Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Макаренко Елена Николаевна
Должность: Ректор
Дата подписания: 09.09.2026 16:03:22
Уникальный программный ключ:
c098bc0c1041cb2a4cf926cf171d6715d99a6ae00adc8e27b55cbe1e2dbd7c78

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Ростовский государственный экономический университет (РИНХ)»

УТВЕРЖДАЮ
Начальник
учебно-методического управления
Т.К. Платонова
«25» мая 2026 г.

Рабочая программа дисциплины
Инженерное программирование

Направление подготовки
09.03.04 Программная инженерия

Направленность (профиль) программы бакалавриата
09.03.04.01 Системное и прикладное программное обеспечение

Для набора 2026 года

Квалификация
Бакалавр

КАФЕДРА **Информационные технологии и программирование**

Распределение часов дисциплины по семестрам / курсам

Семестр (<Курс>.<Семестр на курсе>)	**1 (1.1)**		**2 (1.2)**		Итого	
Недель	16		16			
Вид занятий	УП	РП	УП	РП	УП	РП
Лекции	32	32	32	32	64	64
Лабораторные	32	32	32	32	64	64
Итого ауд.	64	64	64	64	128	128
Контактная работа	64	64	64	64	128	128
Сам. работа	44	44	44	44	88	88
Часы на контроль			36	36	36	36
Итого	108	108	144	144	252	252

ОСНОВАНИЕ

Учебный план утвержден учёным советом Университета (протокол № 9 от 03.03.2026 г.).

Программу составил(и): к.э.н., доцент, Жилина Е.В.

Зав. кафедрой: к.э.н., доцент Е.В. Ефимова

Методический совет: д.э.н., профессор Е.Н. Тищенко

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1	Изучение методов структурного и объектно-ориентированного программирования, владение знаниями, умениями и навыками в области информатики и проектирования, конструирования и начала тестирования программного обеспечения с помощью современных инструментальных средств.

2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

ПК-3. способен разрабатывать компоненты программных комплексов (в том числе интерфейсы, драйвера, компиляторы, загрузчики, сборщики, системные утилиты) и баз данных с использованием современных инструментальных средств и технологий программирования

В результате освоения дисциплины обучающийся должен:

Знать:

методы структурного и объектно-ориентированного программирования, особенности технологий разработки программного обеспечения (соотнесено с индикатором ПК-3.1);

Уметь:

применять методы проектирования, конструирования и тестирования программного обеспечения, использовать современные технологии разработки программных продуктов (соотнесено с индикатором ПК-3.2);

Владеть:

навыками разработки программного обеспечения с использованием современных технологий (соотнесено с индикатором ПК-3.3).

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Раздел 1. «Структуры данных»

№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
1.1	Тема 1. «Типы данных в языке С++» Понятие типов данных в языках программирования, классификация типов данных в С++, особенности представления базовых типов и операций над ними, рекомендации и правила выполнения операции преобразования базовых типов в С++. Подставляемые (встраиваемые) функции. Базовые алгоритмы: линейный, ветвящийся (конструкция if, switch), циклы (for, while). Основная библиотека iostream. Библиотека iomanip. Библиотека locale, вывод русскоязычных слов в кириллическом формате. Основы составления алгоритмов в графическом виде.	Лекционные занятия	1	4	ПК-3
1.2	Тема 1. Решение задачи по математической обработке данных, применений тригонометрических функций.	Лабораторные занятия	1	4	ПК-3
1.3	Тема 1. «Типы данных в языке С++» Изучение стандартной библиотеки math.h на основе API языка C/C++, описанного в справочном разделе.	Самостоятельная работа	1	4	ПК-3
1.4	Тема 2. «Функции на языке С++» Понятие функции. Параметры функции: формальные, фактические, входные, результатные. Типы данных функций. Способы программной организации функций в различных проектах. Заголовочные файлы. Подключение зависимостей в проекте. Передача параметров. Перегрузка функций.	Лекционные занятия	1	4	ПК-3
1.5	Тема 2. Решение разветвляющихся алгоритмов.	Лабораторные занятия	1	4	ПК-3
1.6	Тема 2. «Функции на языке С++» Самостоятельная реализация программного решения по расчету заработной платы сотрудника с учетом его оклада, премии и налоговых отчислений. Предусмотреть возможность обработки нескольких сотрудников. Вывести итоговую ведомость начислений в табличном виде.	Самостоятельная работа	1	4	ПК-3
1.7	Тема 3. «Массивы: одномерные и двумерные» Понятие и виды массивов в языках программирования, определение, объявление, инициализация, способы генерации и вывод одномерных массивов, расположение в памяти элементов массивов, связь между указателями и массивами, задачи поиска, замены и перестановок элементов массива, сортировок элементов массива. Двумерные массивы: задачи поиска, замены и суммирования элементов двумерного массива, задачи сортировок и перестановок в двумерных массивах. Применение функций при обработке элементов массива.	Лекционные занятия	1	4	ПК-3

	Библиотека time.h.				
1.8	Тема 3. Решение циклических алгоритмов.	Лабораторные занятия	1	4	ПК-3
1.9	Тема 3. «Массивы: одномерные и двумерные» Реализация алгоритмов по обработке массивов на основе сторонних примеров, с описанием проделанной работы и со ссылкой на используемый пример из книги из списка литературы	Самостоятельная работа	1	6	ПК-3
1.10	Тема 4. «Ссылки и указатели. Динамическая память» Понятия и определения. Особенности применения. Отличительные возможности. Способы инициализация указателей на объекты, методы доступа к данным через указатели, размещение указателей и адресуемых ими объектов в памяти, операции над указателями. Передача параметров по ссылке в функцию. Передача параметров по указателю в функцию. Возврат значений из функции через указатель. Распределение памяти. Динамическое выделение памяти. Одномерные динамические массивы. Двумерные динамические массивы. Очистка памяти, удаление динамических объектов. Указатели на функции.	Лекционные занятия	1	4	ПК-3
1.11	Тема 4. Программирование с использованием функций и рекурсий. Перегрузка функций.	Лабораторные занятия	1	4	ПК-3
1.12	Тема 4. «Ссылки и указатели. Динамическая память» Разработка программных решений по обработке динамических двумерных массивов. Написание функций обработки элементов: поиск, добавление и удаление. Программно реализовать поворот матрицы на 90 градусов.	Самостоятельная работа	1	4	ПК-3

Раздел 2. «Алгоритмы обработки данных»

