновича – Изд. 3-е, перераб. и доп. – СПб.: Питер, 2011. – 640 с.

5 Никитин, Д. Программные способы оптимизации алгоритмов цифровой обработки сигналов / Д. Никитин // Схемотехника. - 2006. - №3. - С. 24-27.

6 Волконский, В. Ю. Безопасная реализация языков программирования на базе аппаратной и системной поддержки / В. Ю. Волконский // Вопросы радиоэлектроники. Сер. Электронная вычислительная техника . – 2008. – №2. – С. 98-140.

7 Справочник экономиста по start-up [Электронный ресурс]. – Электрон. прогр. дан. – М. : Профессиональное изд-во, 2013. – 1 электрон. опт. диск (CD-ROM).

8 Мангутова, С. Д. Библиографическое описание сетевых ресурсов при оформлении ссылок и списков [Электронный ресурс] / С. Д. Мангутова // Библиография. – 2005. – № 4. – URL: <http://vss.nlr.ru/mangutova2.php> (17.05.2013).

3.10 Приложения

Приложение оформляют как продолжение пояснительной записки на последующих ее листах.

В тексте пояснительной записки на все приложения должны быть даны ссылки. Приложения располагают в порядке ссылок на них в тексте.

Каждое приложение следует начинать с новой страницы с указанием сверху посередине страницы слова "Приложение", его обозначения.

Приложение должно иметь заголовок, который записывают симметрично относительно текста с прописной буквы отдельной строкой.

Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением букв Е, З, Й, О, Ч, Ъ, Ы, Ь. После слова "Приложение" следует буква, обозначающая его последовательность. Допускается обозначение приложений буквами латинского алфавита, за исключением букв I и O. В случае полного использования букв русского и латинского алфавитов допускается обозначать приложения арабскими цифрами.

Если в пояснительной записке одно приложение, оно обозначается "Приложение А".

Текст каждого приложения, при необходимости, может быть разделен на разделы, подразделы, пункты, подпункты, которые нумеруют в пределах каждого приложения. Перед номером ставится обозначение этого приложения.

Приложения должны иметь общую с остальной частью документа сквозную нумерацию страниц.

4 ЗАЩИТА КУРСОВОЙ РАБОТЫ

При выполнении курсовой работы необходимо помнить следующее:

- работа, не имеющая самостоятельного характера и представляющая собой механически списанные куски из различных источников, возвращается студенту для доработки, либо полного переделывания, при этом преподаватель вправе поставить студенту оценку “неудовлетворительно”;
- небрежно и безграмотно написанные работы, также возвращаются;
- курсовая работа должна быть представлена в срок, установленный для её приёма.

Защита курсовой работы проводится устно в форме доклада в случае, если работа допущена к защите, с целью формирования окончательной оценки за работу студента. На защиту представляется пояснительная записка курсовой работы и файл с программой.

Допускается выставление оценки по результатам выполненной курсовой работы без ее защиты на последнем занятии при условии, что работа выполнена досрочно и оформлена с высоким качеством.

При получении неудовлетворительной оценки студенты выполняют работу по новой теме или перерабатывают прежнюю в установленные сроки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 ГОСТ 7.32-2001. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления [Электронный ресурс]. – URL: <http://protect.gost.ru/document.aspx?control=7&id=130946> (21.10.2014).

2 ГОСТ Р 7.0.5-2008. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая ссылка. Общие требования и правила составления [Электронный ресурс]. – URL: <http://gostexpert.ru/gost/gost-7.0.5-2008> (21.10.2014).

3 ГОСТ 7.1-2003. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления [Электронный ресурс]. – URL: http://diss.rsl.ru/datadocs/doc_291wu.pdf (21.10.2014).

4 ГОСТ 7.80-2000. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Заголовок. Общие требования и правила составления [Электронный ресурс]. – URL: <http://protect.gost.ru/document.aspx?control=7&id=138322> (21.10.2014).

5 ГОСТ 7.82-2001 Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления [Электронный ресурс]. – URL: <http://protect.gost.ru/document.aspx?control=7&id=131137> (21.10.2014).

6 ГОСТ 19.701-90 (ИСО 5807-85) ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения [Электронный ресурс]. – URL: http://ohranatruda.ru/ot_biblio/normativ/data_normativ/7/7707/ (21.10.2014).

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Задание на курсовую работу включает нескольких пунктов. Пункты задания в программе разрабатываются в *рамках отдельных функций*. Интерфейс пользователя оформляется *в виде меню*.

