

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность: Ректор

Дата подписания: 12.11.2024 14:05:55

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

Платонова Т.К.

«25» июня 2024 г.

**Рабочая программа дисциплины**  
**Технологии разработки программного обеспечения в цифровой экономике**

Направление 38.03.05 Бизнес-информатика  
Направленность 38.03.05.02 Информационное и программное обеспечение бизнес-  
процессов в цифровой экономике

Для набора 2022 года

Квалификация  
Бакалавр

КАФЕДРА **Информационные технологии и программирование****Распределение часов дисциплины по семестрам**

Семестр (<Курс>.<Семестр на курсе>)	1 (1.1)		2 (1.2)		3 (2.1)		Итого	
	Неделя		16		16			
Вид занятий	УП	РП	УП	РП	УП	РП	УП	РП
Лекции	16	16	16	16	16	16	48	48
Лабораторные	32	32	32	32	32	32	96	96
Итого ауд.	48	48	48	48	48	48	144	144
Контактная работа	48	48	48	48	48	48	144	144
Сам. работа	60	60	96	96	96	96	252	252
Часы на контроль			36	36	36	36	72	72
Итого	108	108	180	180	180	180	468	468

**ОСНОВАНИЕ**

Учебный план утвержден учёным советом вуза от 25.06.2024 г. протокол № 18.

Программу составил(и): к.ф-м.н, доц., Карнаухов С.Н.

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

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

### 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1	изучение парадигм и методов разработки программного обеспечения и технологий программирования с помощью инструментов визуальной разработки программ.
-----	--

### 2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

**ПК-6: Способен разрабатывать и анализировать требования к программному обеспечению**

**В результате освоения дисциплины обучающийся должен:**

**Знать:**

методы программирования и методы разработки эффективных алгоритмов решения профессиональных задач (соотнесено с индикатором ПК-6.1).

**Уметь:**

применять программное обеспечение для решения профессиональных задач (соотнесено с индикатором ПК-6.2).

**Владеть:**

основами применения инструментария программирования для решения профессиональных задач (соотнесено с индикатором ПК-6.3).

### 3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

#### Раздел 1. Развитие технологий и методов программирования

№	Наименование темы / Вид занятия	Семестр / Курс	Часов	Компетенции	Литература
1.1	"Общие принципы разработки программного обеспечения" Классификация программных продуктов. Жизненный цикл ПО. Стадии разработки ПО. Документирование ПО / Лек /	1	4	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.2	"Общие принципы разработки программного обеспечения" C#. Приложение WPF. Компонновка. Grid, GridSplitter, StackPanel, DockPanel, WrapPanel, Canvas в Code - OSS / Лаб /	1	10	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.3	"Проектирование программного обеспечения" Методы проектирования. Разработка структурной и функциональной схем. Проектирование программного обеспечения, основанное на декомпозиции данных в Draw.io. Case-технологии, основанные на структурных методологиях анализа. Разработка структуры программного обеспечения при объектном подходе. Определение отношений между объектами. Проектирование классов. Реорганизация проекта. / Лек /	1	6	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.4	"Проектирование программного обеспечения" C#. Приложение WPF. Элементы управления. ListBox, ComboBox, ListView. TabControl. Меню. ToolBar. TreeView. DataGrid. Calendar.. Image. InkCanvas в Code - OSS / Лаб /	1	10	ПК-6	Л1.1, Л1.2, Л2.1, Л2.2, Л2.3
1.5	Case-технологии, основанные на структурных методологиях анализа. Разработка структуры программного обеспечения при объектном подходе. Определение отношений между объектами. Проектирование классов. Реорганизация проекта. / Лек /	1	6	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.6	"Практическое программирование" Арифметические функции Найти значение алгебраического выражения, соответствующего варианту задания. Вывести результаты на печать. Все результаты выводить в развернутом виде (например: «Сумма чисел А и В равна 3.7854») Значения вводимых величин должны иметь не менее четырех значащих цифр и задаются студентом самостоятельно. / Лаб /	1	12	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.7	Стиль программирования. Выбор языка программирования Виды ошибок. Основные принципы отладки ПС Основные принципы организации тестирования ПС Виды программных документов Обеспечение функциональности, надежности и качества ПС. Технологии оценки качества ПС Обеспечение качества программного обеспечения	1	60	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3