№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
2.1	Тема 5. «Символьные данные и строки» Понятия и определения символьных данных и строк, сходство и отличия их внутреннего представления, способы объявления, инициализация строк, методы доступа к элементам строк, определение размера строк, различные способы организации ввода/вывода символьных данных и строк. Обработка char переменных, и char массивов. Библиотека string.h. Обработка string переменных, ввод предложений. Обработка текста в функциях.	Лекционные занятия	1	4	ПК-3
2.2	Тема 5. Программирование с использованием строк	Лабораторные занятия	1	4	ПК-3
2.3	Тема 5. "Символьные данные и строки" Разработка программного решения задачи по замене предыдущего года в тексте на текущий. Программно обработать строку, удалив из нее символьную последовательность, расположенную между символами «/*», «*/». Реализовать алгоритм по нахождению в строке указанную подстроку и замене ее на новую, ввод данных через диалоговый режим на основе API языка C/C++	Самостоятельная работа	1	10	ПК-3
2.4	Тема 6. «Структуры и объединения» Битовые поля. Определения, способы объявления, инициализация структур, методы доступа к данным структуры, размещение структур и определение их размера в памяти, массивы структур. Структуры и указатели. Объединения.	Лекционные занятия	1	4	ПК-3
2.5	Тема 6. Программирование с использованием структур	Лабораторные занятия	1	4	ПК-3
2.6	Тема 6. "Структуры и объединения" Разработка программного решения в по обработке пользовательского типа данных на примере сложной структуры.Организация ввода/вывода русскоязычного текста.	Самостоятельная работа	1	6	ПК-3
2.7	Тема 7. «Файлы. Функции по работе с файлами» Понятия и определения файлов и потоков, классификация файлов, стандартные функции по работе с файлами, основные алгоритмы работы с файлами, примеры решения задач на организацию ввода-вывода и обработку данных в файлах. Обработка файлов в потоковом режиме. Прямой доступ к данным файла. Работа с элементами файлов. Передача файлов через командную строку операционной системы, обработка количества входных параметров.	Лекционные занятия	1	4	ПК-3
2.8	Тема 7. Работа с файлами: стандартные библиотеки и функции по работе с файлами. Обработка файлов в потоковом режиме. Прямой доступ к данным файла. Работа с элементами файлов. Чтение/Запись.	Лабораторные занятия	1	4	ПК-3
2.9	Тема 8. «Алгоритмы сортировки массивов» Определение и классификация алгоритмов сортировок массивов. Быстрая сортировка, параметры, трудоемкость алгоритмов. Программное описание и примеры кодов базовых алгоритмов сортировок: пузырьковая, вставками, выбором. Бинарная пирамидальная сортировка, сортировка Шелла.	Лекционные занятия	1	4	ПК-3

2.10	Тема 8. Сортировка по ключу элементов массива. Метод сортировки: пузырьковая, вставками, выбором.	Лабораторные занятия	1	4	ПК-3
2.11	Тема 8. «Алгоритмы сортировки массивов» Сортировка Хоара. Алгоритмы на графах. Алгоритмы нахождения кратчайшего пути: постановка задачи и описание алгоритмов нахождения кратчайшего пути в графах, программные реализации алгоритма Дейкстры. Определение и виды деревьев поиска, приемы снижения трудоемкости поиска в древовидных структурах, описания алгоритмов поиска в двоичных упорядоченных, случайных и сбалансированных в высоту деревьях, примеры программной реализации бинарного дерева поиска.	Самостоятельная работа	1	10	ПК-3
2.12	Подготовка к промежуточной аттестации	Зачет	1	0	ПК-3

Раздел 3. «Языки программирования»

№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
3.1	Тема 9. «Развитие языков программирования» История ЯП. Классификация ЯП. Области применения ЯП. Парадигмы программирования. Процедурные языки. Аппликативные языки. Языки логического программирования. Объектно-ориентированные языки. Стандартизация языков программирования.	Лекционные занятия	2	2	ПК-3
3.2	Тема 9. Разработка программы-класса, содержащей атрибуты, методы класса.	Лабораторные занятия	2	4	ПК-3
3.3	Тема 10. «Основные понятия объектно-ориентированного программирования (ООП)» Объектно-ориентированное программирование. Объект. Класс. Экземпляр класса. Инкапсуляция. Наследование. Полиморфизм. Секции private, protected, public. Указатель this. Переопределение операций.	Лекционные занятия	2	2	ПК-3
3.4	Тема 10. Разработка программы-класса, методы класса. Защищенность. Переопределение операций	Лабораторные занятия	2	2	ПК-3
3.5	Тема 10. «Основные понятия объектно-ориентированного программирования (ООП)» Преобразование типов на языке C++. static_cast. dynamic_cast. Практика преобразования объектов.	Самостоятельная работа	2	4	ПК-3
3.6	Тема 11. «Компоновка файлов в одну программу» Поток и is_open(). Передача параметров через консоль (int argc, char *argv[]). Режимы файла. Бинарные файлы. Решение. Проект. Директивы #include, #define, #undef, #ifdef, #else, #if, #pragma. Макроимена. __FILE__ и __LINE__.Условная компиляция.	Лекционные занятия	2	2	ПК-3
3.7	Тема 11. Реализовать объединение нескольких проектов в одно решение	Лабораторные занятия	2	2	ПК-3

Раздел 4. «Практическое инженерное программирование»

№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
4.1	Тема 12. «Конструкторы и деструкторы. Инкапсуляция» Конструктор. Инициализация переменных в конструкторе. Конструктор без параметров. Копирующий конструктор. Содержательный конструктор. Константные ссылки и указатели. Константные методы классов. Перегрузка. Инкапсуляция. Реализация класса комплексного числа.	Лекционные занятия	2	4	ПК-3
4.2	Тема 12. Разработка Разработка программы с конструктором и деструктором с использованием шаблона. Инициализация свойств. Const методы класса. Разработка программы с конструктором и деструктором без использования шаблона. Code - OSS	Лабораторные занятия	2	2	ПК-3
4.3	Тема 12. «Конструкторы и деструкторы. Инкапсуляция» Реализация алгоритма по вычислению площади треугольника по формуле Герона. Программирование решения полинома.	Самостоятельная работа	2	10	ПК-3
4.4	Тема 13. «Наследование» Понятие наследования. Виды наследования. Множественное наследование. Особенности.	Лекционные занятия	2	4	ПК-3
4.5	Тема 13. Реализация программного кода, используя наследование классов. Разработка программы с множественным наследованием. Code - OSS	Лабораторные занятия	2	4	ПК-3
4.6	Тема 13. «Наследование» Реализуйте класса Vectors, содержащий 2 конструктора, деструктор и функцию определения длины вектора в трёхмерном пространстве через координаты x, y, z.	Самостоятельная работа	2	4	ПК-3
4.7	Тема 14. «Дружественные функции. Дружественные классы» Дружественные функции. Дружественные члены класса (методы).	Лекционные занятия	2	4	ПК-3

