

Документ подписан Министерством науки и высшего образования Российской Федерации
Информация о владельце:
ФИО: Макаренко Елена Николаевна
Должность: Декан
Дата подписания: 25.10.2023 10:10:36
Уникальный программный ключ:
c098bc0c1041cb2a4cf926cf171d6715d99a6ae00adc8e27b55cbe1e2dbd7c78

УТВЕРЖДАЮ
Начальник отдела лицензирования и
аккредитации
Чаленко К.Н.
« 01 » 106 20 20 г.

**Рабочая программа дисциплины
Программирование**

по профессионально-образовательной программе направление 38.03.05 "Бизнес-информатика" профиль 38.03.05.01 "Информационно-аналитические системы"

Для набора 2020 года


Квалификация
Бакалавр


КАФЕДРА Информационные технологии и защита информации**Распределение часов дисциплины по курсам**


Курс Вид занятий	2		Итого	
	уп	рп		
Лекции	6	6	6	6
Лабораторные	8	8	8	8
Итого ауд.	14	14	14	14
Контактная работа	14	14	14	14
Сам. работа	265	265	265	265
Часы на контроль	9	9	9	9
Итого	288	288	288	288

ОСНОВАНИЕ

Учебный план утвержден учёным советом вуза от 25.02.2020 протокол № 8.

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

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

Методическим советом направления: д.э.н., профессор Тищенко Е.Н. 

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ	
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	
ОПК-3:	способностью работать с компьютером как средством управления информацией, работать с информацией из различных источников, в том числе в глобальных компьютерных сетях
ОПК-1:	способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности
ОК-3:	способностью использовать основы экономических знаний в различных сферах деятельности

В результате освоения дисциплины обучающийся должен:	
Знать:	основные принципы структурного и объектно-ориентированного программирования; основы базовых алгоритмов решения задач профессиональной деятельности; теорию информации, современные методы поиска информации в глобальных компьютерных сетях;
Уметь:	использовать методы структурного и объектно-ориентированного программирования; применять методы проектирования и тестирования программного обеспечения; использовать теорию управления информацией при обосновании выбора метода программирования;
Владеть:	навыками разработки программного обеспечения на основе методов структурного и объектно-ориентированного программирования; навыками тестирования программного обеспечения; навыками анализа фактов и теорий, связанными с управлением информацией при описании предметной области;

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ					
Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетенции	Литература
	Раздел 1. «Структуры данных»				
1.1	Тема 1. «Типы данных в языке C++» Понятие типов данных в языках программирования, классификация типов данных в C++, особенности представления базовых типов и операций над ними, рекомендации и правила выполнения операции преобразования базовых типов в C++. 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. /Лек/	2	2	ОПК-3 ОПК-1 ОК-3	Л1.1 Л1.2 Л1.3 Л1.4Л2.1 Л2.3 Л2.4 Л2.5 Л2.6
1.2	Тема 1. «Типы данных в языке C++» Разработка программного решения в MS Visual Studio по обработке ввода и вывода числовых данных и математической их обработке, проверка на ошибки ввода/вывода, организация пунктов меню, разные виды завершения работы проекта. Решение системы уравнений для заданных ограничений входных параметров. Обработка все условий и проверка работоспособности исходного кода по всем веткам проекта. Составление блок-схем по всем проектам темы в MS Visio. /Лаб/	2	2	ОПК-3 ОПК-1 ОК-3	Л1.1 Л1.2 Л1.3 Л1.4Л2.1 Л2.4 Л2.5 Л2.6
1.3	Тема 2. «Функции на языке C++» 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. /Лек/	2	2	ОПК-3 ОПК-1 ОК-3	Л1.1 Л1.2 Л1.3 Л1.4Л2.1 Л2.2 Л2.4 Л2.5 Л2.6