	Аттестация программного средства / Ср /				
1.8	/ Зачёт /	1	0	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.9	"Практическое программирование" Общие принципы устройства И функционирования ATMEGA AVR. Общее устройство, организация памяти, тактирование, сброс. / Лек /	2	6	ПК-6	Л1.1, Л1.2, Л1.3, Л2.2, Л2.3
1.10	"Практическое программирование" Описание классов должны быть вынесены в отдельный заголовочный файл. Реализация функций-элементов класса должна быть написана в отдельном модуле, а основную программу, иллюстрирующую применение всех методов вашего класса, следует реализовать еще одним модулем. Во всех заданиях предусмотрите конструкторы с аргументами по умолчанию, а также дружественную перегруженную операцию вывода в поток и чтения из потока. Помните, что в каждом классе должны быть предусмотрены константные функции get / Лаб /	2	8	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.11	«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++. / Лек /	2	6	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.12	"Практическое программирование" Знакомство с периферийными устройствами Порты ввода-вывода Аналогово-цифровой преобразователь. Последовательные порты UART Интерфейс SPI. Интерфейс TWI (I2C) / Лаб /	2	8	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.13	Практическое программирование. Знакомство с периферийными устройствами Порты ввода-вывода Аналогово-цифровой преобразователь. Последовательные порты UART Интерфейс SPI. Интерфейс TWI (I2C) / Ср /	2	46	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.14	«Функции на языке C++» Разработка проектов в Code - OSS по решению классических задач на нахождение факториалов и сумм числовых последовательностей. Алгоритмическая реализация рекурсии. / Лаб /	2	4	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1
1.15	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 / Лек /	2	4	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.16	«Типы данных в языке C++» Разработка программного решения Code - OSS 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 Code - OSS. / Лаб /	2	6	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.17	Разработка программных продуктов Требования к современным технологиям Общие принципы разработки программных средств Разработка структуры программы и модульное программирование Разработка программного модуля. Структурное программирование / Ср /	2	50	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.18	Отладка приложений: в задании, содержащих те или иные ошибки (логические, ошибки синтаксиса и т.п.). Задача состоит в следующем: используя средства отладчика среды, обнаружить все ошибки и сделать программу работоспособной с использованием	2	6	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3

	Code - OSS / Лаб /				
1.19	/ Экзамен /	2	36	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
<b>Раздел 2. Практическое программирование</b>					
№	Наименование темы / Вид занятия	Семестр / Курс	Часов	Компетенции	Литература
2.1	Тема 2.1 "Разработка программных продуктов" Языки программирования. Стили программирования. Case-средства разработки ПО. Эффективность. Оценка качества ПО / Лек /	3	2	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.2	Общие принципы разработки программных средств Разработка структуры программы и модульное программирование Разработка программного модуля. Структурное программирование / Лаб /	3	4	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.3
2.3	"Отладка, тестирование и сопровождение программных продуктов" Классификация ошибок ПО. Методы отладки ПО. Методы тестирования ПО. Сопровождение программ. Интеллектуальная собственность на ПО. / Лек /	3	2	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.4	«Массивы: одномерные и двумерные» Массивы: одномерные массивы, задачи поиска, замены и перестановок элементов массива, задачи сортировок элементов массива. / Лаб /	3	2	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.5	Методы тестирования ПО. Модульное тестирование Интеграционное тестирование Системное тестирование Приемочные испытания / Лек /	3	2	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.6	"Разработка программных продуктов" С#. Приложение WPF. Ресурсы. Стили. Анимация. / Лаб /	3	4	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.7	"Отладка, тестирование и сопровождение программных продуктов" С#. Приложение WPF. Привязка. Шаблоны. Code - OSS. / Лаб /	3	4	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.8	Методы объектно-ориентированного проектирования ПО. UML. Диаграммы прецедентов. Диаграммы деятельности. UML. Диаграммы классов. Диаграммы последовательности действий. Диаграммы компонентов. Коллективная разработка ПО. Организация работ. / Ср /	3	12	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.9	Диаграммы прецедентов. Диаграммы деятельности. UML. Диаграммы классов. Диаграммы последовательности действий. Диаграммы компонентов. / Лаб /	3	6	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.10	Отладка, тестирование и сопровождение программных продуктов Ручной контроль Классификация ошибок / Ср /	3	16	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.11	Отладка, тестирование и сопровождение программных продуктов Ручной контроль Классификация ошибок с использованием Code - OSS. / Лаб /	3	2	ПК-6	Л1.1, Л1.2, Л1.3, Л2.2, Л2.3
2.12	«Ссылки и указатели. Динамическая память» Реализация в Code - OSS проекта по обработке элементов динамического одномерного массива: - Поместить положительные элементы массива X (n) в начало массива Y, а следом – его отрицательные элементы. - На заданном отрезке, с заданным шагом изменения аргумента (x) вычислить и поместить в массив H (n) значения функции $e-x \sin(6x)$ . / Лаб /	3	2	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.13	«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. / Лек /	3	2	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.14	«Символьные данные и строки» Реализация в Code - OSS проектов: - подсчет количества слов в предложении. Предварительно привести строку к нормированному виду, удалить все лишние	3	2	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1