	Дружественные классы. Объявление класса. Использование классов IOSTREAM. Операции << (>>) вставки (вывода) данных в поток (inserting).				
4.8	Тема 14. Разработка функций-друзей и обращение к ним. Code - OSS	Лабораторные занятия	2	4	ПК-3
4.9	Тема 14. «Дружественные функции. Дружественные классы» Программная реализация класса по выводу чисел в HEX формате, без пробелов. Реализация проекта «дождь»: на экране появляется (случайным образом) 50 точек и они движутся сверху вниз; в момент, когда одна из точек достигнет нижней границы экрана, точка должна появиться в верхней части.	Самостоятельная работа	2	6	ПК-3
4.10	Тема 15. «Виртуальные методы. Абстрактные классы. Обработка исключений» Виртуальные методы. Виртуальная функция. Виртуальные классы. Абстрактные классы. Виды ошибок. Исключительные ситуации. Обработка исключительных ситуаций.	Лекционные занятия	2	4	ПК-3
4.11	Тема 15. Разработка классов-друзей и обращение к ним.	Лабораторные занятия	2	4	ПК-3
4.12	Тема 15. «Виртуальные методы. Абстрактные классы. Обработка исключений» Реализация класса MyLine (линия), который является производным от классов Position (позиция: x1, y1, x2, y2) и Properties (свойства: цвет, ширина). Программирование метода NewDraw в классе MyLine для отрисовки линии с заданными параметрами. Классс HWND, HDC, HPEN и POINT Изучение методов MoveToEx() - перемещение курсора, LineTo() - отрисовка линии, DeleteObject() - удаление кисти	Самостоятельная работа	2	4	ПК-3
4.13	Тема 16. «Стандартная библиотека STL» Шаблоны методов. Шаблоны классов. Назначение шаблонов. Функции-шаблоны. Умный указатель. Динамические структуры данных: однонаправленные и двунаправленные списки, очередь и стек.	Лекционные занятия	2	6	ПК-3
4.14	Тема 16. Разработка виртуальных методов, абстрактных классов.	Лабораторные занятия	2	4	ПК-3
4.15	Тема 17 «Статистические и динамические библиотеки» Разработка LIB и DLL. Сетевая модель TCP/IP. Понятие сокета. Библиотека "winsock.h". Разработка программного решения "Сканер портов". Установка зависимостей в проекте. Директива #pragma comment. Конструкция __declspec(dllexport). Ссылки на библиотеку статическую.	Лекционные занятия	2	4	ПК-3
4.16	Тема 17 Разработка шаблонов классов.	Лабораторные занятия	2	6	ПК-3
4.17	Выполнение индивидуального задания	Самостоятельная работа	2	16	ПК-3
4.18	Подготовка к промежуточной аттестации	Экзамен	2	36	ПК-3

4. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Структура и содержание фонда оценочных средств для проведения текущего контроля и промежуточной аттестации представлены в Приложении 1 к рабочей программе дисциплины.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

5.1. Учебные, научные и методические издания

	Авторы, составители	Заглавие	Издательство, год	Библиотека / Количество
1	Кетков Ю. Л.	Введение в языки программирования С и С++: курс: учебное пособие	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2008	ЭБС «Университетская библиотека онлайн»
2	Страуструп Б.	Язык программирования С++ для профессионалов: практическое пособие	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2006	ЭБС «Университетская библиотека онлайн»
3	Мейер Б.	Объектно-ориентированное программирование и программная инженерия: учебное пособие	Москва: Национальный Открытый Университет «ИНТУИТ», 2016	ЭБС «Университетская библиотека онлайн»

	Авторы, составители	Заглавие	Издательство, год	Библиотека / Количество
4	Агафонов Е. Д., Ващенко Г. В.	Прикладное программирование: учебное пособие	Красноярск: Сибирский федеральный университет (СФУ), 2015	ЭБС «Университетская библиотека онлайн»
5	Сорокин А. А.	Объектно-ориентированное программирование: учебное пособие: курс лекций	Ставрополь: Северо-Кавказский Федеральный университет (СКФУ), 2014	ЭБС «Университетская библиотека онлайн»
6		Программные продукты и системы: журнал	Тверь: Центрпрограммсистем, 2017	ЭБС «Университетская библиотека онлайн»
7	Баранова, И. В., Баранов, С. Н., Баженова, И. В., Кучунова, Е. В., Толкач, С. Г.	Объектно-ориентированное программирование на C++: учебник	Красноярск: Сибирский федеральный университет, 2019	ЭБС «IPR SMART»
8	Родыгин А. В.	Информационные технологии: алгоритмизация и программирование: учебное пособие	Новосибирск: Новосибирский государственный технический университет, 2017	ЭБС «Университетская библиотека онлайн»
9	Зайцев М. Г.	Объектно-ориентированный анализ и программирование: учебное пособие	Новосибирск: Новосибирский государственный технический университет, 2017	ЭБС «Университетская библиотека онлайн»

5.2. Профессиональные базы данных и информационные справочные системы

Портал сообщества Cppreference. Раздел справки по языку C++ - https://devdocs.io/cpp
Образовательный портал "Основы программирования на языках Си и C++ для начинающих" - http://cppstudio.com/
ИСС "КонсультантПлюс"
ИСС "Гарант"http://www.internet.garant.ru/

5.3. Перечень программного обеспечения

Операционная система РЕД ОС
Code - OSS
MySQL

5.4. Учебно-методические материалы для обучающихся с ограниченными возможностями здоровья

При необходимости по заявлению обучающегося с ограниченными возможностями здоровья учебно-методические материалы предоставляются в формах, адаптированных к ограничениям здоровья и восприятия информации. Для лиц с нарушениями зрения: в форме аудиофайла; в печатной форме увеличенным шрифтом. Для лиц с нарушениями слуха: в форме электронного документа; в печатной форме. Для лиц с нарушениями опорно-двигательного аппарата: в форме электронного документа; в печатной форме.

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Помещения для всех видов работ, предусмотренных учебным планом, укомплектованы необходимой специализированной учебной мебелью и техническими средствами обучения:
- столы, стулья;
- персональный компьютер / ноутбук (переносной);
- проектор;
- экран / интерактивная доска.