1.4	Тема 2. «Функции на языке C++» Разработка проектов в MS Visual Studio по решению классических задач нахождение факториалов и сумм числовых последовательностей. Алгоритмическая реализация рекурсии. Программная реализация решения квадратного уравнения в MS Visual Studio. Написание функции по расчету количества корней в уравнении. Обработка ошибок на получение несуществующего решения, организация диалогового режима с пользователем. /Лаб/	2	2	ОПК-3 ОПК-1 ОК-3	Л1.1 Л1.2 Л1.3 Л1.4Л2.1 Л2.4 Л2.5
1.5	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. Библиотека time.h. /Лек/	2	2	ОПК-3 ОПК-1 ОК-3	Л1.1 Л1.2 Л1.3 Л1.4Л2.1 Л2.2 Л2.4 Л2.5
1.6	Тема 3. «Массивы: одномерные и двумерные» Массивы: одномерные массивы, задачи поиска, замены и перестановок элементов массива, задачи сортировок элементов массива. Двумерные массивы: задачи поиска, замены и суммирования элементов двумерного массива, задачи сортировок и перестановок в двумерных массивах. Диагонали квадратных матриц. Разработка проектов в MS Visual Studio. /Лаб/	2	2	ОПК-3 ОПК-1 ОК-3	Л1.1 Л1.2 Л1.3 Л1.4Л2.1 Л2.4 Л2.5
1.7	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. /Ср/	2	16	ОПК-3 ОПК-1 ОК-3	Л1.1 Л1.2 Л1.3 Л1.4Л2.1 Л2.2
	Раздел 2. «Алгоритмы обработки данных»				
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. /Ср/	2	16	ОПК-3 ОПК-1 ОК-3	Л1.1 Л1.2 Л1.3 Л1.4Л2.1 Л2.2
2.2	Тема 5. «Символьные данные и строки» Реализация в MS Visual Studio проектов: - подсчет количества слов в предложении. Предварительно привести строку к нормированному виду, удалить все лишние пробелы; - нахождение короткого слово в строке; - обработка строки, изображающей арифметическое выражение вида «<цифра>±<цифра>±...±<цифра>», где на месте знака операции «±» находится символ «+» или «-». /Лаб/	2	2	ОПК-3 ОПК-1 ОК-3	Л1.1 Л1.2 Л1.3 Л1.4Л2.1
2.3	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. /Ср/	2	16	ОПК-3 ОПК-1 ОК-3	Л1.1 Л1.2 Л1.3 Л1.4Л2.1 Л2.2 Л2.6

2.4	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. /Ср/	2	10	ОПК-3 ОПК -1 ОК-3	Л1.1 Л1.2 Л1.3 Л1.4Л2.1
2.5	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. /Ср/	2	10	ОПК-3 ОПК -1 ОК-3	Л1.1 Л1.2 Л1.3 Л1.4Л2.1 Л2.5 Л2.6
Раздел 3. «Языки программирования»					
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. Среда проектирования Visual Studio.NET. Настройка IDE. /Ср/	2	6	ОПК-3 ОПК -1 ОК-3	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.3 Л2.4
3.2	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. /Ср/	2	8	ОПК-3 ОПК -1 ОК-3	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.6
3.3	Тема 10. «Основные понятия объектно-ориентированного программирования (ООП)» Преобразование типов на языке C++. static_cast. dynamic_cast. Практика преобразования объектов. /Ср/	2	4	ОПК-3 ОПК -1 ОК-3	Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.6
Раздел 4. «Практическое инженерное программирование»					
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. /Ср/	2	10	ОПК-3 ОПК -1 ОК-3	Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.6
4.2	Тема 13. «Наследование» Понятие наследования. Виды наследования. Множественное наследование. Особенности. /Ср/	2	8	ОПК-3 ОПК -1 ОК-3	Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.6
4.3	Тема 14. «Дружественные функции. Дружественные классы» Дружественные функции. Дружественные члены класса (методы). Дружественные классы. Объявление класса. Использование классов IOSTREAM. Операции << (>>) вставки (вывода) данных в поток (inserting). /Ср/	2	10	ОПК-3 ОПК -1 ОК-3	Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.6
4.4	Тема 15. «Виртуальные методы. Абстрактные классы. Обработка исключений» Виртуальные методы. Виртуальная функция. Виртуальные классы. Абстрактные классы. Виды ошибок. Исключительные ситуации. Обработка исключительных ситуаций. /Ср/	2	12	ОПК-3 ОПК -1 ОК-3	Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3 Л2.4 Л2.6
4.5	Тема 16. «Стандартная библиотека STL» Шаблоны методов. Шаблоны классов. Назначение шаблонов. Функция-шаблоны. Умный указатель. Динамические структуры данных: однонаправленные и двунаправленные списки, очередь и стек. /Ср/	2	16	ОПК-3 ОПК -1 ОК-3	Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2