	пробелы; - нахождение короткого слово в строке / Лаб /				
2.15	«Символьные данные и строки» Реализация в Code - OSS проектов: - обработка строки, изображающей арифметическое выражение вида «<цифра>±<цифра>±...±<цифра>», где на месте знака операции «±» находится символ «+» или «-». / Лаб /	3	2	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.16	Реализовать алгоритм по нахождению в строке указанную подстроку и замене ее на новую, ввод данных через диалоговый режим на основе API языка C/C++, описанного в Справочном разделе <a href="https://devdocs.io/cpp">https://devdocs.io/cpp</a> / Ср /	3	20	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.17	«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. / Лек /	3	2	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.18	«Структуры и объединения» Обработка элементов struct, union. Программная реализация в Code - OSS структур данных, возможности ввода элементов вручную и инициализация в коде. / Лаб /	3	2	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2
2.19	«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, / Ср /	3	28	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.20	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. / Лек /	3	2	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.21	«Файлы. Функции по работе с файлами» Реализация в Code - OSS программного решения по копированию файлов в директориях операционной системы. Нахождение статистики лексических форм в файлах. Запись структуры в файл, возможность добавления данных, возможность перезаписи файлов. / Лаб /	3	2	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.22	Файлы. Функции по работе с файлами» Прямой доступ к данным файла. Работа с элементами файлов. Передача файлов через командную строку операционной системы, обработка количества входных параметров. / Лек /	3	2	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.23	«Развитие языков программирования» Разработка проекта в Code - OSS по обработке логических и синтаксических выражений на разных языках программирования (псевдо компилятор). / Лек /	3	2	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.24	Курсовой проект. Перечень тем представлен в приложении 1 к рабочей программе дисциплины. / Ср /	3	20	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.25	/ Экзамен /	3	36	ПК-6	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3

#### 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,	Заглавие	Издательство, год	Колич-во
---------	----------	-------------------	----------

	Авторы,	Заглавие	Издательство, год	Колич-во
Л1.1	Мишова В. В.	Технологии программирования: практикум	Кемерово: Кемеровский государственный институт культуры (КемГИК), 2016	<a href="https://biblioclub.ru/index.php?page=book&amp;id=472686">https://biblioclub.ru/index.php?page=book&amp;id=472686</a> неограниченный доступ для зарегистрированных пользователей
Л1.2	Горелов, С. В., Лукьянова, П. Б.	Современные технологии программирования: разработка Windows-приложений на языке C#. В 2 томах. Т.1: учебник	Москва: Прометей, 2019	<a href="https://www.iprbookshop.ru/94532.html">https://www.iprbookshop.ru/94532.html</a> неограниченный доступ для зарегистрированных пользователей
Л1.3	Зайцев, М. Г.	Современные технологии программирования: практикум	Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2008	<a href="https://www.iprbookshop.ru/55460.html">https://www.iprbookshop.ru/55460.html</a> неограниченный доступ для зарегистрированных пользователей