Лабораторные занятия проводятся в компьютерных классах, рабочие места в которых оборудованы необходимыми лицензионными и/или свободно распространяемыми программными средствами и выходом в Интернет, и/или в специализированных лабораториях, предусмотренных образовательной программой.
Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа к электронной информационно-образовательной среде.

7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Методические указания по освоению дисциплины представлены в Приложении 2 к рабочей программе дисциплины.

Приложение 1

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

1 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

1.1 Показатели и критерии оценивания компетенций:

ЗУН, составляющие компетенцию	Показатели оценивания	Критерии оценивания	Средства оценивания*
ПК-3: способен разрабатывать компоненты программных комплексов (в том числе интерфейсы, драйвера, компиляторы, загрузчики, сборщики, системные утилиты) и баз данных с использованием современных инструментальных средств и технологий программирования			
З. методы структурного и объектно-ориентированного программирования, особенности технологий разработки программного обеспечения	изучение основной и дополнительной литературы, лекционного материала, использование профессиональных баз данных для изучения технологий разработки программного обеспечения, обработки элементов массивов	полнота и содержательность ответа, обоснованность выбора стека технологий разработки программного обеспечения при обработке элементов массивов	Вопросы к зачету (1-30), практико-ориентированные задания к зачету (1-10), вопросы к экзамену (1-40), практико-ориентированные задания к экзамену (1-10), опрос (1 семестр: 1-24, 2 семестр: 1-12), лабораторные задания (1 семестр: раздел 1-2, 2 семестр: раздел 3-4), индивидуальное задание (2 семестр: 1-10)
У. применять методы проектирования, конструирования и тестирования программного обеспечения, использовать современные технологии разработки программных продуктов	решение лабораторных, практико-ориентированных заданий при обработке элементов массивов и индивидуального задания с использованием современных технологий	правильность выполнения заданий с использованием современных технологий	Вопросы к зачету (1-30), практико-ориентированные задания к зачету (1-10), вопросы к экзамену (1-40), практико-ориентированные задания к экзамену (1-10), опрос (1 семестр: 1-24, 2 семестр: 1-12), лабораторные задания (1 семестр: раздел 1-2, 2 семестр: раздел 3-4), индивидуальное задание (2 семестр: 1-10)
В. навыками разработки программного обеспечения с использованием современных технологий	авторское программное решение лабораторных и практико-ориентированных заданий на обработку элементов массивов, выполнение индивидуального задания	полнота автоматизации авторского решения с использованием современных технологий	Вопросы к зачету (1-30), практико-ориентированные задания к зачету (1-10), вопросы к экзамену (1-40), практико-ориентированные задания к экзамену (1-10), опрос (1 семестр: 1-24, 2 семестр: 1-12), лабораторные задания (1 семестр: раздел 1-2, 2 семестр: раздел 3-4), индивидуальное задание (2 семестр: 1-10)

1.2 Шкалы оценивания:

Текущий контроль успеваемости и промежуточная аттестация осуществляется в рамках накопительной балльно-рейтинговой системы в 100-балльной шкале:

1 семестр:

50-100 баллов (зачтено)

0-49 баллов (не зачтено)

2 семестр:

84-100 баллов (оценка «отлично»)

67-83 баллов (оценка «хорошо»)

50-66 баллов (оценка «удовлетворительно»)

0-49 баллов (оценка «неудовлетворительно»)

2 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

1 семестр

Вопросы к зачету

1. Понятие типов данных в языках программирования, классификация типов данных в С++.
2. Особенности представления базовых типов и операций над ними.
3. Рекомендации и правила выполнения операции преобразования базовых типов в С++.
4. Функции в С++. Формальные параметры, фактические параметры.
5. Перегрузка функций.

6. Передача параметров в функции по значению. Передача массивов в функцию в качестве параметров.
7. Рекурсивные функции.
8. Определение и виды указателей, способы объявления, инициализация указателей на объекты.
9. Методы доступа к данным через указатели, размещение указателей и адресуемых ими объектов в памяти, операции над указателями.
10. Понятие и виды массивов, определение, объявление, инициализация, способы генерации и вывод одномерных массивов.
11. Задачи поиска, замены и перестановок элементов массива, нахождения мин/макс элемента.
12. Двумерные массивы: задачи поиска, замены и суммирования элементов двумерного массива, задачи перестановок в двумерных массивах.
13. Одномерные динамические массивы. Двумерные динамические массивы.
14. Расположение в памяти элементов динамических массивов, связь между указателями и массивами.
15. Распределение, способы выделения и освобождения динамической памяти, взаимодействие указателей и участков динамической памяти.
16. Программирование с использованием строк. Ввод/вывод строк. Тип данных char.
17. Программирование с использованием строк. Ввод/вывод строк. Тип данных string.
18. Программирование с использованием строк. Возможности библиотеки string.h.
19. Программирование с использованием строк. Возможности библиотеки string.
20. Определения, способы объявления, инициализация структур, методы доступа к данным структуры, размещение структур и определение их размера в памяти, массивы структур.
21. Битовые поля.
22. Структуры и указатели.
23. Объединения.
24. Понятия и определения файлов и потоков, классификация файлов, стандартные функции по работе с файлами, основные алгоритмы работы с файлами.
25. Обработка файлов в потоковом режиме. Прямой доступ к данным файла. Работа с элементами файлов.
26. Запись/чтение элементов структуры в файл.
27. Запись/чтение элементов массивов в файл.
28. Пузырьковая сортировка.
29. Сортировка вставками.
30. Сортировка выбором.

