

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность: Ректор

Дата подписания: 21.11.2024 11:31:14

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 и методы программирования

Направление 10.03.01 "Информационная безопасность"

Направленность 10.03.01.02 Организация и технологии защиты информации (по отрасли или в сфере профессиональной деятельности)

Для набора 2021 года

Квалификация
Бакалавр

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

Семестр (<Курс>.<Семестр на курсе>)	2 (1.2)		3 (2.1)		4 (2.2)		Итого	
	Неделя		16		16			
Вид занятий	УП	РП	УП	РП	УП	РП	УП	РП
Лекции	32	32	32	32	16	16	80	80
Лабораторные	32	32	48	48	32	32	112	112
Итого ауд.	64	64	80	80	48	48	192	192
Контактная работа	64	64	80	80	48	48	192	192
Сам. работа	8	8	28	28	60	60	96	96
Часы на контроль			36	36	36	36	72	72
Итого	72	72	144	144	144	144	360	360

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

ОПК-7: Способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности;

В результате освоения дисциплины обучающийся должен:

Знать:
методы программирования и методы разработки эффективных алгоритмов решения профессиональных задач (соотнесено с индикатором ОПК-7.1).
Уметь:
выбирать необходимое программное обеспечение для создания информационных систем (соотнесено с индикатором ОПК-7.2);
Владеть:
основами применения инструментария программирования для решения профессиональных задач (соотнесено с индикатором ОПК-7.3).

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 /	2	4	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.2	"Общие принципы разработки программного обеспечения" С#. Приложение WPF. Компоновка. Grid, GridSplitter, StackPanel, DockPanel, WrapPanel, Canvas в visual studio code / Лаб /	2	2	ОПК-7	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 /	2	6	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.4	"Проектирование программного обеспечения" С#. Приложение WPF. Элементы управления. ListBox, ComboBox, ListView. TabControl. Меню. ToolBar. TreeView. DataGrid. Calendar.. Image. InkCanvas в visual studio code / Лаб /	2	2	ОПК-7	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 /	2	6	ОПК-7	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 /	2	4	ОПК-7	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.	2	2	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3

	Технологии оценки качества ПС Обеспечение качества программного обеспечения Аттестация программного средства / Ср /				
1.8	"Практическое программирование" Общие принципы устройства И функционирования ATMEGA AVR. Общее устройство, организация памяти, тактирование, сброс. / Лек /	2	6	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.9	"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	4	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.10	«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	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.11	"Практическое программирование" Знакомство с периферийными устройствами Порты ввода-вывода Аналогово-цифровой преобразователь. Последовательные порты UART Интерфейс SPI. Интерфейс TWI (I2C) / Лаб /	2	4	ОПК-7	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) / Ср /	2	4	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.13	«Функции на языке C++» Разработка проектов в Visual Studio по решению классических задач на нахождение факториалов и сумм числовых последовательностей. Алгоритмическая реализация рекурсии. / Лаб /	2	4	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1
1.14	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	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.15	«Типы данных в языке C++» Разработка программного решения visual studio code 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 / Лаб /	2	6	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.16	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	2	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.17	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 visual studio code / Лаб /	2	6	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
1.18	/ Зачёт /	2	0	ОПК-7	Л1.1, Л1.2, Л1.3,

Л2.1, Л2.2, Л2.3					
Раздел 2. Практическое программирование					
№	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	8	ОПК-7	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	ОПК-7	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	8	ОПК-7	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	6	ОПК-7	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	8	ОПК-7	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	6	ОПК-7	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. visual studio code / Лаб /	3	8	ОПК-7	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	ОПК-7	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. С использованием Rational Architect. / Лаб /	3	8	ОПК-7	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	ОПК-7	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 visual studio code / Лаб /	3	8	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.2, Л2.3
2.12	«Ссылки и указатели. Динамическая память» Реализация в Visual Studio 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 N (n) значения функции e-x sin(6x). / Лаб /	3	8	ОПК-7	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	8	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.14	/ Экзамен /	3	36	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.15	«Символьные данные и строки» Реализация в Visual Studio проектов: - подсчет количества слов в предложении. Предварительно привести строку к нормированному виду,	4	8	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1

	удалить все лишние пробелы; - нахождение короткого слово в строке / Лаб /				
2.16	«Символьные данные и строки» Реализация в visual studio code проектов: - обработка строки, изображающей арифметическое выражение вида «<цифра>±<цифра>±...±<цифра>», где на месте знака операции «±» находится символ «+» или «-». / Лаб /	4	8	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.17	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 https://devdocs.io/cpp / Ср /	4	32	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.18	«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 /	4	2	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.19	«Структуры и объединения» Обработка элементов struct, union. Программная реализация в visual studio code структур данных, возможности ввода элементов вручную и инициализация в коде. / Лаб /	4	8	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2
2.20	«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 /	4	28	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.21	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 /	4	6	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.22	«Файлы. Функции по работе с файлами» Реализация в visual studio code 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 /	4	8	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.23	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 /	4	4	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.24	«Развитие языков программирования» Разработка проекта в visual studio code по обработке логических и синтаксических выражений на разных языках программирования (псевдо компилятор). / Лек /	4	4	ОПК-7	Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3
2.25	/ Экзамен /	4	36	ОПК-7	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	https://biblioclub.ru/index.php?page=book&id=472686 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	https://www.iprbookshop.ru/94532.html неограниченный доступ для зарегистрированных пользователей
Л1.3	Зайцев, М. Г.	Современные технологии программирования: практикум	Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2008	https://www.iprbookshop.ru/55460.html 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	https://biblioclub.ru/index.php?page=book&id=459086 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	https://www.iprbookshop.ru/83601.html 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	https://www.iprbookshop.ru/102141.html неограниченный доступ для зарегистрированных пользователей