4.6	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. /Ср/	2	16	ОПК-3 ОПК -1 ОК-3	Л1.3 Л1.4Л2.1 Л2.2 Л2.4
4.7	Курсовой проект. Перечень тем представлен в Приложении 1 к рабочей программе дисциплины. /Ср/	2	107	ОПК-3 ОПК -1 ОК-3	Л1.2 Л1.3 Л1.4Л2.1 Л2.2 Л2.4
4.8	/Экзамен/	2	9	ОПК-3 ОПК -1 ОК-3	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3 Л2.4 Л2.5 Л2.6

4. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

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

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

5.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год	Колич-во
Л1.1	Кетков Ю. Л.	Введение в языки программирования С и С++: курс: учебное пособие	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2008	https://biblioclub.ru/index.php?page=book&id=234040 неограниченный доступ для зарегистрированных пользователей
Л1.2	Страуструп Б.	Язык программирования С++ для профессионалов: практическое пособие	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2006	https://biblioclub.ru/index.php?page=book&id=234816 неограниченный доступ для зарегистрированных пользователей
Л1.3	Мейер Б.	Объектно-ориентированное программирование и программная инженерия: учебное пособие	Москва: Национальный Открытый Университет «ИНТУИТ», 2016	https://biblioclub.ru/index.php?page=book&id=429034 неограниченный доступ для зарегистрированных пользователей
Л1.4	Сорокин А. А.	Объектно-ориентированное программирование: учебное пособие: курс лекций	Ставрополь: Северо-Кавказский Федеральный университет (СКФУ), 2014	https://biblioclub.ru/index.php?page=book&id=457696 неограниченный доступ для зарегистрированных пользователей
Л1.5	Баранова, И. В., Баранов, С. Н., Баженова, И. В., Кучунова, Е. В., Толкач, С. Г.	Объектно-ориентированное программирование на С++: учебник	Красноярск: Сибирский федеральный университет, 2019	http://www.iprbookshop.ru/100067.html неограниченный доступ для зарегистрированных пользователей

5.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год	Колич-во
--	---------------------	----------	-------------------	----------

	Авторы, составители	Заглавие	Издательство, год	Колич-во
Л12.1	Зольников В. К., Машевич П. Р., Анциферова В. И., Литвинов Н. Н.	Программирование и основы алгоритмизации: учебное пособие	Воронеж: Воронежская государственная лесотехническая академия, 2011	https://biblioclub.ru/index.php?page=book&id=142309 неограниченный доступ для зарегистрированных пользователей
Л12.2	Агафонов Е. Д., Ващенко Г. В.	Прикладное программирование: учебное пособие	Красноярск: Сибирский федеральный университет (СФУ), 2015	https://biblioclub.ru/index.php?page=book&id=435640 неограниченный доступ для зарегистрированных пользователей
Л12.3		Программные продукты и системы: журнал	Тверь: Центрпрограммсистем, 2017	https://biblioclub.ru/index.php?page=book&id=459225 неограниченный доступ для зарегистрированных пользователей
Л12.4		Прикладное программирование на C/C++. С нуля до мультимедийных и сетевых приложений	Москва: СОЛОН-ПРЕСС, 2011	http://www.iprbookshop.ru/65139.html неограниченный доступ для зарегистрированных пользователей
Л12.5	Родыгин А. В.	Информационные технологии: алгоритмизация и программирование: учебное пособие	Новосибирск: Новосибирский государственный технический университет, 2017	https://biblioclub.ru/index.php?page=book&id=576499 неограниченный доступ для зарегистрированных пользователей
Л12.6	Зайцев М. Г.	Объектно-ориентированный анализ и программирование: учебное пособие	Новосибирск: Новосибирский государственный технический университет, 2017	https://biblioclub.ru/index.php?page=book&id=576800 неограниченный доступ для зарегистрированных пользователей

5.3 Профессиональные базы данных и информационные справочные системы

Портал сообщества Cppreference. Раздел справки по языку C++ - https://devdocs.io/cpp
Документация по Microsoft C++, C - https://docs.microsoft.com/ru-ru/cpp
Образовательный портал "Основы программирования на языках Си и C++ для начинающих" - http://cppstudio.com/
Официальная библиотека Crypto++ - https://www.cryptopp.com
Консультант +

5.4. Перечень программного обеспечения

MS Visual Studio
MS Visio

5.5. Учебно-методические материалы для студентов 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 (МОДУЛЯ)

Помещения для проведения всех видов работ, предусмотренных учебным планом, укомплектованы необходимой специализированной учебной мебелью и техническими средствами обучения. Для проведения лекционных занятий используется демонстрационное оборудование. Лабораторные занятия проводятся в компьютерных классах, рабочие места в которых оборудованы необходимыми лицензионными программными средствами и выходом в Интернет.