Практико-ориентированные задания к зачету

1. Подсчитать количество слов, заканчивающихся на букву “z”.
2. В двумерном массиве найти среднее значение элементов каждого столбца.
3. Список товаров, имеющихся в магазине, включает в себя наименование товара, количество единиц товара, цену единицы, дату поступления товара на склад и срок годности. Вывести список товаров, срок годности, которых истек.
4. Ведомость Клиентов содержит ФИО, адрес, телефон. Сформировать список клиентов, проживающих в городе Белая Калитва.
5. Информация об участниках спортивных соревнований содержит название страны, название команды, ФИО игрока, игровой номер, возраст, рост и вес. Вывести фамилии спортсменов, возраст которых больше 21 года.
6. В одномерном массиве переставьте элементы таким образом, чтобы на 5-ом месте стояло бы наибольшее, а на 6-ом — наименьшее число.
7. Подсчитать количество слов в предложении, при условии, что слова разделены пробелом (но могут встречаться и несколько пробелов), в конце может быть . ! ?
8. Найти слова, содержащие подстроку “app”.
9. Написать функцию, которая вычисляет значение a^b. Числа a и b целые, могут быть как положительными, так и отрицательными.
10. Написать функцию по вычислению дискриминанта квадратного уравнения

Критерии оценивания:
- 50-100 баллов («зачтено»): – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленной программой курса целью обучения; правильные, уверенные действия по применению полученных навыков и умений при решении практико-ориентированного задания, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой;
- 0-49 баллов («не зачтено») – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять навыки и умения при решении практико-ориентированного задания, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

2 семестр

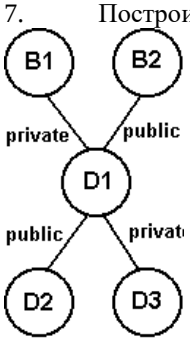
Вопросы к экзамену

1. История ЯП. Классификация и область применения ЯП.
2. Парадигмы программирования. Процедурные языки. Аппликативные языки.

3. Языки логического программирования. Объектно-ориентированные языки.
4. Стандартизация языков программирования.
5. Среда проектирования VisualStudio.NET.
6. Потоковый ввод\вывода данных на языке C/C++. Операторы ввода\вывода (cout и cin). Библиотека <iostream>. Команда using namespace std.
7. Операция увеличения ++ и уменьшения --: «префиксная» форма, и «постфиксная» форма. Проверка четности числа. Логические операции. Библиотека conio.h. Применение метода system в проектах.
8. Программирование разветвляющихся алгоритмов. Условные операторы if. Оператор выбора switch
9. Функции библиотеки math.h. Линейный алгоритм.
10. Программирование циклических алгоритмов.
11. Программирование с использованием одномерных массивов.
12. Программирование с использованием двумерных массивов.
13. Обработка строк.
14. Компоновка файлов в решении. Сборщик «мусора».
15. Объектно-ориентированное программирование. Объект. Класс. Экземпляр.
16. Инкапсуляция.
17. Полиморфизм.
18. Секции private, protected, public. Указатель this.
19. Переопределение операций.
20. Потоки работы с файлами.
21. Передача параметров через консоль (int argc, char *argv[]). Режимы файла. Бинарные файлы.
22. Решение. Проект.
23. Директивы #include, #define, #undef, #ifdef, #else, #if, #pragma.
24. Макроимена. __FILE__ и __LINE__.
25. Конструктор. Инициализация переменных в конструкторе. Конструктор без параметров. Копирующий конструктор. Содержательный конструктор.
26. Константные ссылки и указатели. Константные методы классов. Перегрузка.
27. Понятие наследования. Виды наследования.
28. Множественное наследование.
29. Дружественные функции.
30. Дружественные члены класса (методы). Дружественные классы.
31. Виртуальные методы. Виртуальная функция. Виртуальные классы.
32. Абстрактные классы.
33. Виды ошибок. Исключительные ситуации. Обработка исключительных ситуаций. Операторы try и catch.
34. Шаблоны методов. Шаблоны классов. Назначение шаблонов.
35. Стандартная библиотека STL.
36. Статистические библиотеки.
37. Динамические библиотеки.
38. Библиотека "winsock.h".
39. Библиотека Crypto++.
40. Возможности WinAPI.

Практико-ориентированные задания к экзамену

1. Разработать класс строк, в котором подсчитайте количество слов, начинающихся на букву “a”.
2. Разработайте класс Car (автомобиль), у которого определены приватные переменные name (название автомобиля) и price (цена автомобиля), закрытый метод: setPrice (функция назначения цены) и дружественный класс Person (водитель), который использует объект Auto + объявлены конструкторы.
3. Реализуйте класс MyLine (линия), который является производным от классов Position (позиция: x1, y1, x2, y2) и Properties (свойства: цвет, ширина).
4. Разработайте класс TV, реализующий методы включения/выключения(onoff), переключения каналов(chanDown, chanUp), регулировки уровня громкости(volDown, volUp) и метод, отображающий все настройки (setting).
5. Разработайте класс «Заполнение строки случайными символами», содержащий метод по обработке символов через ASCII.
6. Реализовать класс «Символ в заданной позиции экрана».
7. Построить иерархию классов согласно схеме наследования (схема может отличаться)



8. Разработайте класс Массив, который содержит: перегрузку конструкторов, деструктор, метод вывода элементов и дружественную функцию, определяющую номер первой из строк, не содержащих ни одного положительного элемента (формулировка функции может отличаться).
9. Напишите класс Студент. Структуру класса продумайте самостоятельно.
10. Напишите класс Товар. Структуру класса продумайте самостоятельно.

Критерии оценивания:
- 84-100 баллов (оценка «отлично») – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленной программой курса целью обучения; правильные, уверенные действия по применению полученных навыков и умений при решении практико-ориентированного задания, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой;
- 67-83 баллов (оценка «хорошо») – наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целью обучения, правильные действия по применению навыком и умений при решении практико-ориентированного задания, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины;
- 50-66 баллов (оценка «удовлетворительно») – наличие твердых знаний в объеме пройденного курса в соответствии с целью обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению навыком и умений при решении практико-ориентированного задания;
- 0-49 баллов (оценка «неудовлетворительно») – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять умения и навыки при решении практико-ориентированного задания, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

Опрос 1 семестр

1. Принципы выбора языка программирования для решения задачи
2. Современные тенденции в информатике и программировании для решения определенного класса заданий.
3. Современные подходы для выбора ИТ-стека в разработке ПО.
4. Правила составления блок-схем.
5. Преимущества объектно-ориентированных языков.
6. Преимущества методов структурного программирования
7. Особенности ручного тестирования.
8. Контрольные точки. Точки останова.
9. Правила комментирования программного кода.
10. Выделение идеи представленного решения
11. Возможности и функционал операторов ввода\вывода (cout и cin).
12. Функции библиотеки math.h.
13. Оператор выбора switch.
14. Сложность алгоритмического решения
15. Работа с памятью.
16. Динамичность.
17. Обоснование выбранных типов данных в проекте.
18. Определение класса.
19. Понятие объекта и экземпляра класса.
20. Обработка элементов массивов.
21. Указатель this.
22. Функции потокового ввода\вывода данных printf() и scanf().
23. Программирование циклических алгоритмов.
24. Секции private, protected, public.

Критерии оценивания:
- 1 балл выставляется обучающемуся, если изложенный материал фактически верен и логически обоснован.
- 0 баллов выставляется обучающемуся, если изложенный материал фактически не верен и логически не обоснован.