5.3 Профессиональные базы данных и информационные справочные системы

Консультант +

Портал сообщества Cppreference. Раздел справки по языку C++ - <https://devdocs.io/cpp>

Образовательный портал "Основы программирования на языках Си и C++ для начинающих" - <http://cppstudio.com/>

Официальная библиотека Crypto++ - <https://www.cryptopp.com>

5.4. Перечень программного обеспечения

Операционная система РЕД ОС

visual studio code

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*
ОПК -7 – способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности			
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 (2 сем(1-15) Э (3 сем (1-15) 4 сем (1-26)) О (2 сем (1-16) 3 сем (1-16) 4 сем (1-16))
У выбирать необходимое программное обеспечение для создания информационных систем	решение лабораторных, практико-ориентированных заданий, выбор метода программирования	правильность выполнения заданий, сложность алгоритмических решений, обоснованность применения выбранного метода программирования в курсовом проекте	ПОЗЗ (2 сем (1-5) ПОЗЭ (3 сем 1-5, 4 сем 1-5) ЛЗ (2 сем (1-6) 3 сем (1-6) 4 сем (1-6)).
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))

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

1.2 Шкалы оценивания:

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

2 семестр:

50-100 баллов (зачет)

0-49 баллов (незачет)

3, 4 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

2 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

2 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

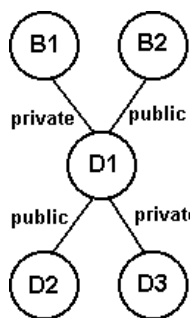
3 семестр

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

3 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

4семестр

1. История ЯП. Классификация и область применения ЯП.
2. Парадигмы программирования. Процедурные языки. Аппликативные языки.
3. Языки логического программирования. Объектно-ориентированные языки.
4. Стандартизация языков программирования.
5. Среда проектирования VisualStudio.NET.
8. Программирование разветвляющихся алгоритмов. Условные операторы if. Оператор выбора switch .
9. Функции библиотеки math.h. Линейный алгоритм.
10. Программирование циклических алгоритмов.
11. Программирование с использованием одномерных массивов.
12. Программирование с использованием двумерных массивов.
13. Объектно-ориентированное программирование. Объект. Класс. Экземпляр.
14. Инкапсуляция.
15. Полиморфизм.
16. Конструктор. Инициализация переменных в конструкторе. Конструктор без параметров. Копирующий конструктор. Содержательный конструктор.
17. Константные ссылки и указатели. Константные методы классов. Перегрузка.
18. Понятие наследования. Виды наследования.
19. Множественное наследование.
20. Дружественные функции.
21. Дружественные члены класса (методы). Дружественные классы.

22. Методы тестирования ПО.
23. Сопровождение ПО.
24. Защита программного кода ПО.
25. CASE-технологии.
26. Коллективная разработка ПО. Организация работ.

Практико-ориентированные задания к экзамену

4 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.

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 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 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 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.

Вопросов для опроса

2 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.

3 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.

4 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 4 семестр: 16 баллов.

Лабораторные задания

2 семестр

Лабораторное задание № 1

"Общие принципы разработки программного обеспечения"

C#. Приложение WPF. Компоновка. Grid, GridSplitter, StackPanel в Visual studio code

Лабораторное задание № 2

"Проектирование программного обеспечения"

C#. Приложение WPF. Элементы управления. ListBox, ComboBox, ListView.
TabControl. в Visual studio code

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 Visual studio code.

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 программных средств

Разработка структуры программы и модульное программирование

Разработка программного модуля. Структурное программирование

Лабораторное задание № 2

Тема 2.1 "Разработка программных продуктов"

C#. Приложение WPF. Ресурсы. Стили. Анимация.

Лабораторное задание № 3

"Отладка, тестирование и сопровождение программных продуктов"

C#. Приложение WPF. Привязка. Шаблоны. Visual studio code

Лабораторное задание № 4

Диаграммы прецедентов. Диаграммы деятельности.

UML. Диаграммы классов.. С использованием Rational Architect.

Лабораторное задание № 5

UML. Диаграммы последовательности действий. Диаграммы компонентов. С использованием Rational Architect.

Лабораторное задание № 6

Отладка, тестирование и сопровождение программных продуктов Ручной контроль Классификация ошибок с использованием Visual studio code.

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.

4 семестр

Лабораторное задание № 1

"Общие принципы разработки программного обеспечения"

Си ++. Компоновка. DockPanel, WrapPanel, Canvas в Visual studio code

Лабораторное задание № 2

"Проектирование программного обеспечения"

Си++. Элементы управления. Меню. ToolBar. TreeView. DataGrid. Calendar.. Image. InkCanvas в Visual studio code

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 Visual studio code

Лабораторное задание № 5

"Проектирование программного обеспечения"

Си++. Элементы управления. ListBox, ComboBox, ListView. TabControl. Меню. в Visual studio code

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.

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 (два теоретических вопроса и одно практико-ориентированное задание). Проверка ответов и объявление результатов производится в день зачета.

Экзамен проводится по расписанию промежуточной аттестации в письменном виде. Количество вопросов в экзаменационном билете – 3 (два теоретических вопроса и одно практико-ориентированное задание). 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 опроса и посредством выполнении лабораторных заданий. В ходе самостоятельной работы каждый студент обязан прочитать основную и по возможности дополнительную литературу по изучаемой теме, дополнить конспекты лекций недостающим материалом, выписками из рекомендованных первоисточников. Выделить непонятные термины, найти их значение в энциклопедических словарях.

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

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