

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность: Ректор

Дата подписания: 13.11.2024 11:38:11

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 38.03.05 Бизнес-информатика
Направленность 38.03.05.01 "Информационно-аналитические системы"

Для набора 2021 года

Квалификация
Бакалавр

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

| Семестр (<Курс>.<Семестр на курсе>) | 4 (2.2) | | Итого | |
|---|---------|-----|-------|-----|
| | 16 | | | |
| Неделя | 16 | | | |
| Вид занятий | УП | РП | УП | РП |
| Лекции | 10 | 10 | 10 | 10 |
| Лабораторные | 12 | 12 | 12 | 12 |
| Итого ауд. | 22 | 22 | 22 | 22 |
| Контактная работа | 22 | 22 | 22 | 22 |
| Сам. работа | 437 | 437 | 437 | 437 |
| Часы на контроль | 9 | 9 | 9 | 9 |
| Итого | 468 | 468 | 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-3: Способен управлять контентом Интернет-ресурсов предприятия, процессами создания информационных сервисов |
| ПК-5: Способен использовать основные инструментальные методы в профессиональной деятельности для решения проблемной ситуации заинтересованных лиц |

В результате освоения дисциплины обучающийся должен:

| |
|---|
| Знать: |
| методы программирования и методы разработки эффективных алгоритмов решения профессиональных задач (соотнесено с индикатором ПК-3.1). общие принципы разработки программного обеспечения (соотнесено с индикатором ПК-5.1). |
| Уметь: |
| применять программное обеспечение для решения профессиональных задач (соотнесено с индикатором ПК-3.2) разрабатывать структуры программного обеспечения (соотнесено с индикатором ПК-5.2). |
| Владеть: |
| основами применения инструментария программирования для решения профессиональных задач (соотнесено с индикатором ПК-3.3). разработки структурных и функциональных схем (соотнесено с индикатором ПК-5.3). |

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 / | 4 | 2 | ПК-3, ПК-5 | Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3 |
| 1.2 | "Общие принципы разработки программного обеспечения" С#. Приложение WPF. Компонновка. Grid, GridSplitter, StackPanel, DockPanel, WrapPanel, Canvas. / Лаб / | 4 | 2 | ПК-3, ПК-5 | 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 / | 4 | 2 | ПК-3, ПК-5 | 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 / Ср / | 4 | 12 | ПК-3, ПК-5 | Л1.1, Л1.2, Л2.1, Л2.2, Л2.3 |
| 1.5 | "Практическое программирование" Знакомство с периферийными устройствами. Порты ввода-вывода. Аналогово-цифровой преобразователь. Последовательные порты UART. Интерфейс SPI. Интерфейс TWI (I2C). / Лаб / | 4 | 2 | ПК-3, ПК-5 | Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3 |
| 1.6 | 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 Аттестация программного средства / Ср / | 4 | 60 | ПК-3, ПК-5 | Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3 |

| | | | | | |
|------|--|---|----|------------|------------------------------------|
| 1.7 | "Практическое программирование" Общие принципы устройства И функционирования ATMELE AVR. Общее устройство, организация памяти, тактирование, сброс. / Лек / | 4 | 2 | ПК-3, ПК-5 | Л1.1, Л1.2, Л1.3, Л2.2, Л2.3 |
| 1.8 | "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 / Ср / | 4 | 12 | ПК-3, ПК-5 | Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3 |
| 1.9 | "Практическое программирование" Знакомство с периферийными устройствами Порты ввода-вывода Аналогово-цифровой преобразователь. Последовательные порты UART Интерфейс SPI. Интерфейс TWI (I2C) / Лаб / | 4 | 2 | ПК-3, ПК-5 | Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3 |
| 1.10 | Практическое программирование. Знакомство с периферийными устройствами Порты ввода-вывода Аналогово-цифровой преобразователь. Последовательные порты UART Интерфейс SPI. Интерфейс TWI (I2C) / Ср / | 4 | 46 | ПК-3, ПК-5 | Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3 |
| 1.11 | «Функции на языке С++» Разработка проектов по решению классических задач на нахождение факториалов и сумм числовых последовательностей. Алгоритмическая реализация рекурсии. / Ср / | 4 | 24 | ПК-3, ПК-5 | Л1.1, Л1.2, Л1.3, Л2.1 |
| 1.12 | Разработка программных продуктов Требования к современным технологиям Общие принципы разработки программных средств Разработка структуры программы и модульное программирование Разработка программного модуля. Структурное программирование / Ср / | 4 | 50 | ПК-3, ПК-5 | 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 | "Разработка программных продуктов" Языки программирования. Стили программирования. Case-средства разработки ПО. Эффективность. Оценка качества ПО / Лек / | 4 | 2 | ПК-3, ПК-5 | 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 / | 4 | 2 | ПК-3, ПК-5 | 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. / Ср / | 4 | 6 | ПК-3, ПК-5 | 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 перестановки элементов массива, задачи сортировок элементов массива. / Лаб / | 4 | 2 | ПК-3, ПК-5 | 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. "Разработка программных продуктов" С#. Приложение WPF. Ресурсы. Стили. Анимация. / Ср / | 4 | 30 | ПК-3, ПК-5 | Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3 |
| 2.6 | Методы объектно-ориентированного проектирования ПО. UML. | 4 | 26 | ПК-3, ПК-5 | Л1.1, Л1.2, Л1.3, |