Варианты заданий

Разработать блок-схему алгоритма программы и программу, которая выполняет заданные операции.

Вариант № 1

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива вещественных чисел размерностью $[4 \times 5]$ элементов и вывод его на экран. Вещественные элементы должны содержать не более 3-х знаков после запятой;

б) производит поиск *max* или *min* элемента в указанной строке матрицы и результат выдает на экран (предусмотреть возможность многократного поиска, по запросу пользователя);

в) по окончании работы в одном файле сохраняет матрицу до следующего сеанса работы, а результаты поиска (*max* или *min* элемент для запрашиваемых строк) – в другом файле.

Вариант № 2

Перечень операций:

а) ввод с клавиатуры или чтение из файла одномерного массива целых чисел размерностью в 15 элементов и вывод его на экран. Элементы массива в файле расположены в виде одного столбца;

б) производит поиск *max* или *min* элемента массива по запросу и результат выдает на экран. Предусмотреть возможность повторного поиска по запросу пользователя;

в) по окончании работы сохраняет массив в файле. Данные этого массива сохраняются в виде матрицы, размерностью $[3 \times 5]$ элементов с двумя пробелами между ними.

Вариант № 3

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива целых чисел размерностью $[4 \times 5]$ элементов и вывод его на экран. Элементы массива в файле могут быть расположены в виде одного столбца или одной строки;

б) многократно производит изменение матрицы путем замены указанного элемента на другой, вводимый с клавиатуры, и выводит измененную матрицу на экран;

в) по окончании работы сохраняет массив в файле. Данные этого массива сохраняются в виде матрицы, размерностью $[4 \times 5]$ элементов с двумя пробелами между ними.

Вариант № 4

Перечень операций:

а) ввод с клавиатуры или чтение из файла одномерного массива целых чисел размерностью в 25 элементов и вывод его на экран. Элементы массива в файле расположены произвольно: в виде одного столбца или одной строки;

б) производит расчет среднего арифметического элементов массива, изменяет указанный пользователем элемент и вновь производит расчет среднего арифметического элементов нового массива. Предусмотреть многократную возможность изменения элементов массива;

в) по окончании работы сохраняет результаты в файлах. В одном файле данные массива сохраняются в виде матрицы, размерностью $[5 \times 5]$ элементов с двумя пробелами между ними, а результаты вычисления среднего арифметического элементов записываются в другой файл.

Вариант № 5

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива целых чисел размерностью $[4 \times 5]$ элементов и вывод его на экран. Элементы массива в файле могут быть расположены в виде одного столбца или одной строки;

б) многократно, в зависимости от запроса, производит расчет среднего арифметического элементов указанного пользователем столбца;

в) по окончании работы сохраняет результаты в файлах. В одном файле данные массива сохраняются в виде матрицы, размерностью $[4 \times 5]$ элементов с двумя пробелами между ними, а результаты вычисления среднего арифметического элементов и номер соответствующего столбца записываются в другой файл.

Вариант № 6

Перечень операций:

а) ввод с клавиатуры или чтение из файла одномерного массива целых чисел размерностью в 15 элементов и вывод его на экран. Элементы массива в файле расположены в виде одной строки;

б) многократно, в зависимости от запроса, производит подсчет количества отрицательных, или положительных, или нулевых элементов массива и выдает результат на экран;

в) по окончании работы сохраняет результаты в файлах. В одном файле данные массива сохраняются в виде матрицы, размерностью $[3 \times 5]$ элементов с двумя пробелами между ними, а результаты количества отрицательных, положительных и нулевых элементов массива записываются в другой файл.

Вариант № 7

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива целых чисел размерностью $[4 \times 5]$ элементов и вывод его на экран в виде матрицы. Элементы массива в файле могут быть расположены в виде одного столбца или одной строки;

б) многократно, в зависимости от запроса, производит подсчет суммы элементов указанной строки и выдает результат на экран;

в) по окончании работы сохраняет результаты в файлах. В одном файле данные массива сохраняются в виде матрицы, размерностью $[4 \times 5]$ элементов с двумя пробелами между ними, а результаты подсчета суммы элементов указанной строки и ее номер записываются в другой файл.

