

МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ТУЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Институт прикладной математики и компьютерных наук

Кафедра информационной безопасности

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ**

по дисциплине

ПРОГРАММИРОВАНИЕ
(2 СЕМЕСТР)

Уровень профессионального образования: *высшее образование – бакалавриат*

Квалификация выпускника: *бакалавр*

Тула, 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. ЦЕЛЬ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ	4
2. ОСНОВНЫЕ ТРЕБОВАНИЯ К КУРСОВОЙ РАБОТЕ.....	5
2.1. Тематика курсовой работы.....	5
2.2. Исходные данные к курсовой работе	5
2.3. Задание на курсовую работу	5
2.4. Объем курсовой работы, стадии и этапы разработки.....	5
2.5. Работа над курсовой работой	6
2.6. Защита курсовой работы	6
3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К КУРСОВОЙ РАБОТЕ.....	6
3.1. Содержание разделов пояснительной записки	6
3.2. Рекомендации по алгоритмизации, программированию и отладке программ	8
3.3. Оформление пояснительной записки	9
ПРИЛОЖЕНИЕ 1	11
ПРИЛОЖЕНИЕ 2	12
ПРИЛОЖЕНИЕ 3	13
ПРИЛОЖЕНИЕ 4	14
ПРИЛОЖЕНИЕ 5	15
ПРИЛОЖЕНИЕ 6	16

ВВЕДЕНИЕ

Характерной особенностью научно-технического прогресса в нашей стране является широкое внедрение вычислительной техники во все отрасли народного хозяйства. Предполагается значительно расширить работы по созданию программного обеспечения (ПО) ЭВМ.

Выполнение курсовой работы должно способствовать закреплению, углублению и обобщению знаний, полученных студентами за время обучения, а также позволит приобрести опыт изучения вопросов алгоритмизации и программирования, усвоить конкретные методы и технологию программирования, привить навыки работы с технической документацией, научной и справочной литературой.

1. ЦЕЛЬ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ

Основная цель работы заключается в закреплении навыков самостоятельного использования теоретического материала и практического опыта для решения прикладных задач с использованием языка программирования C++.

2. ОСНОВНЫЕ ТРЕБОВАНИЯ К КУРСОВОЙ РАБОТЕ

Основными требованиями к курсовой работе являются описание подходов, методов, способов решения задач (согласно индивидуальному варианту), использование при разработке программ современных методов и приемов программирования, выполнение графика курсовой работы.

2.1. Тематика курсовой работы

Курсовая работа посвящена исследованию базовых конструкций языка программирования C++, методов и способов работы с данными, составлению алгоритмов для решения прикладных задач, а также разработке программ на языке программирования C++ согласно индивидуального задания.

2.2. Исходные данные к курсовой работе

Исходные данные определяются преподавателем при выдаче задания на курсовую работу.

Для очно-заочной и заочной формы, включая дистанционное обучение, обучения индивидуальные задания определяются по последним двум цифрам номера зачетной книжки (приложение 5).

В процессе работы они могут уточняться или изменяться преподавателем.

2.3. Задание на курсовую работу

Задание на курсовую работу выдается индивидуально каждому студенту.

Задание на курсовую работу оформляется студентом после постановки задачи преподавателем и имеет единую форму, приведенную в прил.1.

Задание подписывается студентом и руководителем и помещается за титульным листом в пояснительной записке. Задание является документом, на основании которого и выполняется курсовая работа.

2.4. Объем курсовой работы, стадии и этапы разработки

Курсовая работа включает 2 этапа:

1) описание (исследование) задач (согласно индивидуальному варианту), методов, способов, алгоритмов их решения (постановка задачи);

2) разработка и описание программ для решения задач (согласно индивидуальному заданию) предполагает выполнение всех основных этапов и стадий разработки программ, к которым относятся:

- определение общих требований к программе;
- описание входной и выходной информации;
- разработка алгоритма решения задачи;
- описание структуры программы для решения задачи и ее эксплуатации;
- описание контрольного примера.

