**МИНИСТЕРСТВО ОБЩЕГО И ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ   
РОСТОВСКОЙ ОБЛАСТИ**

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ РОСТОВСКОЙ ОБЛАСТИ**

**РОСТОВСКИЙ-НА-ДОНУ КОЛЛЕДЖ СВЯЗИ И ИНФОРМАТИКИ**

**Фонд оценочных средств**

**по учебной практике**

**ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем»**

программы подготовки специалистов среднего звена

для специальности

**09.02.07 «Информационные системы и программирование»**

(базовой подготовки)

г. Ростов-на-Дону

2024г.

|  |  |
| --- | --- |
| **ОДОБРЕНО**  На заседании цикловой комиссии программирования  Протокол №10от «26» июня 2024 г.  Председатель ЦК \_\_\_\_\_\_\_\_\_\_ / А.С. Сулавко / | **УТВЕРЖДАЮ**  Заместитель директора по НМР  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ И.В. Подцатова /  «30» августа 2024 г. |

Фонд оценочных средств разработан на основе рабочей программы учебной практики, разработанной в 2024г. по специальности 09.02.07 «Информационные системы и программирование»

Разработчик: Швецов В.Р. – преподаватель государственного бюджетного образовательного учреждения среднего профессионального образования Ростовской области «Ростовский-на-Дону колледж связи и информатики»

Рецензент:

**СОДЕРЖАНИЕ**

1 Паспорт фонда оценочных средств 4

2 Результаты освоения практики, подлежащие проверке 6

3 Оценка освоения учебной практики 8

4 Контрольно-оценочные материалы для промежуточной аттестации по учебной дисциплине 64

1. **Паспорт фонда оценочных средств**

В результате освоения учебной практики по модулю ПМ.01 обучающийся должен обладать предусмотренными ФГОС по специальности 09.02.07 «Информационные системы и программирование» со следующими умениями, знаниями:

ОК 1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

ОК 2. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности

ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях

ОК 4 Эффективно взаимодействовать и работать в коллективе и команде

ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК 9 Пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

ПК1.6. Разрабатывать модули программного обеспечения для мобильных платформ.

ПК 3.1 Осуществлять ревьюирование программного кода в соответствии с технической документацией.

ПК 3.2 Выполнять измерение характеристик компонент программного продукта для определения соответствия заданным критериям.

ПК 5.1. Собирать исходные данные для разработки проектной документации на информационную систему

ПК 5.2. Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика.

ПК 6.1 Разрабатывать техническое задание на сопровождение информационной системы.

ПК 8.1. Разрабатывать дизайн-концепции веб-приложений в соответствии с корпоративным стилем заказчика

ПК 8.2. Формировать требования к дизайну веб-приложений на основе анализа предметной области и целевой аудитории.

У1. Осуществлять разработку кода программного модуля на языках низкого и высокого уровней;

У2. Создавать программу по разработанному алгоритму как отдельный модуль;

У3. Выполнять отладку и тестирование программы на уровне модуля;

У4. Осуществлять разработку кода программного модуля на современных языках программирования;

У5. Уметь выполнять оптимизацию и рефакторинг программного кода;

У6. Оформлять документацию на программные средства,

З1. Основные этапы разработки программного обеспечения;

З2. Основные принципы технологии структурного и объектно-ориентированного программирования;

З3. Способы оптимизации и приемы рефакторинга;

З4. Основные принципы отладки и тестирования программных продуктов.

Формой аттестации в соответствии с рабочей программой по учебной дисциплине является зачет.

**2. Результаты освоения учебной практики, подлежащие проверке**

2.1. В результате аттестации по учебной УП.01.01 Учебная практика осуществляется комплексная проверка следующих умений и знаний:

Таблица 1.1

|  |  |  |
| --- | --- | --- |
| **Результаты обучения: умения, знания и общие и профессиональные компетенции** | **Показатели оценки результата** | **Форма контроля и оценивания** |
| **Общие компетенции:**  ОК 1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам  ОК 2. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности  ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях  ОК4. Эффективно взаимодействовать и работать в коллективе и команде  ОК5. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста  ОК 9. Пользоваться профессиональной документацией на государственном и иностранном языках. | * обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач; * адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач; * использование различных источников, включая электронные ресурсы, медиаресурсы, Интернет-ресурсы, периодические издания по специальности для решения профессиональных задач;   эффективность использования информационно-коммуникационных технологий в профессиональной деятельности согласно формируемым умениям и получаемому практическому опыту. | * практическая проверка (защита практических работ).   (6-й семестр). |
| **Профессиональные компетенции:**  ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.  ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.  ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.  ПК 1.4. Выполнять тестирование программных модулей.  ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.  ПК1.6. Разрабатывать модули программного обеспечения для мобильных платформ.  ПК 3.1 Осуществлять ревьюирование программного кода в соответствии с технической документацией.  ПК 3.2 Выполнять измерение характеристик компонент программного продукта для определения соответствия заданным критериям.  ПК 5.1. Собирать исходные данные для разработки проектной документации на информационную систему  ПК 5.2. Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика.  ПК 6.1 Разрабатывать техническое задание на сопровождение информационной системы.  ПК 8.1. Разрабатывать дизайн-концепции веб-приложений в соответствии с корпоративным стилем заказчика  ПК 8.2. Формировать требования к дизайну веб-приложений на основе анализа предметной области и целевой аудитории. |
| **Уметь:** |
| У1. Осуществлять разработку кода программного модуля на языках низкого и высокого уровней;  У2. Создавать программу по разработанному алгоритму как отдельный модуль;  У3. Выполнять отладку и тестирование программы на уровне модуля;  У4. Осуществлять разработку кода программного модуля на современных языках программирования;  У5. Уметь выполнять оптимизацию и рефакторинг программного кода;  У6. Оформлять документацию на программные средства |
| **Знать:** |
| З1. Основные этапы разработки программного обеспечения;  З2. Основные принципы технологии структурного и объектно-ориентированного программирования;  З3. Способы оптимизации и приемы рефакторинга;  З4. Основные принципы отладки и тестирования программных продуктов. |

**3. Оценка освоения учебной практики:**

**3.1. Формы и методы оценивания**

Предметом оценки служат умения и знания, предусмотренные **рабочей программой** по дисциплине ОПЦ.08 «Основы проектирования баз данных».

Технология оценки знаний и умений по дисциплине соответствует действующему Положению о текущем контроле и промежуточной аттестации студентов ГБПОУ РО «РКСИ».

Контроль и оценка освоения учебной практики по темам (разделам)

Таблица 2.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Элемент учебной практики** | **Формы и методы контроля** | | | | |
| **Текущий контроль** | | **Промежуточная аттестация** | |
| **Форма контроля** | **Проверяемые ОК, ПК. У, З,** | **Форма контроля** | **Проверяемые ОК, ПК, У, З,** |
| Тема 1. Структурное программирование | 1)Устный опрос  2)**Практическое занятие №1.**Структурное программирование | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |  | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |
| Тема 2. Объектно-ориентированное программирование | 1)Устный опрос  2)**Практическое занятие №2.**Объектно-ориентированное программирование | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |  | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |
| Тема 3. Паттерны проектирования | 1)Устный опрос  2)**Практическое занятие №3.**Паттерны проектирования | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |  | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |
| Тема 4. Классы. | 1)Устный опрос  2)**Практическое занятие №4.** Классы. | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |  | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |
| Тема 5. Событийно-управляемое программирование | 1)Устный опрос  2)**Практическое занятие №5.** Событийно-управляемое программирование | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |  | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |
| Тема 6. Оптимизация и рефакторинг кода | 1)Устный опрос  2)**Практическое занятие №6.** Оптимизация и рефакторинг кода | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |  | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |
| Тема 7. Проектирования пользовательских интерфейсов | 1)Устный опрос  2)**Практическое занятие №7.** Проектирования пользовательских интерфейсов | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |  | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |
| Тема 8. Разработка пользовательского интерфейса. | 1)Устный опрос  2)**Практическое занятие №8.** Разработка пользовательского интерфейса. | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |  | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |
| Тема 9. Разработка приложения. Работа с БД. Доступ к данным. | 1)Устный опрос  2)**Практическое занятие №9.** Разработка приложения.  3) **Практическое занятие №10.** Работа с БД.  4) **Практическое занятие №11.** Доступ к данным. | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |  | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |
| Тема 10. Отладка и тестирование программного обеспечения | 1)Устный опрос  2)**Практическое занятие №12** Отладка и тестирование программного обеспечения | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |  |  |
|  |  |  | Зачет (отчет по учебной практике) | ОК 01, ОК 02, ОК 5, ОК 9  У1, У2, У3, У4, У5, У6 З1, З2, З3, З4 |