Максимальное количество баллов за 1 семестр: 24 балла

Опрос 2 семестр

1. Опишите понятие конструктора.
2. Опишите понятие деструктора.
3. Полиморфизм
4. Инкапсуляция.
5. Наследование
6. Дружественные функции.
7. Особенности, использование.

8. Дружественные классы.
9. Особенности, использование.
10. Особенности абстракции в ООП.
11. Роль виртуальных классов в ООП.
12. Современные ИТ-стеки в разработке ПО

Критерии оценивания:

- 1 балл выставляется обучающемуся, если изложенный материал фактически верен и логически обоснован.
- 0 баллов выставляется обучающемуся, если изложенный материал фактически не верен и логически не обоснован.

Максимальное количество баллов за 2 семестр: 12 баллов.

Лабораторные задания

1 семестр

Раздел 1«Структуры данных»

Лабораторное задание № 1.1(5 баллов)
Решение задачи по математической обработке данных, применений тригонометрических функций.

Лабораторное задание № 1.2 (5 баллов)
Решение разветвляющихся алгоритмов.

Лабораторное задание № 1.3 (5 баллов)
Решение циклических алгоритмов.

Лабораторное задание № 1.4 (5 баллов)
Программирование с использованием функций и рекурсий. Перегрузка функций.

Лабораторное задание № 1.5 (5 баллов)
Программирование с использованием одномерных массивов: поиск, замена и перестановка элементов.

Лабораторное задание № 1.6 (5 баллов)
Программирование с использованием двумерных динамических массивов: поиск, замена и перестановка элементов.

Лабораторное задание № 1.7 (5 баллов)
Программирование с использованием динамических одномерных массивов.

Лабораторное задание № 1.8 (5 баллов)
Программирование с использованием динамических двумерных массивов: обработка элементов массива, диагонали, передача массива в функции.

Критерии оценивания:

5 б. – задание выполнено верно;
4 б. – при выполнении задания были допущены неточности, не влияющие на результат;
3 б. – при выполнении задания были допущены ошибки;
2-1 б. – при выполнении задания были допущены существенные ошибки;
0 б. – задание не выполнено.

Раздел 2 «Алгоритмы обработки данных»

Лабораторное задание № 1.9 (9 баллов)
Программирование с использованием строк

Лабораторное задание № 1.10 (9 баллов)
Программирование с использованием структур

Лабораторное задание № 1.11 (9 баллов)
Работа с файлами: стандартные библиотеки и функции по работе с файлами. Обработка файлов в потоковом режиме. Прямой доступ к данным файла. Работа с элементами файлов. Чтение/Запись.

Лабораторное задание № 1.12 (9 баллов)
Сортировка по ключу элементов массива. Метод сортировки: пузырьковая, вставками, выбором.

Критерии оценивания:

9 б. – задание выполнено верно;
8-7 б. – при выполнении задания были допущены неточности, не влияющие на результат;
6-4 б. – при выполнении задания были допущены ошибки;
3-1 б. – при выполнении задания были допущены существенные ошибки;
0 б. – задание не выполнено.

Критерии оценивания:
Баллы по каждому заданию проставлены в скобках.
Максимальное количество баллов, которые могут быть получены обучающимся в семестре, - 76.

2 семестр
Раздел 3. «Языки программирования»

Лабораторное задание № 2.1 (3 балла)
Разработка программы-класса, содержащей атрибуты, методы класса.

Лабораторное задание № 2.2 (3 балла)
Разработка программы-класса, методы класса. Защищенность. Переопределение операций

Лабораторное задание № 2.3 (3 балла)
Реализовать объединение нескольких проектов в одно решение

Раздел 4. «Практическое инженерное программирование»

Лабораторное задание № 2.4 (3 балла)
Разработка программы с конструктором и деструктором с использованием шаблона. Инициализация свойств. Const методы класса. Разработка программы с конструктором и деструктором без использования шаблона. Code - OSS

Лабораторное задание № 2.5 (5 баллов)
Реализация программного кода, используя наследование классов. Разработка программы с множественным наследованием. Code - OSS
Лабораторное задание № 2.6 (5 баллов)
Разработка функций-друзей и обращение к ним. Code - OSS

Лабораторное задание № 2.7 (5 баллов)
Разработка классов-друзей и обращение к ним.

Лабораторное задание № 2.8 (5 баллов)
Разработка виртуальных методов, абстрактных классов.

Лабораторное задание № 2.9 (5 баллов)
Разработка шаблонов классов.

Лабораторное задание № 2.10 (6 баллов)
Разработка алгоритма по взаимодействию с базой данных MySQL

Лабораторное задание № 2.11 (6 баллов)
Разработка динамической библиотеки.

Лабораторное задание № 2.12 (6 баллов)
Разработка алгоритма по шифрованию данных.

Критерии оценивания:
Баллы по каждому заданию проставлены в скобках.
Для ЛЗ 2.1-2.4
3 б. – задание выполнено верно;
2 б. – при выполнении задания были допущены ошибки;
1 б. – при выполнении задания были допущены существенные ошибки;
0 б. – задание не выполнено.
Для ЛЗ 2.5-2.9
5 б. – задание выполнено верно;
4 б. – при выполнении задания были допущены неточности, не влияющие на результат;
3 б. – при выполнении задания были допущены ошибки;
2-1 б. – при выполнении задания были допущены существенные ошибки;
0 б. – задание не выполнено.
Для ЛЗ 2.10-2.12

6 б. – задание выполнено верно;
4-5 б. – при выполнении задания были допущены неточности, не влияющие на результат;
3 б. – при выполнении задания были допущены ошибки;
2-1 б. – при выполнении задания были допущены существенные ошибки;
0 б. – задание не выполнено.

Неправильное выполнение задания – 0 баллов.
Максимальное количество баллов, которые могут быть получены обучающимся в течение 2 семестра, - 55.