Объем курсовой работы (пояснительной записки) не менее 20-30 листов формата А4.

2.5. Работа над курсовой работой

Перед началом выполнения работы заполняется бланк задания (Приложение 1), подписывается студентом и отдается преподавателю.

После получения задания на КР студент должен самостоятельно исследовать его:

- найти и изучить литературу по теме КР, изучить методы и способы реализации задания, определить область применения решения;
- сформулировать самостоятельно 2-3 возможных различных вариантов решения задачи с входными данными по теме индивидуального задания, решить ее, доказать правильность подхода и решения;

- запрограммировать решение на языке C++, отладить программу;

[Такая последовательность выполнения задачи по КР относится к каждому индивидуальному заданию по КР. Индивидуальных заданий на КР – два.]

- оформить пояснительную записку по выполнению КР.

Перед процедурой защиты КР необходимо получить рецензию (Приложение 4) у любого преподавателя Вашего института. К защите допускается работа, на которую дана рецензия.

При дистанционном обучении бланк задания загружаются с работой (пояснительная записка, файлы с программным кодом) вместе на проверку.

2.6. Защита курсовой работы

После выполнения всех этапов проектирования студент сдает руководителю на проверку пояснительную записку. Проверив представленные к защите пояснительные записки, руководитель назначает срок защиты курсовой работы. Затем пояснительная записка оценивается рецензентом. Защита происходит перед комиссией в присутствии студентов данной группы. Студент готовит доклад на 5-10 минут, в котором отражает основные результаты работы, доводит его до сведения комиссии, используя при этом пояснительную записку и необходимый графический материал. После защиты члены комиссии оценивают выполнение курсовой работы и руководитель выставляет отметку.

3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К КУРСОВОЙ РАБОТЕ

3.1. Содержание разделов пояснительной записки

Перечень разделов пояснительной записки приведен в прил.3.

Введение

Определяется цель курсовой работы, дается краткое содержание задач (согласно индивидуальному варианту).

Раздел. «Постановка задачи»

Приводится описание структур, моделей, методов, способов и т.п. решения задач (согласно индивидуальному заданию), их исследование.

Раздел. «Описание входной и выходной информации»

Приводятся исходные данные, определяются входные и выходные данные по каждой задаче.

Раздел. «Алгоритмы решения задач»

Приводятся алгоритмы (блок-схема алгоритма) решения каждой задачи. Дается их описание.

Раздел. «Общие требования к программе»

Указываются требуемые функциональные характеристики программ, обосновывается выбор языка программирования.

Раздел. «Описание структуры программы для решения задачи»

Указываются наименования и назначение программ, язык программирования, специальное программное обеспечение (при наличии), сведения о функциональных ограничениях на применение программ, структура программы с описанием функций подпрограмм и связей между ними.

Дается описание логической структуры программы с учетом текста (кода) программ на языке программирования C++, который приводится в приложении.

Описывается организация входных и выходных данных в программах, приводятся форматы и способы кодирования данных.

Раздел. «Инструкции по эксплуатации программ»

Указывается последовательность действий, обеспечивающих запуск, выполнение и завершение программ; приводится описание режимов функционирования программ; указывается набор сообщений, выдаваемых в ходе выполнения программы.

Раздел. «Описание контрольного примера»

Приводятся сведения о функционировании программ на примере текстового набора данных; описываются входные данные контрольного примера, последовательность работы с программами, приводятся полученные результаты.

ЗАКЛЮЧЕНИЕ

Делается вывод по курсовой работе (по каждому индивидуальному заданию), приводятся полученные результаты, проводится их сравнение.

Библиографический список

Приложения

Код программы.

3.2. Рекомендации по алгоритмизации, программированию и отладке программ

При разработке алгоритма следует пользоваться методом последовательной (поэтапной) детализации.