**3.2 Задания для оценки освоения учебной практики**

**3.2.1 Задания для оценки знаний и умений в ходе проведения текущего контроля**

Перечень вопросов по темам учебной практики УП 01.01. Учебная практика

**Тема 1. Структурное программирование**

1. Что такое структурное программирование и какие принципы лежат в его основе?
2. В чем заключаются основные различия между структурным программированием и объектно-ориентированным программированием?
3. Какие основные элементы используются в структурном программировании для создания программ? Опишите их.
4. Что такое блок-схема и как она используется в структурном программировании?
5. Объясните понятие "последовательность", "ветвление" и "цикл" в контексте структурного программирования.
6. Почему считается, что структурное программирование способствует улучшению читаемости и поддержки кода?
7. Как структурное программирование помогает в управлении сложностью программного кода?
8. В чем заключаются основные недостатки структурного программирования по сравнению с другими парадигмами программирования?
9. Объясните принцип единственного входа и выхода (single entry, single exit) и его важность в структурном программировании.
10. Как используются функции и процедуры в структурном программировании для улучшения модульности и повторного использования кода?

**Критерии оценки устного опроса**

* оценка **«отлично»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
* оценка **«хорошо»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
* оценка **«удовлетворительно»** выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;
* оценка **«неудовлетворительно»** выставляется студенту, если он не смог изложить суть основного вопроса

**Практическое занятие №1: Структурное программирование**

ОК1 выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

ОК2 использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности

ОК5 осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК9 пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

**Задание 1**: Основы структурного программирования

Напишите программу на языке программирования C (или другом поддерживающем структурное программирование), которая решает следующую задачу:

Вариант 1:

Запросите у пользователя два числа.

Выведите сумму их квадратов, используя последовательность операторов.

Вариант 2:

Запрашивает у пользователя три числа.

Выводит на экран результат деления первого числа на второе, а затем выводит остаток от деления этого результата на третье число.

Вариант 3:

Запрашивает у пользователя два числа и знак операции (‘+’, ‘-‘, ‘\*’ или ‘/’).

Выводит результат выбранной операции с использованием соответствующих операторов (if-else).

Вариант 4:

Запрашивает у пользователя два целых числа.

Выводит наименьшее общее кратное (НОК) и наибольший общий делитель (НОД) этих чисел, используя алгоритм Евклида.

**Задание 2**: Подпрограммы и функции

Создайте программу, включающую минимум три функции:

Вариант 1:

Функция для вычисления факториала числа.

Функция для определения, является ли число простым.

Функция для вывода Квадратного корня числа.

В основной программе вызовите каждую из этих функций с различными аргументами.

Вариант 2

Разработайте программу, которая:

Запрашивает у пользователя радиус круга.

Использует функцию для вычисления площади круга (S = π \* r^2) и функцию для вычисления длины окружности (C = 2 \* π \* r).

Выводит результаты расчетов.

Вариант 3

Напишите программу, в которой:

Пользователь вводит два числа.

Функция определяет, является ли первое число степенью второго числа.

Выводит результат проверки.

Вариант 4

Реализуйте программу, включающую:

Функцию для нахождения наименьшего общего кратного (НОК) двух чисел.

Функцию для нахождения наибольшего общего делителя (НОД) двух чисел с использованием алгоритма Евклида.

В основной программе запросите у пользователя два числа и выведите их НОК и НОД.

**Задание 3**: Работа с массивами

Напишите программу, использующую массивы для решения следующей задачи:

Вариант 1:

Заполните массив случайными числами.

Найдите минимальное и максимальное значения в массиве.

Отсортируйте массив в порядке убывания.

Вариант 2:

Создайте программу, в которой:

Заполните массив случайными целыми числами.

Найдите и выведите сумму всех элементов массива, стоящих на четных позициях.

Вариант 3:

Напишите программу, включающую следующие шаги:

Инициализируйте массив числами от 1 до 10.

Разверните порядок элементов массива (первый элемент становится последним, второй – предпоследним и так далее).

Выведите полученный массив на экран.

Вариант 4:

Реализуйте программу, в которой:

Создайте два массива одинаковой длины.

Заполните их случайными числами.

Посложите элементы массивов по соответствующим индексам и сохраните результат в третий массив.

Выведите исходные массивы и результат сложения на экран.

**Задание 4**: Структуры данных

Вариант 1:

Определите структуру данных "Студент", содержащую поля: имя, возраст и средний балл. Напишите программу, включающую следующие шаги:

Создайте массив студентов.