### 5.2. Дополнительная литература

	Авторы,	Заглавие	Издательство, год	Колич-во
Л2.1		Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика: журнал	Астрахань: Астраханский государственный технический университет (АГТУ), 2017	<a href="https://biblioclub.ru/index.php?page=book&amp;id=459086">https://biblioclub.ru/index.php?page=book&amp;id=459086</a> неограниченный доступ для зарегистрированных пользователей
Л2.2	Дерябкин, В. П., Козлов, В. В.	Проектирование информационных систем по методологии UML с использованием Qt-технологии программирования: учебное пособие	Самара: Самарский государственный технический университет, ЭБС АСВ, 2017	<a href="https://www.iprbookshop.ru/83601.html">https://www.iprbookshop.ru/83601.html</a> неограниченный доступ для зарегистрированных пользователей
Л2.3	Лебеденко, Л. Ф., Моренкова, О. И.	Технологии программирования: учебно-методическое пособие	Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2019	<a href="https://www.iprbookshop.ru/102141.html">https://www.iprbookshop.ru/102141.html</a> неограниченный доступ для зарегистрированных пользователей

### 5.3 Профессиональные базы данных и информационные справочные системы

ИСС "КонсультантПлюс"

ИСС "Гарант"<http://www.internet.garant.ru/>

Информационная система "Единое окно доступа к образовательным ресурсам". <http://window.edu.ru/>

Бесплатная база данных ГОСТ. <https://docplan.ru/>

### 5.4. Перечень программного обеспечения

Операционная система РЕД ОС

Code - OSS

Draw.io

### 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

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.

**7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

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



## 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-6: способен разрабатывать и анализировать требования к программному обеспечению			
З методы программирования и методы разработки эффективных алгоритмов решения профессиональных задач	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 зачете/экзамене	З (1 сем(1-15) Э (2 сем (1-15) 3 сем (1-23)) О (1 сем (1-16) 2 сем (1-16) 3 сем (1-16))
У применять программное обеспечение для решения профессиональных задач	решение лабораторных, практико-ориентированных заданий, выбор метода программирования	правильность выполнения заданий, сложность алгоритмических решений, обоснованность применения выбранного метода программирования в курсовом проекте	ПОЗЗ (1 сем (1-5) ПОЗЭ ( 2 сем 1-5, 3 сем 1-5) ЛЗ (1 сем (1-6) 2 сем (1-6) 3 сем (1-6). КП (1-20)
В основами применения инструментария программирования для решения профессиональных задач	авторское программное решение лабораторных, практико-ориентированных заданий, на основе базовых принципов разработки программного обеспечения	объем выполнения заданий, полнота анализа программных решений в курсовом проекте	ПОЗЗ (2 сем (1-5) 3 сем (1-5) 4 сем (1-5)) ПОЗЭ (1-5) ЛЗ (2 сем (1-6) 3 сем (1-6) 4 сем (1-3) 5 сем (1-3). КП (1-20)

*О – опрос; З – вопросы к зачету; ПОЗЗ - практико-ориентированные задания к зачету; 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:

##### 1 семестр:

50-100 баллов (зачет)

0-49 баллов (незачет)

##### 2, 3 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

#### Вопросы к зачету

##### 1 семестр

- 1) Программные продукты и их основные характеристики.
- 2) Классификация ПО.
- 3) Жизненный цикл ПО
- 4) Стадии разработки ПО.
- 5) Документирование ПО.

- 6) Методы структурного проектирования ПО.
- 7) Методы объектно-ориентированного проектирования ПО.
- 8) UML. Диаграммы прецедентов. Диаграммы деятельности.
- 9) UML. Диаграммы классов. Диаграммы последовательности действий. Диаграммы компонентов.
- 10) Стилль программирования.
- 11) Языки программирование.
- 12) Эффективность и оптимизация ПО.
- 13) Критерии качества ПО.
- 14) Ошибки ПО.
- 15) Отладка ПО.