При этом на первом этапе составляется укрупненный алгоритм, затем выделяются его составные части, для выполнения которых используются стандартные процедуры. Алгоритм должен строиться по модульному принципу, который предполагает представление алгоритма в виде отдельных относительно независимых частей. При построении алгоритма следует пользоваться ограниченным числом типовых структур. Для представления схемы алгоритма рекомендуется применять графический способ описания в соответствии с ГОСТом 19.701-90. Полученный алгоритм должен обладать свойствами определенности, массовости, результативности, инвариантности.

При программировании задачи необходимо пользоваться принципами модульного программирования. Данная технология программирования заключается в разбиении задачи на определенное число программных модулей. При этом каждый модуль должен обладать свойствами функциональной законченности и минимальной связности. Такое представление программы позволяет вести параллельную разработку программы и легко устранять ошибки при отладке. При использовании неструктурированных языков программирования следует придерживаться методов структурного программирования, которые сводятся к использованию базовых типов структур (линейной, разветвляющейся, циклической).

Разработанная программа должна быть универсальной, т.е. допускать многократное выполнение и обеспечивать независимость от конкретных данных. При реализации ввода-вывода необходимо выбирать такое представление данных, которое бы обеспечивало максимальное удобство при подготовке данных, минимум возможных ошибок и удобочитаемость полученных результатов. При вводе необходимо всегда проверять достоверность данных и предусмотреть защиту от неправильных данных.

Текст программы должен легко восприниматься и быть удобочитаемым. Для этого необходимо использовать пробелы в операторах, пропуски и отступы строк, насыщать программу комментариями. Комментарии делятся на вводные и пояснительные. Вводные комментарии располагаются в начале программы и содержат краткие сведения о программе. Пояснительные комментарии сопровождают участки программы, которые трудно воспринимаются. При этом комментарии должны объяснять назначение группы операторов, а не описывать действия, выполняемые этими операторами. Средней нормой можно считать одну строку комментариев на 10 строк программы. Располагать комментарии в программе надо так, чтобы они были заметными. Для этого можно использовать пропуск строк, отделять комментарии заметными символами.

Отладка программы заключается в определении, локализации и устранении ошибок в программе. Отладку программы рекомендуется проводить с помощью специальных отладочных средств, которые входят в состав языка

программирования. Допускается также использовать собственные средства отладки, которые вводятся в программу при ее написании. К таким отладочным средствам относятся: отображение значений переменных, отслеживание хода выполнения программы и обращение к подпрограммам. Процесс отладки производится в два этапа - сначала проводят автономную отладку подпрограммы, а затем и комплексную отладку всей программы.

После того, как программа становится работоспособной, производится ее тестирование, задачей которого является проверка правильности функционирования программы во всем диапазоне допустимых значений исходных данных. Тестирование производится с помощью контрольного примера. При составлении контрольного примера необходимо предусмотреть, чтобы программа была апробирована во всех режимах ее функционирования.

3.3. Оформление пояснительной записки

Материал в пояснительной записке размещается в последовательности, указанной в приложении 2. Образец титульного листа приведен в приложении 3, а бланк задания в приложении 1. Текстовая часть пояснительной записки, а также схемы алгоритмов и программа, которые являются основными результатами выполнения курсовой работы, должны выполняться в строгом соответствии с требованиями ЕСПД. В приложении представляются листинг программы. Листинг программы должен быть разделен на страницы и сложен соответствующим образом. Пояснительная записка оформляется на сброшюрованных листах белой бумаги формата А4 в объеме 20-25 страниц.

Основной текст рекомендуется выравнивать по ширине страницы, шрифт Times New Roman высотой 14пт с интервалом 21пт между строками. *РАЗДЕЛЫ* должны иметь порядковые номера в пределах всего документа, обозначенные арабскими цифрами без точки и записанные с абзацного отступа. *ПОДРАЗДЕЛЫ* должны иметь нумерацию в пределах каждого раздела. Номер подраздела состоит из номеров раздела и подраздела, разделенных точкой. В конце номера подраздела точка не ставится.