Заполните его данными о студентах.

Выведите информацию о студентах, чей средний балл выше 4.5.

Вариант 2:

Определите структуру "Книга" с полями: название, автор и год издания. Создайте массив из 5 экземпляров структуры "Книга". Выведите на экран информацию о книгах, изданных после 2000 года.

Вариант 3

Разработайте программу, в которой:

Определите структуру "Фильм" с полями: название, режиссер и год выпуска.

Создайте массив из 3 структур "Фильм".

Заполните массив данными о любых фильмах.

Выведите информацию о фильмах, выпущенных ранее 2010 года.

Вариант 4

Напишите программу, включающую следующие шаги:

Определите структуру "Сотрудник" с полями: имя, должность и зарплата.

Заполните массив данными о сотрудниках.

Выведите информацию о сотрудниках, чья зарплата превышает среднюю зарплату по всем сотрудникам.

**Задание 5**: Рекурсия

Вариант 1:

Напишите рекурсивную функцию для вычисления n-го числа Фибоначчи. В основной программе запросите у пользователя число n и выведите соответствующее число Фибоначчи.

Вариант 2

Напишите программу, используя рекурсию, которая:

Запрашивает у пользователя натуральное число.

Рекурсивно вычисляет сумму цифр этого числа.

Выводит полученную сумму.

Вариант 3

Создайте программу, включающую следующие шаги:

Реализуйте рекурсивную функцию для вычисления n-го члена последовательности Фибоначчи.

Пользователь вводит число n.

Выведите на экран n-й член последовательности Фибоначчи.

Вариант 4

Напишите программу, которая:

Реализует рекурсивную функцию для вычисления факториала числа.

Запрашивает у пользователя число.

Выводит на экран факториал введенного числа, используя рекурсивную функцию.

Пример рекурсивной функции на языке программирования Python для вычисления факториала числа может выглядеть следующим образом:

def factorial(n):

# Базовыйслучай

if n == 0 or n == 1:

return 1

# Случайрекурсии

else:

return n \* factorial(n-1)

В этом примере, если n равно 0 или 1, функция возвращает 1 (базовый случай). В противном случае, она вызывает саму себя для вычисления факториала числа n-1 (случай рекурсии), и так продолжается, пока не достигается базовый случай.

**Тема 2. Объектно-ориентированное программирование**

1. Что такое объект в контексте объектно-ориентированного программирования?
2. Чем отличается класс от объекта?
3. Объясните концепцию наследования и приведите пример его использования.
4. Что такое полиморфизм и как он реализуется в различных языках программирования?
5. Что такое инкапсуляция и почему она важна в ООП?
6. Опишите разницу между абстрактным классом и интерфейсом.
7. Каковы основные принципы SOLID в ООП и как они помогают в разработке программного обеспечения?
8. Что такое метод перегрузки (overloading) и метод переопределения (overriding)?
9. Объясните, что такое композиция и агрегация, и в чем их отличие.
10. Как работают геттеры и сеттеры и почему их использование считается хорошей практикой?

**Критерии оценки устного опроса**

* оценка **«отлично»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
* оценка **«хорошо»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
* оценка **«удовлетворительно»** выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;
* оценка **«неудовлетворительно»** выставляется студенту, если он не смог изложить суть основного вопроса

**Практическое занятие №2. Объектно-ориентированное программирование**

ОК1 выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

ОК2 использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности

ОК5 осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК9 пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

Задание

Решите задания согласно полученному варианту:

Разработка класса для геометрических фигур

Создайте базовый класс "ГеометрическаяФигура" с методами для вычисления площади и периметра.

Реализуйте подклассы, представляющие различные фигуры, такие как круг, треугольник, прямоугольник.

Поддержите полиморфизм для расчета площади и периметра для каждой конкретной фигуры.

2) Система управления библиотекой

Создайте классы "Книга" и "Библиотека".

Реализуйте взаимодействие между ними с использованием концепций инкапсуляции.

Добавьте методы для добавления новой книги, выдачи книги читателю, возврата книги.

3) Создание простого калькулятора

Создайте класс "Калькулятор" с методами для выполнения основных арифметических операций (сложение, вычитание, умножение, деление).

Используйте концепции инкапсуляции для безопасного доступа к операндам и результату.

4) Моделирование банковской системы

Создайте классы "БанковскийСчет" и "Клиент".