7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

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
ОК-3: способностью использовать основы экономических знаний в различных сферах деятельности			
З основные принципы структурного и объектно-ориентированного программирования	изучение основной и дополнительной литературы, лекционного материала, использование профессиональных баз данных для изучения методов структурного и объектно-ориентированного программирования, включая особенности работы с функциями, типизацию данных, указатели и особенности применения классов, объектов, экземпляров, скрытых указателей	полнота и содержательность ответа, обоснованность выбора метода структурного и/или объектно-ориентированного программирования, полнота приведенных примеров в ответах на вопросы опроса, теоретических вопросов на экзамене	Э (вопрос 1-60) О (вопрос 1-20)
У использовать методы структурного и объектно-ориентированного программирования	решение лабораторных и практико-ориентированных заданий, включая особенности работы с функциями, типизацию данных, указатели и особенности применения классов, объектов, экземпляров, скрытых указателей, использование методов программирования в курсовом проекте	правильность выполнения задания, актуальность выбора тематики курсового проекта, обоснованность применения выбранного метода программирования в курсовом проекте, наличие в программном коде авторских дополнений, умение отстаивать свою точку зрения	ПОЗЭ (задание 1-20) ЛЗ (задание 1-8). КП (задание 1-10)
В навыками разработки программного обеспечения на основе методов структурного и объектно-ориентированного программирования	решение лабораторных и практико-ориентированных заданий, включая особенности работы с функциями, типизацию данных, указатели и особенности применения классов, объектов, экземпляров, скрытых	объем и индивидуальность выполнения задания, курсового проекта с использованием методов структурного и объектно-ориентированного программирования	ПОЗЭ (задание 1-20) ЛЗ (задание 1-8). КП (задание 1-10)

	указателей, выполнение курсового проекта с использованием методов программирования		
ОПК-1: способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности			
З основы базовых алгоритмов решения задач профессиональной деятельности;	изучение основной и дополнительной литературы, лекционного материала, использование профессиональных баз данных для выбора алгоритма решения задачи программирования	полнота и содержательность ответа, обоснованность выбора базовых алгоритмов программирования в ответах на вопросы опроса, теоретических вопросов на экзамене	Э (вопрос 1-60) О (вопрос 1-4)
У применять методы проектирования и тестирования программного обеспечения;	решение лабораторных и практико-ориентированных заданий, выбор метода программирования в курсовом проекте	правильность выполнения заданий, сложность алгоритмических решений, обоснованность применения выбранного метода программирования в курсовом проекте	ПОЗЭ (задание 1-20) ЛЗ (задание 1-8) КП (задание 1-10)
В навыками тестирования программного обеспечения	авторское программное решение лабораторных, практико-ориентированных заданий, курсового проекта на основе базовых принципов разработки программного обеспечения	объем выполнения заданий, полнота анализа программных решений в курсовом проекте	ПОЗЭ (задание 1-20) ЛЗ (задание 1-8) КП (задание 1-10)
ОПК-3: способностью работать с компьютером как средством управления информацией, работать с информацией из различных источников, в том числе в глобальных компьютерных сетях			
З теорию информации, современные методы поиска информации в глобальных компьютерных сетях;	изучение основной и дополнительной литературы, лекционного материала, использование профессиональных баз данных для изучения основ информатики и программирования при обработке строк, объединений и структур, файлов, основ объектного подхода к разработке ПО, обработке ошибок и исключений	полнота и содержательность ответа, актуальность с точки зрения современных тенденций в информатике и программировании, правильность интерпретации основ программирования для обработки строк, объединений, структур, файлов, объектов, применения базовых конструкций обработки ошибок и исключений в	Э (вопрос 1-60) О (вопрос 1-12)