Индивидуальное задание (2 семестр)

1. «Разработка проекта (программного модуля) автоматизации расписания занятий в университете»: В системе должны поддерживаться режимы поиска занятия по заданному критерию (время, преподаватель), регистрации занятий, учета занятий по типу, генерации расписаний.
2. «Разработка проекта (программного модуля) автоматизации библиотеки»: В системе должны поддерживаться режимы поиска книги по заданному критерию (автор, название), заказа книги, учета клиентов и книг в книгохранилище, выдачи отчетов по запросам (местонахождение книги в архиве или ее отсутствие), выдачи документов о должниках.
3. «Разработка проекта (программного модуля) автоматизации школы»: В системе должны поддерживаться режимы учета классов и учеников в них, регистрации нового ученика, учета посещаемости занятий и оценок учащихся, генерации отчетов по успеваемости учеников.
4. «Разработка проекта (программного модуля) поликлиники»: В системе должны поддерживаться режимы учета докторов и пациентов, поиска пациента по заданному критерию, анализа заболеваемости по районам, по месяцам и выдачи справок о болезни.
5. «Разработка проекта (программного модуля) автоматизации аптеки»: В системе должны поддерживаться режимы поиска лекарства по заданному критерию (название, болезнь, цена), заказа, покупки и учета лекарств, анализа спроса на лекарства в зависимости от стоимости, времени года и т. п., выдачи соответствующих отчетов.
6. «Разработка проекта (программного модуля) автоматизации музея»: В системе должны поддерживаться режимы учета посетителей и экспонатов музея, регистрации новых экспонатов, поиска экспонатов по заданному критерию (название, эпоха и т. д.), учета доходов и расходов, связанных с проведением экскурсий, выдачи отчетов по запросам.
7. Создание системы обмена сообщениями через протокол TCP.
8. Создание системы обмена изображениями (видео) через протокол UDP.
9. Создание FTP клиента.
10. Применение современных ИТ в работе коммивояжера (администратора, риэлтора, менеджера по закупкам, менеджера по продажам, менеджера по услугам, товароведа, в работе электронного бизнеса и т.д.).
Обучающимся самостоятельно может быть выбран индивидуальный стек ИТ-технологий, в том числе направленный на алгоритмы разработки веб-модулей, css-стилей, игр, графики, cms-систем, crm-систем, на мобильную или кроссплатформенную разработку, системное программирование (реестр, службы, драйвера), практику использования фреймворков, криптографических алгоритмов, разработку простейшего антивирусника, практику тестирования исходного кода, сравнение IDE, языков программирования, компиляторов, баз данных (SQL и NoSQL технологии), теорию алгоритмов (разные способы реализации, производительность), защищённый программный код и т.д.

Требования к оформлению индивидуального задания приведены в Приложении 2.

Критерии оценивания:
- 16-33 балла – разработанный интерфейс программы и ее функциональные возможности соответствуют требованиям задания; текстовое описание составлено в полном объеме; программа работает без сбоев для всех типовых экспериментов, для которых она разрабатывалась, предусмотрена защита от ввода некорректных данных; обучающийся показал свободное владение тематикой проекта, знание используемых компонентов, их свойств и специальных функций; изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в рамках пройденной программы; правильные, уверенные действия по применению полученных умений и навыков на практике; усвоение основной и дополнительной литературы, работа с профессиональными базами данных;
- 0-15 баллов – разработанный интерфейс программы и ее функциональные возможности не в полной мере соответствуют требованиям задания; текстовое описание составлено не в полном объеме и не достаточно аккуратно; программа работает без сбоев, не для всех типовых задач, для которых она разрабатывалась, не предусмотрена защита от ввода некорректных данных; обучающийся показал слабые знания по тематике проекта; неправильные в целом действия по применению умений и навыков на практике, отсутствие материала из основной и дополнительной литературы и профессиональных баз данных;

3 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедуры оценивания включают в себя текущий контроль и промежуточную аттестацию.
Текущий контроль успеваемости проводится с использованием оценочных средств, представленных в п. 2 данного приложения. Результаты текущего контроля доводятся до сведения студентов до промежуточной аттестации.

Промежуточная аттестация проводится в форме зачета, экзамена.

Зачет проводится по расписанию промежуточной аттестации. Количество вопросов в билете – 3. Проверка ответов и объявление результатов производится в день зачета. Экзамен проводится по расписанию промежуточной аттестации в

письменном виде. Количество вопросов в экзаменационном билете – 3. Проверка ответов и объявление результатов производится в день экзамена.

Результаты аттестации заносятся в ведомость и зачетную книжку студента. Студенты, не прошедшие промежуточную аттестацию по графику, должны ликвидировать задолженность в установленном порядке.

Приложение 2

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Учебным планом предусмотрены следующие виды занятий:
- лекции;
- лабораторные занятия.

В ходе лекционных занятий рассматриваются теоретические вопросы с учетом практико-ориентированности изучаемой дисциплины, даются рекомендации для самостоятельной работы и подготовке к лабораторным занятиям.

В ходе лабораторных занятий углубляются и закрепляются знания студентов по ряду рассмотренных на лекциях вопросов, развиваются навыки программирования, проектирования и ручного тестирования исходного кода, применения методов и технологий разработки программного обеспечения.

При подготовке к лабораторным занятиям каждый обучающийся должен:
– изучить рекомендованную учебную литературу;
– изучить практические примеры, рассмотренные на лекциях;
– подготовить ответы на все вопросы по изучаемой теме.

По согласованию с преподавателем студент может подготовить индивидуальное задание. В процессе подготовки к лабораторным занятиям студенты могут воспользоваться консультациями преподавателя.

Вопросы, не рассмотренные на лекциях и лабораторных занятиях, должны быть изучены студентами в ходе самостоятельной работы. Контроль самостоятельной работы студентов над учебной программой курса осуществляется в ходе занятий методом опроса или при выполнении лабораторных заданий. В ходе самостоятельной работы каждый студент обязан прочитать основную и по возможности дополнительную литературу по изучаемой теме, дополнить конспекты лекций недостающим материалом, выписками из рекомендованных первоисточников. Выделить непонятные термины, найти их значение в энциклопедических словарях.

Студент должен готовиться к предстоящему лабораторному занятию по всем, обозначенным в рабочей программе дисциплины вопросам.

Для подготовки к занятиям, текущему контролю и промежуточной аттестации студенты могут воспользоваться электронно-библиотечными системами. Также обучающиеся могут взять на дом необходимую литературу на абонементе университетской библиотеки или воспользоваться читальными залами.

Методические рекомендации по оформлению индивидуального задания.

Работа должна содержать теорию относительно выбранной предметной области: понятия, определения, краткая история, классификации, возможности применения, плюсы и недостатки; а также обязательно практика применения выбранного инструментария, IDE, языка программирования, библиотеки, включая скрины разработки (применения) индивидуального проекта.

Индивидуальное задание выполняется с учетом приобретенных знаний по данной дисциплине и интереса обучающегося.

Основными этапами выполнения индивидуального задания являются:
1. Постановка задачи.
2. Построение модели.
3. Разработка алгоритма.
4. Реализация алгоритма.
5. Проверка и тестирование исходного кода.
6. Составление отчета.