Нумерация страниц сквозная и проставляется в правом нижнем углу страницы. Первой страницей является титульный лист, на котором номер страницы не ставится. Затем помещается бланк задания, который включается в нумерацию. Третьей страницей является содержание, которое нумеруется. Рисунки нумеруются последовательно арабскими цифрами и сопровождаются содержательными подрисуночными подписями. Номер и название рисунка располагается внизу по центру относительно самого рисунка. Пример оформления рисунка приведен на рис.1.

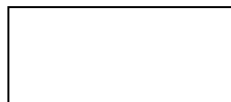


Рисунок 1- Схема программы

Рис.1. Пример оформления рисунка

На все рисунки должны быть ссылки в тексте, при этом указывается слово <рисунок> и указывается его номер. Рисунки помещаются после первого упоминания в тексте. Если рисунок размещается на нескольких листах, то на всех последующих листах пишется <Продолжение рисунка > и указывается его номер.

Цифровой материал, помещенный в записке, рекомендуется оформлять в виде таблицы. Таблицу помещают после первого упоминания в тексте. (Над правым верхним углом таблицы помещается надпись "Таблица" с указанием ее порядкового номера, затем, через тире, следует заголовок таблицы. При ссылке на таблицу в тексте пишется слово <таблица> полностью затем указывается ее номер. Пример оформления таблицы представлен на рис.2.

Таблица 1 – Структура исходных данных

Поле1	Поле2			ПолеN

Рис.2. Пример оформления таблицы

Формулы в записке должны быть пронумерованы. Номер формулы следует заключать в круглые скобки и помещать на правом поле страницы на уровне нижней строки формулы, к которой он относится. При ссылке на формулу необходимо указывать ее номер в скобках.

При оформлении текстовой части допускается сокращение слов, терминов, обозначений только общепринятых. Изложение должно быть кратким и четким.

ЗАПИСКА ПОДПИСЫВАЕТСЯ СТУДЕНТОМ И РУКОВОДИТЕЛЕМ.

ПРИЛОЖЕНИЕ 1

УТВЕРЖДАЮ

Дир. ИПМКН

_____ А.А.Сычугов

"__" _____ 20__ г.

ЗАДАНИЕ

на курсовую работу по программированию

студента гр. _____
(ФИО, группа)

ТЕМА: _____

(Название, номер варианта)

Исходные данные _____

Задание получил: _____
(ФИО, подпись)

Дата выдачи задания : _____

Задание выдал: _____
(ФИО, подпись)

Срок защиты курсовой работы: _____

Замечания консультанта: _____

К защите допущен. Консультант работы _____

"__" _____ 20__ г.

ПРИЛОЖЕНИЕ 2

МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ТУЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт прикладной математики и компьютерных наук
Кафедра информационной безопасности

Разработка программ по варианту №
(тема курсовой работы)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
по дисциплине

(полное наименование учебной дисциплины)

Студент гр.

(индекс группы)

(подпись и дата)

(инициалы и
фамилия)

Руководитель

доц. ИПМКН,
К.Т.Н.,
(должность и
ученая степень)

(подпись и дата)

Сафронова М.А.
(инициалы и
фамилия)

ТУЛА 202__

ПЕРЕЧЕНЬ РАЗДЕЛОВ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

Титульный лист

Бланк задания

Содержание

ВВЕДЕНИЕ

ЗАДАЧА 1

1. Постановка задачи
2. Описание входной и выходной информации
3. Алгоритм решения задачи
4. Общие требования к программе
5. Описание структуры программы для решения задачи
6. Инструкции по эксплуатации программ
7. Описание контрольного примера

ЗАДАЧА 2

1. Постановка задачи
2. Описание входной и выходной информации
3. Алгоритм решения задачи
4. Общие требования к программе
5. Описание структуры программы для решения задачи
6. Инструкции по эксплуатации программ
7. Описание контрольного примера

ЗАКЛЮЧЕНИЕ

Библиографический список

Приложения <Листинг программ>

Приложение 4

РЕЦЕНЗИЯ

на курсовую работу (курсовой проект) студента Тульского государственного университета,
гр. _____

(фамилия, имя, отчество)

по дисциплине _____

на тему: _____

Тематика работы (задание) _____ профилю (направленности)
(соответствует/ не соответствует)

образовательной программы, а также в полной мере способствует формированию необходимых компетенций (установленных в рабочей программе) у обучающихся.