		ответах на вопросы опроса, теоретических вопросов на экзамене	
У использовать теорию управления информацией при обосновании выбора метода программирования;	решение лабораторных и практико-ориентированных заданий, включая обработку строк, объединений и структур, файлов, разработку классов, обработку ошибок и исключений, выполнение курсового проекта	правильность применения методов проектирования, и программирования в заданиях при обработке строк, объединений и структур, файлов, разработку классов, обработке ошибок и исключений, а также в курсовом проекте	ПОЗЭ (задание 1-20) ЛЗ (задание 1-8) КП (задание 1-10)
В навыками анализа фактов и теорий, связанными с управлением информацией при описании предметной области;	алгоритмическое решение лабораторных и, практико-ориентированных заданий в виде блок-схемы, результаты алгоритмизации в курсовом проекте	правильность авторских блок-схем, полнота и достоверность результатов в заданиях, а также в курсовом проекте	ПОЗЭ (задание 1-20) ЛЗ (задание 1-8) КП (задание 1-10)

О – опрос; Э – вопросы к экзамену; ПОЗЭ - практико-ориентированные задания к экзамену; ЛЗ – лабораторные задания; КП – курсовой проект

1.2 Шкалы оценивания:

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

- 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

Вопросы к экзамену

1. Понятие типов данных в языках программирования, классификация типов данных в C++.
2. Особенности представления базовых типов и операций над ними.
3. Рекомендации и правила выполнения операции преобразования базовых типов в C++.
4. Функции в C++. 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. Битовые поля.
21. Структуры и указатели.
22. Объединения.
23. Понятия и определения файлов и потоков, классификация файлов, стандартные функции по работе с файлами, основные алгоритмы работы с файлами.
24. Запись/чтение элементов структуры в файл.
25. Запись/чтение элементов массивов в файл.
26. Пузырьковая сортировка.
27. Сортировка вставками.
28. Сортировка выбором.
29. История ЯП. Классификация и область применения ЯП.
30. Парадигмы программирования. Процедурные языки. Аппликативные языки.
31. Языки логического программирования. Объектно-ориентированные языки.
32. Операция увеличения ++ и уменьшения --: «префиксная» форма, и «постфиксная» форма. Проверка четности числа. Логические операции. Библиотека conio.h. Применение метода system в проектах.
33. Программирование разветвляющихся алгоритмов. Условные операторы if. Оператор выбора switch.
34. Программирование циклических алгоритмов.
35. Компоновка файлов в решении. Сборщик «мусора».
36. Объектно-ориентированное программирование. Объект. Класс. Экземпляр.
37. Инкапсуляция.
38. Полиморфизм.
39. Секции private, protected, public. Указатель this.
40. Переопределение операций.
41. Передача параметров через консоль (int argc, char *argv[]). Режимы файла. Бинарные файлы. Решение. Проект.
42. Решение. Проект.
43. Директивы #include, #define, #undef, #ifdef, #else, #if, #pragma.
44. Макроимена. __FILE__ и __LINE__.
45. Конструктор. Инициализация переменных в конструкторе. Конструктор без параметров. Копирующий конструктор. Содержательный конструктор.
46. Константные ссылки и указатели. Константные методы классов. Перегрузка.
47. Понятие наследования. Виды наследования.

48. Множественное наследование.
49. Дружественные функции.
50. Дружественные члены класса (методы). Дружественные классы.
51. Виртуальные методы. Виртуальная функция. Виртуальные классы.
52. Абстрактные классы.
53. Виды ошибок. Исключительные ситуации. Обработка исключительных ситуаций.

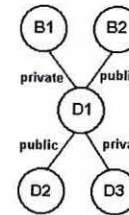
Операторы try и catch.

54. Шаблоны методов. Шаблоны классов. Назначение шаблонов.
55. Стандартная библиотека STL.
56. Статистические библиотеки.
57. Динамические библиотеки.
58. Библиотека "winsock.h".
59. Библиотека Crypto++.
60. Возможности WinAPI.

Практико-ориентированные задания к экзамену

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
11. Разработать класс строк, в котором подсчитайте количество слов, начинающихся на букву "a".
12. 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.
13. Реализуйте класс MyLine (линия), который является производным от классов Position (позиция: $x1, y1, x2, y2$) и Properties (свойства: цвет, ширина).
14. Разработайте класс TV, реализующий методы включения/выключения (on/off), переключения каналов (chanDown, chanUp), регулировки уровня громкости (volDown, volUp) и метод, отображающий все настройки (setting).
15. Разработайте класс «Заполнение строки случайными символами», содержащий метод по обработке символов через ASCII.
16. Реализовать класс «Символ в заданной позиции экрана».