### Практико-ориентированные задания к зачету

#### 1 семестр

1. В одномерном массиве переставьте элементы таким образом, чтобы на 5-ом месте стояло бы наибольшее, а на 6-ом — наименьшее число.
2. Подсчитать количество слов в предложении, при условии, что слова разделены пробелом (но могут встречаться и несколько пробелов), в конце может быть . ! ?
3. Найти слова, содержащие подстроку “app”.
4. Написать функцию, которая вычисляет значение  $a^b$ . Числа  $a$  и  $b$  целые, могут быть как положительными, так и отрицательными.
5. Написать функцию по вычислению дискриминанта квадратного уравнения

#### **Критерии оценивания:**

- 50-100 баллов («зачет»): – 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 («незачет») – 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.

### Вопросы к экзамену

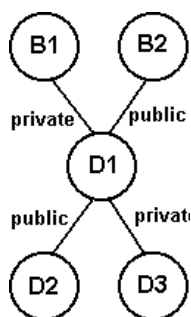
#### 2 семестр

1. Виды наследования.
2. Множественное наследование.
3. Дружественные функции.
4. Дружественные члены класса (методы).
5. Дружественные классы.
6. Методы тестирования ПО.
7. Сопровождение ПО.
8. Защита программного кода ПО.
9. CASE-технологии.
10. Коллективная разработка ПО. Организация работ.
11. Преимущества объектно-ориентированных языков.
12. Определение класса. Понятие объекта и экземпляра класса.
13. Указатель this.
14. Секции private, protected, public.
15. Опишите понятие конструктора. Опишите понятие деструктора

### Практико-ориентированные задания к экзамену

#### 2 семестр

1. Реализовать класс «Символ в заданной позиции экрана».
2. Построить иерархию классов согласно схеме наследования (схема может отличаться)



3. 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).
4. Напишите класс Студент. Структуру класса продумайте самостоятельно.
5. 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.

### Вопросы к экзамену

#### 3 семестр

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. Программирование разветвляющихся алгоритмов. Условные операторы if. Оператор выбора switch .
6. Функции библиотеки math.h. Линейный алгоритм.
7. Программирование циклических алгоритмов.
8. Программирование с использованием одномерных массивов.
9. Программирование с использованием двумерных массивов.
10. Объектно-ориентированное программирование. Объект. Класс. Экземпляр.
11. Инкапсуляция.
12. Полиморфизм.
13. Конструктор. Инициализация переменных в конструкторе. Конструктор без параметров. Копирующий конструктор. Содержательный конструктор.
14. Константные ссылки и указатели. Константные методы классов. Перегрузка.
15. Понятие наследования. Виды наследования.
16. Множественное наследование.
17. Дружественные функции.

18. Дружественные члены класса (методы). Дружественные классы.
19. Методы тестирования ПО.
20. Сопровождение ПО.
21. Защита программного кода ПО.
22. CASE-технологии.
23. Коллективная разработка ПО. Организация работ.

### **Практико-ориентированные задания к экзамену**

#### 3 семестр

1. Разработать класс строк, в котором подсчитайте количество слов, начинающихся на букву "Б".
2. Разработайте класс Car (автопарк), у которого определены приватные переменные name (вид транспорта) и price (первоначальная стоимость), 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.

#### **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.

### **Вопросы для опроса**

#### 1 семестр

1. Принципы выбора языка программирования для решения задачи
2. Современные тенденции в информатике и программировании для решения определенного класса заданий.
3. Современные методы программирования
4. Современные системы программирования
5. Современные подходы для выбора ИТ-стека в разработке ПО.
6. Способы задания алгоритма
7. Основные структуры алгоритма
8. Правила составления блок-схем.
9. Преимущества методов структурного программирования
10. Правила комментирования программного кода.
11. Выделение идеи представленного решения
12. Возможности и функционал операторов ввода\вывода (cout и cin).
13. Функции библиотеки math.h.

14. Функции потокового ввода\вывода данных printf() и scanf().
15. Современный язык программирования
16. Аргументы при выборе языка разработки.