| | | | | | |
|------|--|---|----|------------|------------------------------------|
| | <p>Диаграммы прецедентов. Диаграммы деятельности. UML. Диаграммы классов. Диаграммы последовательности действий. Диаграммы компонентов. Коллективная разработка ПО. Организация работ.</p> <p>/ Ср /</p> | | | | Л2.1, Л2.2, Л2.3 |
| 2.7 | <p>Отладка, тестирование и сопровождение программных продуктов</p> <p>Ручной контроль Классификация ошибок / Ср /</p> | 4 | 16 | ПК-3, ПК-5 | Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3 |
| 2.8 | <p>«Символьные данные и строки»</p> <p>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 /</p> | 4 | 2 | ПК-3, ПК-5 | Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3 |
| 2.9 | <p>"Символьные данные и строки"</p> <p>Подсчет количества слов в предложении. Предварительно привести строку к нормированному виду, удалить все лишние пробелы. Нахождение короткого слово в строке / Лаб /</p> | 4 | 2 | ПК-3, ПК-5 | Л1.1, Л1.2, Л1.3, Л2.1 |
| 2.10 | <p>«Структуры и объединения»</p> <p>Битовые поля. Определения, способы объявления, инициализация структур, методы доступа к данным структуры, размещение структур и определение их размера в памяти, массивы структур. Структуры и указатели. Объединения. / Ср /</p> | 4 | 46 | ПК-3, ПК-5 | Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3 |
| 2.11 | <p>«Алгоритмы сортировки массивов»</p> <p>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 /</p> | 4 | 46 | ПК-3, ПК-5 | Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3 |
| 2.12 | <p>Файлы. Функции по работе с файлами»</p> <p>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. / Ср /</p> | 4 | 28 | ПК-3, ПК-5 | Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3 |
| 2.13 | <p>Курсовой проект. Перечень тем представлен в приложении 1 к рабочей программе дисциплины. / Ср /</p> | 4 | 35 | ПК-3, ПК-5 | Л1.1, Л1.2, Л1.3, Л2.1, Л2.2, Л2.3 |
| 2.14 | <p>/ Экзамен /</p> | 4 | 9 | ПК-3, ПК-5 | 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 |
|------|----------------------------------|---|--|---|
| 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 |

| | Авторы, | Заглавие | Издательство, год | Колич-во |
|------|---------------|--|---|---|
| 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 | Лебедевко, 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