Вариант № 8

Перечень операций:

а) ввод с клавиатуры или чтение из файла одномерного массива целых чисел размерностью в 15 элементов и вывод его на экран. Элементы массива в файле расположены в виде одного столбца;

б) производит расчет суммы элементов массива, изменяет указанный пользователем элемент и вновь производит расчет суммы элементов нового массива. Предусмотреть многократную возможность изменения элементов массива;

в) по окончании работы сохраняет новый (последний) массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[3 \times 5]$ элементов с двумя пробелами между ними, а результаты суммы элементов всех получаемых в результате выполнения п. б) массивов записываются в другой файл.

Вариант № 9

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива целых чисел размерностью $[3 \times 6]$ элементов и вывод его на экран в виде матрицы. Элементы массива в файле могут быть расположены произвольно;

б) многократно производит в зависимости от запроса поиск *max* или *min* элемента указанной строки матрицы и результат выдает на экран;

в) по окончании работы сохраняет массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[3 \times 6]$ элементов с

двумя пробелами между ними. Результаты поиска записываются во второй файл.

Вариант № 10

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива целых чисел размерностью $[3 \times 4]$ элементов и вывод его на экран в виде матрицы. Элементы массива в файле могут быть расположены произвольно;

б) многократно производит в зависимости от запроса поиск *max* или *min* элемента указанного столбца матрицы и результат выдает на экран;

в) по окончании работы сохраняет массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[3 \times 4]$ элементов с двумя пробелами между ними, а результаты поиска *max* или *min* элемента указанных столбцов матрицы и их номера записываются во второй файл.

Вариант № 11

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива целых чисел размерностью $[3 \times 6]$ элементов и вывод его на экран в виде матрицы. Элементы массива в файле могут быть расположены произвольно;

б) многократно производит изменение матрицы путем замены указанного элемента на другой, вводимый с клавиатуры, и выводит измененную матрицу на экран;

в) по окончании работы сохраняет последний измененный массив в файле. Данные этого массива сохраняются в виде матрицы, размерностью $[3 \times 6]$ элементов с двумя пробелами между ними.

Вариант № 12

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива целых чисел размерностью $[4 \times 6]$ элементов и вывод его на экран в виде матрицы. Элементы массива в файле могут быть расположены произвольно;

б) многократно, в зависимости от запроса, производит расчет среднего арифметического элементов указанной строки;

в) по окончании работы сохраняет массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[4 \times 6]$ элементов с двумя пробелами между ними, а результаты расчета среднего арифметического элементов указанных строк матрицы и их номера записываются во второй файл.

Вариант № 13

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива вещественных чисел размерностью $[3 \times 6]$ элементов и вывод его на экран. Вещественные числа должны содержать не более 5-и знаков после запятой;

б) многократно, в зависимости от запроса, производит расчет среднего арифметического элементов указанного столбца;

в) по окончании работы сохраняет массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[3 \times 6]$ элементов с двумя пробелами между ними. Результаты расчета среднего арифметического элементов указанных столбцов матрицы и их номера записываются во второй файл.

Вариант № 14

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива вещественных чисел размерностью $[3 \times 6]$ элементов и вывод его на экран. Вещественные числа должны содержать не более 5-и знаков после запятой;

б) многократно, в зависимости от запроса, производит подсчет количества отрицательных или положительных, или нулевых элементов матрицы и выдает результат на экран;

в) по окончании работы сохраняет массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[3 \times 6]$ элементов с двумя пробелами между ними. Число отрицательных, положительных и нулевых элементов массива записываются во второй файл.

Вариант № 15

Перечень операций:

а) ввод с клавиатуры или чтение из файла , имя которого задает пользователь, двумерного массива целых чисел размерностью $[3 \times 6]$ элементов и вывод его на экран;

б) многократно, в зависимости от запроса, производит подсчет суммы элементов указанной строки и выдает результат на экран;

в) по окончании работы сохраняет матрицу в файле, имя которого задает пользователь, до следующего сеанса работы.

Вариант № 16

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива целых чисел размерностью $[3 \times 6]$ элементов и вывод его на экран в виде матрицы. Элементы массива в файле могут быть расположены произвольно;

б) многократно, в зависимости от запроса, производит подсчет суммы элементов указанного столбца и выдает результат на экран;

в) по окончании работы сохраняет массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[3 \times 6]$ элементов с

двумя пробелами между ними. Результаты расчета суммы элементов указанных столбцов матрицы и их номера записываются во второй файл.