17. Построить иерархию классов согласно схеме наследования (схема может отличаться)



18. 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).
19. Напишите класс Студент. Структуру класса продумайте самостоятельно.
20. Напишите класс Товар. Структуру класса продумайте самостоятельно.

Критерии оценивания:

- 84-100 (34-40 за ответ на 2 теоретических вопроса, 50-60 за решение 2 практико-ориентированных заданий) 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 (27-33 за ответ на 2 теоретических вопроса, 40-50 за решение 2 практико-ориентированных заданий) баллов (оценка «хорошо») – наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целью обучения, правильные действия по применению навыков 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 (20-26 за ответ на 2 теоретических вопроса, 30-40 за решение 2 практико-ориентированных заданий) 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 навыков и умений при решении практико-ориентированного задания;
- 0-49 (0-19 за ответ на 2 теоретических вопроса, 0-30 за решение 2 практико-ориентированных заданий) 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.

Вопросы для опроса

1. Классификация языков и систем программирования
2. Принципы выбора языка программирования для решения задачи
3. Основные технологии программирования
4. Современные тенденции в информатике и программировании для решения определенного класса заданий.
5. Современные подходы для выбора ИТ-стека в разработке ПО.
6. Алгоритм. Способы задания алгоритмов
7. Основные структуры алгоритма.
8. Правила составления блок-схем.

9. Преимущества методов структурного программирования
10. Недостатки методов структурного программирования
11. Правила комментирования программного кода.
12. Выделение идеи представленного решения
13. Возможности и функционал операторов ввода\вывода (cout и cin).
14. Функции библиотеки math.h.
15. Функции потокового ввода\вывода данных printf()
16. Ввода\вывода данных scanf().
17. Оператор выбора switch.
18. Работа с памятью.
19. Динамичность.
20. Обработка элементов массивов.

Критерии оценивания:

- 1 балла выставляется обучающемуся, если изложенный материал фактически верен и логически обоснован.
- 0 баллов, если ответ неверный.

Максимальное количество баллов: 20 баллов.

Лабораторные задания

Лабораторное задание №1

Решение задачи по математической обработке данных, применений тригонометрических функций. Дать геометрическое представление табулированной функции. Двумерную геометрическую реализацию обеспечить следующими функциями:

plot(x,y) – для построения точечного графика,
 bar(x,y) и line(x,y) – для построения столбчатой диаграммы,
 stem(x,y) – для построения точечного графика с визуализацией ординат значения функции

Лабораторное задание № 2

Решение разветвляющихся алгоритмов. Составить программу, которая запрашивает ввод трех значений температуры и проверяет, есть ли среди них температура таяния льда. Если такая температура введена, вывести на экран сообщение «Введена температура таяния льда», иначе «Такой температуры нет»

Решение циклических алгоритмов. Вывести на экран 3 раза слово «Hello». Особенность: одинаковые действия выполняются 3 раза. Напечатать столбиком все целые числа от a до b (значения a и b вводятся с клавиатуры).

Лабораторное задание № 3

Программирование с использованием функций и рекурсий. Перегрузка функций
 Вычислить факториал числа n.

Пусть имеются описания const n = 100; type vector = array[1..n] of real;
 Рекурсивно описать процедуру PrNeg(x), которая печатает отрицательные элементы вектора x.

Лабораторное задание № 4

Программирование с использованием одномерных массивов: поиск, замена и перестановка элементов.

Даны значения массива {ai}, i = 0, ..., 10 и переменной x. Составьте программу вычисления алгебраического многочлена 10-й степени по формуле Горнера:

$$a_{10}x^{10} + a_9x^9 + \dots + a_1x + a_0 = ((\dots(a_{10}x + a_9)x + a_8)x + \dots + a_1)x + a_0.$$

Дан вектор {z}, i = 1, ..., 50. Составьте программу ввода значений и вычисления длины этого вектора по следующей формуле:

$$L = \sqrt{z_1^2 + z_2^2 + \dots + z_{50}^2}.$$

Лабораторное задание № 5

Программирование с использованием динамических одномерных массивов.

Создать динамические массивы, используя указатели. Дан двумерный массив a, размером (n-m). Заполнить одномерный массив, найдя сумму положительных элементов в каждом столбце матрицы.

Создать динамические массивы, используя указатели. Дан двумерный массив a, размером (n-m). Заполнить одномерный массив, найдя произведение отрицательных элементов в каждом столбце матрицы.