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

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-3: Способен управлять контентом Интернет-ресурсов предприятия, процессами создания информационных сервисов | | | |
| 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 экзамене | Э (1-38) О (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 | ПОЗЭ (1-5) ЛЗ (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 | ПОЗЭ (1-5) ЛЗ (1-6). КП (1-20) |
| ПК-5: Способен использовать основные инструментальные методы в профессиональной деятельности для решения проблемной ситуации заинтересованных лиц | | | |
| З общие принципы разработки программного обеспечения | изучение основной и дополнительной литературы, лекционного материала, использование профессиональных баз, данных для выбора алгоритма решения задачи программирования | полнота и содержательность ответа, обоснованность выбора базовых алгоритмов программирования в ответах на вопросы опроса, теоретических вопросов на экзамене | Э (1-38) О (1-16) |
| У разрабатывать структуры программного обеспечения | решение лабораторных, практико-ориентированных заданий, выбор метода программирования | правильность выполнения заданий, сложность алгоритмических решений, обоснованность применения выбранного метода программирования в курсовом проекте | ПОЗЭ (1-5) ЛЗ (1-6). КП (1-20) |
| В разработки структурных и функциональных схем | авторское программное решение лабораторных, практико-ориентированных заданий, на основе базовых принципов разработки программного обеспечения | объем выполнения заданий, полнота анализа программных решений в курсовом проекте | ПОЗЭ (1-5) ЛЗ (1-6). КП (1-20) |

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

1.2 Шкалы оценивания:

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

84-100 баллов (оценка «отлично»)

67-83 баллов (оценка «хорошо»)

50-66 баллов (оценка «удовлетворительно»)

0-49 баллов (оценка «неудовлетворительно»)

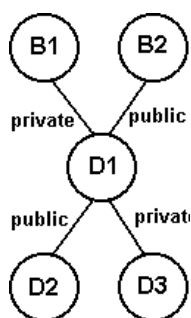
2 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Вопросы к экзамену

1. 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
16. История ЯП. Классификация и область применения ЯП.
17. Парадигмы программирования. Процедурные языки. Аппликативные языки.
18. Языки логического программирования. Объектно-ориентированные языки.
19. Стандартизация языков программирования.
20. Программирование разветвляющихся алгоритмов. Условные операторы `if`. Оператор выбора `switch`.
21. Функции библиотеки `math.h`. Линейный алгоритм.
22. Программирование циклических алгоритмов.
23. Программирование с использованием одномерных массивов.
24. Программирование с использованием двумерных массивов.
25. Объектно-ориентированное программирование. Объект. Класс. Экземпляр.
26. Инкапсуляция.
27. Полиморфизм.
28. Конструктор. Инициализация переменных в конструкторе. Конструктор без параметров. Копирующий конструктор. Содержательный конструктор.
29. Константные ссылки и указатели. Константные методы классов. Перегрузка.
30. Понятие наследования. Виды наследования.
31. Множественное наследование.
32. Дружественные функции.
33. Дружественные члены класса (методы). Дружественные классы.
34. Методы тестирования ПО.
35. Сопровождение ПО.
36. Защита программного кода ПО.
37. CASE-технологии.
38. Коллективная разработка ПО. Организация работ.

Практико-ориентированные задания к экзамену

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.

Вопросов для опроса

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: 16 баллов.

Лабораторные задания

Тематика лабораторных заданий по разделам

Раздел 1. Развитие технологий и методов программирования

Лабораторное задание № 1 "Общие принципы разработки программного обеспечения"

С#. Приложение WPF. Компонировка. Grid, GridSplitter, StackPanel, DockPanel, WrapPanel, Canvas.

Лабораторное задание № 2 "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 № 3 "Практическое программирование"

Знакомство с периферийными устройствами. Порты ввода-вывода. Аналогово-цифровой преобразователь. Последовательные порты UART. Интерфейс SPI. Интерфейс TWI (I2C).

Раздел 2. Практическое программирование

Лабораторное задание №.4 "Общие принципы разработки программных средств"

Разработка структуры программы и модульное программирование. Разработка программного модуля. Структурное программирование.

Лабораторное задание № 5 "Массивы: одномерные и двумерные"

Массивы: одномерные массивы, задачи поиска, замены и перестановок элементов массива, задачи сортировок элементов массива.

Лабораторное задание №.6 "Символьные данные и строки"

Подсчет количества слов в предложении. Предварительно привести строку к нормированному виду, удалить все лишние пробелы. Нахождение короткого слово в строке

Критерии оценивания:

Каждое лабораторное задание максимально оценивается в 14 баллов:

- 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, sms-систем, stn-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 экзамена, защиты курсового проекта.

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. 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 работает».

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.