Постановка задачи. Это начальный этап и начинается он с ознакомления задачей и рекомендуемой литературой. Прежде чем решать задачу, необходимо ее точно сформулировать. Процесс точной формулировки задачи сводится к постановке правильных вопросов:

Понятна ли терминология, используемая в предварительной формулировке?
Что дано? Что нужно найти?
Как определить решение?
Каких данных не хватает и все ли они нужны?
Являются ли какие-то имеющиеся данные бесполезными? Какие сделаны допущения?

Возможны и другие вопросы в зависимости от конкретной задачи. Точную постановку задачи необходимо сформулировать в процессе консультации с преподавателем.

Построение модели. Задача четко поставлена, нужно сформулировать для нее математическую модель. Это очень важный шаг в процессе решения, и его надо хорошо обдумать. Выбор модели существенно влияет на остальные этапы в процессе решения. Большинство задач должно рассматриваться индивидуально.

Приступая к разработке модели, следует задать по крайней мере несколько основных вопросов:

Существует ли математическая величина, ассоциируемая с искомым результатом?
Какие математические структуры больше всего подходят для задачи?
Имеются ли какие-нибудь полезные отношения между объектами модели?
Существуют ли решенные аналогичные задачи?

Большинство решаемых задач, как правило, являются модификациями ранее решеных и для продвижения вперед приходится руководствоваться накопленным опытом.

Сделав выбор математической структуры, задачу следует переформулировать в терминах соответствующих математических объектов.

Разработка алгоритма. Как только задача четко поставлена и для нее построена модель, необходимо приступить к разработке алгоритма ее решения. Выбор метода разработки, сильно зависящий от выбора модели, может в значительной степени повлиять на эффективность алгоритма решения. Два разных алгоритма могут быть правильными, но очень сильно отличаться по эффективности. Доказательство правильности алгоритма — это один из наиболее трудных этапов создания алгоритма. Вероятно, наиболее распространенная процедура доказательства правильности программы — это прогон ее на разных тестах. Если выданные программой ответы могут быть подтверждены известными или вычисленными вручную данными, возникает вывод, что программа «работает». Однако этот метод редко исключает все сомнения; может существовать случай, в котором программа «не сработает».

Можно предложить следующую общую методику доказательства правильности алгоритма. Предположим, что алгоритм описан в виде последовательности шагов, скажем, от шага 0 до шага n. Необходимо предложить некое обоснование правомерности для каждого шага. В частности, может потребоваться лемма об условиях, действующих до и после пройденного шага. Затем необходимо предложить доказательство конечности алгоритма, при этом будут проверены все подходящие входные данные и получены все подходящие выходные данные.

Реализация алгоритма. Необходимо построить целую систему структур данных (классов) для представления важных аспектов используемой модели. Руководствуясь технологией ООП программа должна быть представлена совокупностью взаимодействующих объектов. Следует внимательно рассмотреть приведенные выше примеры с тем, чтобы попытаться применить разработанные в них классы в качестве прототипов для разработки путем наследования собственных объектов. При этом желательно исключить использование многих глобальных переменных и строгую локализацию данных и действий рамками отдельных объектов, что является основой высокой надежности программы.

Проверка программы. Проверка программы может быть охарактеризована как экспериментальное подтверждение того факта, что программа делает именно то, что должна делать. Проверка программы является также экспериментальной попыткой установить границы использования алгоритма (проекта). Недостаточно доказать правильность алгоритма. Окончательная программа должна быть тщательно проверена и оттестирована. Как выбрать входные данные для тестирования? На этот вопрос невозможно дать общего ответа. Для любого алгоритма ответ зависит от сложности программы, имеющегося ресурса времени, а также от числа вводов (т. е. вариантов входных данных), для которых можно установить правильность выводов, и т.д. Обычно множество всех вводов огромно, и полная проверка практически невозможна. Необходимо выбрать множество вводов, которые проверяют каждый участок программы.

Составление текстовой части индивидуального задания. Индивидуальное задание должен включать следующие разделы.

1. Содержание, включающее наименование всех разделов и пунктов с указанием номеров страниц.
2. Введение.

2.1 Дается характеристика предметной области, к которой относится решаемая задача и обосновывается ее актуальность.

2.2 Цель работы.

Формулируется цель выполнения индивидуального задания.

2.3 Постановка задач.

В этом разделе требуется формализовать задачи, указать возможные ограничения на их решение, CASE-средства, ИТ-технологии и т.п.

3. Техническое задание (прототип).

В соответствии с вариантом формулируется задание по индивидуальному заданию.

4. Теоретическая часть, освещающую теоретические аспекты темы;
5. Практическая часть, в которой разработка программы и ее результаты.

5.1 Анализ и разработка проекта (основной раздел отчета).

Должен отражать результаты анализа возможных вариантов решения задачи и выбора среди них наиболее рационального. Приводятся математические выкладки и рисунки, поясняющие зависимости параметров решения задачи от данных. Обосновывается выбор структур данных и основных операций над ними. Приводятся результаты моделирования программ-прототипов (если они использовались для разработки алгоритма). Определяются источники и форматы исходных данных и содержание вывода программы.

5.2 Алгоритм работы проекта.

В этом разделе приводится блок-схема алгоритма программы.

5.3 Текст программы.

Приводится текст программы с комментариями.

5.4 Результаты моделирования программы на ПК.

Должны быть приведены результаты тестирования программы с различными наборами данных, в том числе и с некорректными данными.

6.Выводы.

Приводятся комментарии к результатам и рекомендации к дальнейшему совершенствованию программы.

7. Список использованных источников.
8. Приложения

Включают материалы иллюстративного и вспомогательного характера (таблицы большого формата; дополнительные расчеты; распечатки и проч.) Приложения обозначаются русскими заглавными буквами - А, Б, В и т.д. (например, «Приложение А»), располагаются в виде заголовка, по центру.

Таблицы, рисунки, формулы оформляются в соответствии с внутривузовским изданием для нормоконтроля. На все таблицы, рисунки, литературные источники, приложения в тексте должны быть ссылки.

Оформление индивидуального задания должно соответствовать требованиям государственных стандартов, в т.ч. и методических рекомендаций вуза (кафедры). Текст работы должен быть набран на белой бумаге формата А4 с одной стороны листа. Размер шрифта: 12, интервал: 1,5. Поля: левое – 30 мм, правое – 10 мм, верхнее – 20 мм, нижнее – 20 мм.