Лабораторное задание № 6

Программирование с использованием динамических двумерных массивов: обработка элементов массива, диагонали, передача массива в функции.

Создать динамические массивы, используя указатели. Дан двумерный массив a, размером (n-m).

Заполнить одномерный массив, найдя количество отрицательных элементов в каждом столбце матрицы. Создать динамические массивы, используя указатели. Дан двумерный массив a, размером (n-m).

Заполнить одномерный массив, найдя количество положительных элементов в каждом столбце матрицы.

Лабораторное задание № 7 Программирование с использованием строк. Программирование с использованием структур

Определить комбинированный (структурный) тип для представления анкеты ребенка, состоящей из его имени, пола и роста. Ввести информацию по 20 детям. Вывести средний рост мальчиков.

Определить комбинированный (структурный) тип для представления информации по горным вершинам, состоящей из названия вершины и ее высоты. Ввести информацию по 20 вершинам. Вывести среднее значение высот всех 20 вершин. Далее вывести названия всех вершин ниже среднего.

Лабораторное задание № 8

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.

Написать функцию, которая читает данные о ноутбуках из файла note.txt в структуру приведенного вида. Написать функцию записывающую данные из структуры в конец бинарного файла. Структура бинарного файла: первые 2 байта(целое) – число записей в файле; далее записи в формате NOTEBOOK.

Написать программу, записывающую в файл данные лишь о тех ноутбуках, диагональ дисплея которых больше 11 дюймов.

Критерии оценивания:

- (для каждого задания):
- 10 б. – задание выполнено верно;
- 9-7 б. – при выполнении задания были допущены неточности, не влияющие на результат;
- 6-3 б. – при выполнении задания были допущены ошибки;
- 2 - 1 б. – при выполнении задания были допущены существенные ошибки;
- 0 б. – задание не выполнено.

Максимальное количество баллов - 80.

Курсовой проект

Перечень заданий на курсовой проект

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 болезнях.

5. «Разработка проекта (программного модуля) автоматизации аптеки»: В системе должны поддерживаться режимы поиска лекарств 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 стек ИТ-технологий, в том числе направленный на алгоритмы разработки веб-модулей, css-стилей, игр, графики, sms-систем, cm-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), защищенный программный код и т.д.

Требования к оформлению курсового проекта приведены в Приложении 2.

Критерии оценивания:

- 84-100 баллов (оценка «отлично») — 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 в полном объеме;

- 67-83 баллов (оценка «хорошо») — разработанный интерфейс программы и ее функциональные возможности соответствуют требованиям задания; текстовое описание составлено в полном объеме; программа работает со сбоями для некоторых типовых экспериментов, для которых она разрабатывалась, предусмотрена защита от ввода некорректных данных; обучающийся показал достаточное владение тематикой проекта, знание используемых компонентов, их свойств и специальных функций; изложенный материал в основном верен, наличие достаточно исчерпывающих знаний в рамках пройденной программы; достаточные действия по применению полученных умений и навыков на практике; усвоение основной и дополнительной литературы, работа с профессиональными базами данных в достаточном объеме;

- 50-66 баллов (оценка «удовлетворительно») — 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;

- 0-49 баллов (оценка «неудовлетворительно») — разработанный интерфейс программы и ее функциональные возможности не соответствуют требованиям задания; текстовое описание составлено не в полном объеме и не аккуратно; программа работает со сбоями, не предусмотрена защита от ввода некорректных данных; обучающийся показал очень слабые знания по тематике проекта; неправильные 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 защиты курсового проекта, экзамена.

Экзамен проводится по расписанию промежуточной аттестации в письменном виде. Количество вопросов в экзаменационном билете – 4. Проверка ответов и объявление результатов производится в день экзамена.

Результаты аттестации заносятся в экзаменационную ведомость и зачетную книжку студента. Студенты, не прошедшие промежуточную аттестацию по графику сессии, 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, 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.

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 с учетом индивидуальности и творческого решения алгоритмов. 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 курсового проекта.

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 скриншоты разработки проекта.

Курсовой проект выполняется с учетом приобретенных знаний по данной дисциплине и интереса обучающегося.

Основными этапами выполнения задания к курсовому проекту 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 решенных 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 правомочности 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 курсового проекта.

Проект 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 задания на курсовой проект.

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 курсовому проекту.

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 курсового проекта 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.