**Критерии оценивания:**

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

Максимальное количество баллов за семестр: 16 баллов.

2 семестр

1. Оператор выбора switch.
2. Работа с памятью. Динамичность.
3. Обработка элементов массивов.1
4. Сложность алгоритмического решения
5. Особенности ручного тестирования. Контрольные точки. Точки останова.
6. Обоснование выбранных типов данных в проекте.
7. Преимущества объектно-ориентированных языков.
8. Определение класса. Понятие объекта и экземпляра класса.
9. Указатель this.
10. Секции private, protected, public.
11. Опишите понятие конструктора.
12. Опишите понятие деструктора.
13. Объект. Класс. Экземпляр
14. Полиморфизм.
15. Наследование
16. Инкапсуляция

**Критерии оценивания:**

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

- 0 баллов, если ответ неверный.

Максимальное количество баллов за семестр: 16 баллов.

3 семестр

1. Инкапсуляция.
2. Дружественные функции.
3. Особенности, использование.
4. Дружественные классы.
5. Особенности, использование.
6. Особенности абстракции в ООП.
7. Роль виртуальных классов в ООП.
8. Приложение WPF.
9. компоновка.
10. Периферийными устройствами.
11. Порты ввода-вывода
12. Средства отладчиков среды.
13. Виды тестирования.
14. Ручное тестирование
15. Тестирование программы
16. Отладка программы

**Критерии оценивания:**

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

- 0 баллов, если ответ неверный.

Максимальное количество баллов за семестр: 16 баллов.

## Лабораторные задания

### 1 семестр

#### Лабораторное задание № 1

"Общие принципы разработки программного обеспечения"

С#. Приложение WPF. Компонировка. Grid, GridSplitter, StackPanel в Code - OSS

#### Лабораторное задание № 2

"Проектирование программного обеспечения"

С#. Приложение WPF. Элементы управления. ListBox, ComboBox, ListView.

TabControl. в Code - OSS

#### Лабораторное задание № 3

"Практическое программирование" Арифметические функции

Найти значение алгебраического выражения, соответствующего варианту задания. Вывести результаты на печать. Все результаты выводить в развернутом виде (например: «Сумма чисел А и В равна 3.7854»). Значения вводимых величин должны иметь не менее четырех значащих цифр и задаются студентом самостоятельно.

#### Лабораторное задание №.4

"Практическое программирование" Описание классов должны быть вынесены в отдельный заголовочный файл. Реализация функций-элементов класса должна быть написана в отдельном модуле, а основную программу, иллюстрирующую применение всех методов вашего класса, следует реализовать еще одним модулем. Во всех заданиях предусмотрите конструкторы с аргументами по умолчанию, а также дружественную перегруженную операцию вывода в поток и чтения из потока. Помните, что в каждом классе должны быть предусмотрены константные функции get.

#### Лабораторное задание № 5

"Практическое программирование" Знакомство с периферийными устройствами

Порты ввода-вывода

Аналогово-цифровой преобразователь. Последовательные порты UART

Интерфейс SPI.

Интерфейс TWI (I2C).

#### Лабораторное задание №.6

Отладка приложений: в задании, содержащих те или иные ошибки (логические, ошибки синтаксиса и т.п.). Задача состоит в следующем: используя средства отладчика среды, обнаружить все ошибки и сделать программу работоспособной с использованием Code - OSS.

### **Критерии оценивания:**

- (для каждого задания):

14 б. – задание выполнено верно;

13-9 б. – при выполнении задания были допущены неточности, не влияющие на результат;

8-5 б. – при выполнении задания были допущены ошибки;

4 - 1 б. – при выполнении задания были допущены существенные ошибки;

0 б. – задание не выполнено.

Максимальное количество баллов за семестр - 84.

### 2 семестр

#### Лабораторное задание № 1

Общие принципы разработки программных средств

Разработка структуры программы и модульное программирование

Разработка программного модуля. Структурное программирование

#### Лабораторное задание № 2

Тема 2.1 "Разработка программных продуктов"

С#. Приложение WPF. Ресурсы. Стили. Анимация.