Содержание работы _____ заданной тематике.
(соответствует/ не соответствует)

Объем работы _____ для раскрытия темы.
(достаточный/ не достаточный)

Оформление работы _____ установленным требованиям.
(соответствует/ не соответствует)

Тема работы раскрыта в _____ мере.
(полной/ не полной)

Использованная при выполнении работы (проекта) литература _____ .
(актуальна/ не актуальна)

Качество приложений (при наличии) _____ установленным
(соответствует/ не соответствует)

требованиям.

Замечания: _____
(отсутствуют или перечисляются замечания рецензента к работе)

Качество выполнения работы свидетельствуют о _____ уровне
(недостаточном/ пороговом/ повышенном)

сформированности необходимых компетенций (установленных в рабочей программе).
Работа выполнена на _____ уровне и, при соответствующей защите
(высоком/ среднем / низком)

может быть оценена " _____ ".

Рецензент _____ / _____
(подпись) (должность, звание, ФИО)

Дата _____

Приложение 5

ВАРИАНТЫ ИНДИВИДУАЛЬНЫХ ЗАДАНИЙ ПО ПОСЛЕДНИМ 2-м ЦИФРАМ НОМЕРА ЗАЧЕТКИ

Номеру Варианта КР соответствуют последние две цифры номера зачетки

Вариант КР/ Две последние цифры зачетки	Номера индивидуальных заданий (Приложение 6)	Вариант КР/ Две последние цифры зачетки	Номера индивидуальных заданий (Приложение 6)	Вариант КР /Две последние цифры зачетки	Номера индивидуальных заданий (Приложение 6)
0/00	52, 17	41/41	70, 39	82/82	5, 14
1/01	43, 29	42/42	46, 30	83/83	9, 56
2/02	65, 36	43/43	14, 34	84/84	26, 5
3/03	69, 42	44/44	65, 16	85/85	26, 62
4/04	65, 6	45/45	40, 11	86/86	31, 12
5/05	15, 28	46/46	6, 14	87/87	42, 10
6/06	41, 63	47/47	22, 37	88/88	42, 38
7/07	48, 54	48/48	6, 67	89/89	67, 37
8/08	64, 19	49/49	10, 44	90/90	52, 68
9/09	38, 69	50/50	37, 61	91/91	25, 45
10/10	44, 8	51/51	15, 52	92/92	25, 70
11/11	38, 69	52/52	48, 61	93/93	46, 37
12/12	29, 36	53/53	44, 10	94/94	5, 40
13/13	44, 38	54/54	11, 38	95/95	8, 47
14/14	54, 65	55/55	1, 42	96/96	16, 19
15/15	63, 47	56/56	8, 67	97/97	62, 40
16/16	41, 43	57/57	66, 65	98/98	14, 21
17/17	20, 55	58/58	10, 68	99/99	60, 18
18/18	59, 64	59/59	23, 61		
19/19	35, 15	60/60	11, 49		
20/20	50, 1	61/61	9, 1		
21/21	64, 68	62/62	48, 47		
22/22	21, 46	63/63	13, 42		
23/23	37, 35	64/64	17, 9		
24/24	19, 7	65/65	29, 12		
25/25	14, 33	66/66	58, 37		
26/26	59, 26	67/67	58, 4		
27/27	65, 48	68/68	57, 69		
28/28	48, 53	69/69	70, 52		
29/29	67, 32	70/70	50, 54		
30/30	27, 13	71/71	43, 15		
31/31	4, 62	72/72	65, 27		
32/32	67, 69	73/73	32, 2		
33/33	64, 66	74/74	23, 45		
34/34	40, 53	75/75	15, 47		
35/35	57, 24	76/76	61, 12		
36/36	50, 69	77/77	61, 14		
37/37	51, 62	78/78	61, 23		
38/38	28, 2	79/79	18, 47		
39/39	65, 51	80/80	70, 29		
40/40	59, 4	81/81	21, 56		