Вариант № 17

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива вещественных чисел размерностью $[4 \times 5]$ элементов и вывод его на экран в виде матрицы. При выводе все вещественные числа ограничены 4-мя знаками после запятой. Элементы массива в файле могут быть расположены произвольно;

б) многократно производит в зависимости от запроса поиск *max* или *min* элемента указанной строки матрицы и результат выдает на экран;

в) по окончании работы сохраняет массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[4 \times 5]$ элементов с двумя пробелами между ними. Результаты поиска *max* или *min* элемента указанных строк матрицы и их номера записываются во второй файл. В этом же файле сохраняется результат определения самого максимального и самого минимального из найденных элементов в указанных строках.

Вариант № 18

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива вещественных чисел размерностью $[4 \times 5]$ элементов и вывод его на экран в виде матрицы. При выводе все вещественные числа ограничены 4-мя знаками после запятой. Элементы массива в файле могут быть расположены произвольно;

б) многократно производит в зависимости от запроса поиск *max* или *min* элемента указанного столбца матрицы и результат выводит на экран;

в) по окончании работы сохраняет массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[4 \times 5]$ элементов с двумя пробелами между ними. Результаты поиска *max* или *min* элемента указанных столбцов матрицы и их номера записываются во второй файл. В этом же файле сохраняется результат определения самого максимального и самого минимального из найденных элементов в указанных столбцах.

Вариант № 19

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива целых чисел размерностью $[3 \times 4]$ элементов и вывод его на экран в виде матрицы. Элементы массива в файле могут быть расположены произвольно;

б) многократно производит изменение матрицы путем замены указанного элемента на другой, вводимый с клавиатуры, и выводит измененную матрицу на экран;

в) по окончании работы сохраняет измененный массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[3 \times 4]$ элементов с двумя пробелами между ними. Величины измененных (старых) элементов и значение их индексов $[i,j]$ записываются во второй файл.

Вариант № 20

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива целых чисел размерностью $[7 \times 2]$ элементов и вывод его на экран в виде матрицы. Элементы массива в файле могут быть расположены произвольно;

б) многократно, в зависимости от запроса, производит расчет среднего арифметического элементов указанной пользователем строки;

в) по окончании работы сохраняет массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[7 \times 2]$ элементов с двумя пробелами между ними. Результаты расчета среднего арифметического элементов указанных строк матрицы и их номера записываются во второй файл.

Вариант № 21

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива целых чисел размерностью $[3 \times 7]$ элементов и вывод его на экран в виде матрицы. Элементы массива в файле могут быть расположены произвольно;

б) многократно, в зависимости от запроса, производит расчет среднего арифметического элементов указанного пользователем столбца;

в) по окончании работы сохраняет массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[3 \times 7]$ элементов с двумя пробелами между ними. Результаты расчета среднего арифметического элементов указанных столбцов матрицы и их номера записываются во второй файл.

Вариант № 22

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива целых чисел размерностью $[3 \times 6]$ элементов и вывод его на экран в виде матрицы. Элементы массива в файле могут быть расположены произвольно;

б) многократно, в зависимости от запроса, производит подсчет количества отрицательных или положительных, или нулевых элементов матрицы и выдает результат на экран;

в) по окончании работы сохраняет массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[3 \times 6]$ элементов с двумя пробелами между ними. Результаты подсчета количества

отрицательных, положительных и нулевых элементов массива записываются во второй файл.

Вариант № 23

Перечень операций:

а) ввод с клавиатуры или чтение из файла, имя которого задает программист, двумерного массива целых чисел размерностью $[4 \times 7]$ элементов и вывод его на экран;

б) многократно, в зависимости от запроса, производит подсчет суммы элементов указанной строки и выдает результат на экран;

в) по окончании работы сохраняет массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[4 \times 7]$ элементов с тремя пробелами между ними. Номера указанных строк, значение всех их элементов и сумма элементов каждой строки последовательно записываются во второй файл.

Вариант № 24

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива целых чисел размерностью $[4 \times 6]$ элементов и вывод его на экран в виде матрицы. Элементы массива в файле могут быть расположены произвольно;

б) многократно, в зависимости от запроса, производит подсчет суммы элементов указанного столбца и выдает результат на экран;

в) по окончании работы сохраняет массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[4 \times 6]$ элементов с тремя пробелами между ними. Номера указанных столбцов, значения всех их элементов и сумма элементов каждого столбца последовательно записываются во второй файл.