### Лабораторное задание № 3

"Отладка, тестирование и сопровождение программных продуктов"  
С#. Приложение WPF. Привязка. Шаблоны. Code - OSS.

### Лабораторное задание № 4

Диаграммы прецедентов. Диаграммы деятельности.  
UML. Диаграммы классов.. С использованием Rational Architect.

### Лабораторное задание № 5

UML. Диаграммы последовательности действий. Диаграммы компонентов.

### Лабораторное задание № 6

Отладка, тестирование и сопровождение программных продуктов Ручной контроль Классификация ошибок с использованием Code - OSS.

### **Критерии оценивания:**

- (для каждого задания):

14 б. – задание выполнено верно;

13-9 б. – при выполнении задания были допущены неточности, не влияющие на результат;

8-5 б. – при выполнении задания были допущены ошибки;

4 - 1 б. – при выполнении задания были допущены существенные ошибки;

0 б. – задание не выполнено.

Максимальное количество баллов за семестр - 84.

### 3 семестр

#### Лабораторное задание № 1

"Общие принципы разработки программного обеспечения"  
Си ++. Компоновка. DockPanel, WrapPanel, Canvas в Code – OSS.

#### Лабораторное задание № 2

"Проектирование программного обеспечения"  
Си++. Элементы управления. Меню. ToolBar. TreeView. DataGrid. Calendar.. Image. InkCanvas в Code – OSS.

#### Лабораторное задание № 3

"Практическое программирование" Арифметические функции  
Найти значение разности выражения, соответствующего варианту задания. Вывести результаты на печать. Все результаты выводить в развернутом виде (например: «Разность чисел А и В равна 3.7854»). Значения вводимых величин должны иметь не менее четырех значащих цифр и задаются студентом самостоятельно.

#### Лабораторное задание № 4

"Общие принципы разработки программного обеспечения"  
Си++. Компоновка. Grid, GridSplitter, StackPanel, DockPanel, WrapPanel, Canvas в Code – OSS.

#### Лабораторное задание № 5

"Проектирование программного обеспечения"  
Си++. Элементы управления. ListBox, ComboBox, ListView. TabControl. Меню. в Code – OSS.

#### Лабораторное задание № 6

"Практическое программирование" Арифметические функции  
Найти значение произведения выражения, соответствующего варианту задания. Вывести результаты на печать. Все результаты выводить в развернутом виде (например: «Произведение чисел А и В равно 3.7854»). Значения вводимых величин должны иметь не менее четырех значащих цифр и задаются студентом самостоятельно.

### **Критерии оценивания:**

- (для каждого задания):

14 б. – задание выполнено верно;

13-9 б. – при выполнении задания были допущены неточности, не влияющие на результат;

8-5 б. – при выполнении задания были допущены ошибки;

4 - 1 б. – при выполнении задания были допущены существенные ошибки;

0 б. – задание не выполнено.

Максимальное количество баллов за семестр - 84.

### **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 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.

11. Применение современных ИТ в работе администратора.

12. Применение современных ИТ в работе риэлтора.

13. Применение современных ИТ в работе менеджера по закупкам

14. Применение современных ИТ в работе менеджера по продажам

15. Применение современных ИТ в работе менеджера по услугам

16. Применение современных ИТ в работе товароведа

17. Применение современных ИТ в работе электронного бизнеса

18. Применение современных ИТ в работе разработчика ИТ

19. Применение современных ИТ в работе бизнес-аналитика

20. Применение современных ИТ в работе мерчендайзера

Обучающимся самостоятельно может быть выбран 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), 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 зачета (1 семестр), экзамена (2, 3 семестр), курсового проекта.

Зачет проводится по окончании теоретического обучения до начала экзаменационной сессии. 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 сессии, должны ликвидировать задолженность в установленном порядке.

## 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, лабораторных занятиях, должны быть изучены студентами в ходе самостоятельной работы. Контроль самостоятельной работы студентов над учебной программой курса осуществляется в ходе занятий методом устного опроса и посредством 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 скрины 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 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 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.

#### 5.1 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.

1. Список использованных источников.
2. 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.