Номер и индивидуальные задания по КР

1. Префиксные и постфиксные суммы
2. Префиксные и постфиксные XOR
3. Префиксные и постфиксные максимумы/минимумы.
Максимумы/минимумы в скользящем окне
4. Битовые операции и их применение
5. Сложение разряженных матриц
6. Алгоритмы сортировки : цифровая сортировка
7. Алгоритмы устойчивой сортировки: сортировка пузырьком
8. Алгоритмы устойчивой сортировки: сортировка перемешиванием
9. Алгоритмы устойчивой сортировки: сортировка вставками
10. Алгоритмы устойчивой сортировки: сортировка вставками и пузырьком (гибрид)
11. Алгоритмы устойчивой сортировки: сортировка слиянием
12. Алгоритмы устойчивой сортировки: сортировка с помощью двоичного дерева
13. Алгоритмы устойчивой сортировки: сортировок вставками и слиянием (гибрид)
14. Алгоритмы неустойчивой сортировки: сортировка выбором
15. Алгоритмы неустойчивой сортировки: сортировка расчёской
16. Алгоритмы неустойчивой сортировки: сортировка Шелла
17. Алгоритмы неустойчивой сортировки: пирамидальная сортировка
18. Алгоритмы неустойчивой сортировки: плавная сортировка
19. Алгоритмы неустойчивой сортировки: быстрая сортировка
20. Алгоритмы неустойчивой сортировки: интроспективная сортировка
21. Непрактичные алгоритмы сортировки: Bogosort
22. Непрактичные алгоритмы сортировки: сортировка перестановкой
23. Алгоритмы, не основывающиеся на сравнениях: блочная сортировка
24. Алгоритмы, не основывающиеся на сравнениях: поразрядная сортировка
25. Алгоритмы, не основывающиеся на сравнениях: сортировка подсчётом
26. Сортировка строк: лексикографический порядок
27. Бинарный поиск в массиве
28. Бинарный поиск по ответу
29. Вещественный бинарный поиск
30. Тернарный поиск (вещественный)
31. Тернарный поиск (дискретный)
32. Организация перебора
33. Сокращение перебора
34. Структуры данных: множества (set)
35. Структуры данных: map

36. Структуры данных: очередь
37. Структуры данных: очередь с приоритетом
38. Структуры данных: стек
39. Структуры данных: хеш-таблица
40. Структуры данных: массивы и кортежи
41. Структуры данных: динамический массив
42. Структуры данных: двоичная куча
43. Деревья отрезков и запросы на диапазонах
44. Графы: представление
45. Графы: поиск в ширину
46. Графы: поиск в глубину
47. Графы: кратчайшие расстояния
48. Графы: мин. остовное дерево
49. Графы: поиск циклов
50. Графы: компоненты связности
51. Графы: корневые деревья
52. Графы: топологическая сортировка
53. Графы: вычисление диаметра дерева
54. Графы: нахождение предков в деревьях и наименьших общих предков
55. Алгоритм Дейкстры поиска кратчайшего пути в графе.
56. Строки: Z-функция строки
57. Строки: префикс-функция
58. Строки: алгоритмы хеширования в задачах на строк
59. Строки: дерево палиндромов
60. Строки: суффиксный массив
61. Рекурсивные алгоритмы
62. Записи
63. Методы сортировки
64. Типизированные файлы
65. Текстовые файлы
66. Нетипизированные файлы
67. Стеки
68. Односвязные списки
69. Двусвязные списки
70. Деревья