Вариант № 25

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива вещественных чисел размерностью $[2 \times 6]$ элементов и вывод его на экран в виде матрицы. При выводе все вещественные числа ограничены 4-мя знаками после запятой. Элементы массива в файле могут быть расположены произвольно;

б) многократно, в зависимости от запроса, производит поиск *max* или *min* элемента указанной строки матрицы и результат выводит на экран;

в) по окончании работы сохраняет массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[2 \times 6]$ элементов с двумя пробелами между ними. Результаты поиска *max* или *min* элемента указанных строк матрицы и их номера записываются во второй файл. В этом

же файле сохраняется результат определения самого максимального и самого минимального из найденных элементов в указанных строках.

Вариант № 26

Перечень операций:

а) ввод с клавиатуры или чтение из файла двумерного массива вещественных чисел размерностью $[2 \times 6]$ элементов и вывод его на экран в виде матрицы. При выводе все вещественные числа ограничены 4-мя знаками после запятой. Элементы массива в файле могут быть расположены произвольно;

б) многократно, в зависимости от запроса, производит поиск *max* или *min* элемента указанного столбца матрицы и результат выдает на экран;

в) по окончании работы сохраняет массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[3 \times 6]$ элементов с двумя пробелами между ними. Результаты поиска *max* или *min* элемента указанных столбцов матрицы и их номера записываются во второй файл. В этом же файле сохраняется результат определения самого максимального и самого минимального из найденных элементов в указанных столбцах.

Вариант № 27

Перечень операций:

а) ввод с клавиатуры или чтение из файла, имя которого задает программист, двумерного целочисленного массива размерностью $[5 \times 6]$ элементов и вывод его на экран;

б) многократно производит изменение матрицы путем замены указанного элемента на другой, вводимый с клавиатуры, и выводит измененную матрицу на экран;

в) по окончании работы сохраняет последний измененный массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[2 \times 6]$ элементов с двумя пробелами между ними. Во втором файле сохраняются значения и индексы $[i,j]$ измененных элементов.

Вариант № 28

Перечень операций:

а) ввод с клавиатуры или чтение из файла одномерного массива вещественных чисел размерностью в 12 элементов и вывод его на экран. Вещественные числа при выводе должны содержать не более 4-х знаков после запятой;

б) многократно, в зависимости от запроса, производит расчет среднего арифметического только четных, или только нечетных элементов массива;

в) по окончании работы сохраняет массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[2 \times 6]$ элементов с

двумя пробелами между ними. Во втором файле сохраняются значения рассчитываемых элементов и результат их среднего арифметического.

Вариант № 29

Перечень операций:

а) ввод с клавиатуры или чтение из файла, имя которого задает программист, двумерного целочисленного массива размерностью $[3 \times 5]$ элементов и вывод его на экран. Данные в файле расположены произвольно;

б) многократно, в зависимости от запроса, производит расчет среднего арифметического элементов указанного столбца;

в) по окончании работы сохраняет массив в первом файле с именем, указанным программистом. Данные этого массива сохраняются в виде матрицы, размерностью $[3 \times 5]$ элементов с двумя пробелами между ними. Результаты расчета среднего арифметического элементов указанных столбцов матрицы и их номера записываются во второй файл.

Вариант № 30

Перечень операций:

а) ввод с клавиатуры или чтение из файла, имя которого задает программист, одномерного массива целых чисел размерностью в 14 элементов и вывод его на экран;

б) многократно, в зависимости от запроса, производит подсчет количества отрицательных или положительных, или нулевых элементов массива и выдает результат на экран;

в) по окончании работы сохраняет массив в первом файле. Данные этого массива сохраняются в виде матрицы, размерностью $[2 \times 7]$ элементов с двумя пробелами между ними. Количество отрицательных, положительных и нулевых элементов массива сохраняет в другом файле.

Пример титульного листа

Санкт-Петербургский государственный университет телекоммуникаций
им.проф. М.А.Бонч-Бруевича
Факультет информационных систем и технологий

Курсовая работа

по дисциплине «Алгоритмизация и программирование»

Выполнил
студ.1 курса, гр.ИСТ311
направление подготовки 230400
_____Иванов А.А.
«__»_____ 2015

Принял
ст.пр.каф.БИС

оценка
_____Евстигнеев В.А.
«__»_____ 2015

Санкт-Петербург, 2015

**Липанова Ирина Александровна
Медведев Валерий Александрович
Хорошенко Сергей Викторович**

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

**Учебно-методическое пособие
по выполнению курсовой работы**

Редактор

План 2014 г., п. 4

**Подписано к печати 12.09.2014
Объем 2,75 усл.-печ. л. Тираж 105 экз. Заказ 145**

Издательство СПбГУТ. 191186 СПб., наб. р. Мойки, 61