Реализуйте связь между клиентами и их банковскими счетами, используя агрегацию.

Добавьте методы для проведения операций по счету, таких как пополнение и снятие средств.

5) Симуляция виртуального зоопарка

Создайте базовый класс "Животное" с атрибутами: Имя, Вид, Вес.

Реализуйте подклассы, представляющие различные виды животных.

Используйте наследование и полиморфизм для добавления уникального поведения для каждого вида.

6) Управление задачами в проекте

Создайте классы "Проект" и "Задача".

Реализуйте связь между задачами и проектом, используя агрегацию.

Добавьте методы для назначения задачи исполнителю, изменения статуса задачи.

7) Работа с базой данных объектов

Создайте класс "БазаДанных", который может хранить объекты разных классов.

Реализуйте методы для добавления, удаления и поиска объектов в базе данных.

Используйте принцип полиморфизма для работы с различными типами объектов.

**Критерии оценки выполнения практических заданий.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе практики, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет по итогам окончания практики студентом не подготовлен.

**Тема 3 Паттерны проектирования**

1. Что такое паттерны проектирования и почему они важны в разработке программного обеспечения?
2. Объясните разницу между порождающими, структурными и поведенческими паттернами проектирования.
3. Что такое Singleton паттерн и когда его следует использовать?
4. Опишите паттерн Factory Method и приведите пример его применения.
5. В чем суть паттерна Observer и как он реализуется на практике?
6. Как работает паттерн Decorator и для каких задач он подходит лучше всего?
7. Что такое паттерн Strategy и как он помогает в решении проблем проектирования?
8. Объясните, как паттерн Adapter может быть использован для интеграции несовместимых интерфейсов.
9. Что представляет собой паттерн Command и в каких случаях его использование наиболее оправдано?

**Критерии оценки устного опроса**

* оценка **«отлично»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
* оценка **«хорошо»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
* оценка **«удовлетворительно»** выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;
* оценка **«неудовлетворительно»** выставляется студенту, если он не смог изложить суть основного вопроса

**Практическое занятие №3. Паттерны проектирования**

ОК1 выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

ОК2 использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности

ОК5 осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК9 пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

Цель работы: закрепить знания по паттернам в программировании

Задание: выполнить задания по каждому из типов паттернов.

Ход работы:

1. Задание по паттерну "Одиночка" (Singleton):

Реализуйте класс с использованием паттерна "Одиночка".

Создайте несколько объектов и убедитесь, что они являются одним экземпляром.

2. Задание по порождающим паттернам:

Напишите фабричный метод для создания различных типов продуктов.

Разработайте абстрактную фабрику, создающую связанные объекты.

Примените паттерны в простом проекте, используя порождающие паттерны.

3. Задание по структурным паттернам:

Реализуйте декоратор для расширения функциональности базового класса.

Используйте адаптер для взаимодействия между двумя несовместимыми интерфейсами.

Создайте структуру с использованием компоновщика для управления древовидными объектами.

4. Задание по поведенческим паттернам:

Разработайте стратегию для реализации различных алгоритмов.

Используйте наблюдатель для реализации механизма подписки и оповещения.

**Критерии оценки выполнения практических заданий.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе практики, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет по итогам окончания практики студентом не подготовлен.

**Тема 4. Классы**

1. Что такое класс в объектно-ориентированном программировании и как он отличается от объекта?
2. Как объявить и определить класс на примере выбранного языка программирования (например, Python, Java, C++)?
3. Какие основные компоненты могут содержаться в классе? Объясните их назначение.
4. Что такое конструктор в классе и для чего он используется?
5. Как работает наследование классов и как оно реализуется в различных языках программирования?
6. Что такое инкапсуляция и как она достигается с помощью классов?
7. Объясните концепцию полиморфизма в контексте классов и объектов.
8. Как реализовать методы доступа (геттеры и сеттеры) в классе и для чего они нужны?
9. Что такое абстрактный класс и интерфейс? В чем их основные различия и как они применяются?
10. Как работают статические методы и свойства в классе, и чем они отличаются от экземплярных методов и свойств?

**Критерии оценки устного опроса**

* оценка **«отлично»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
* оценка **«хорошо»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
* оценка **«удовлетворительно»** выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;
* оценка **«неудовлетворительно»** выставляется студенту, если он не смог изложить суть основного вопроса

**Практическое занятие №4. Классы**

ОК1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

ОК2 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

ОК5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК9 Пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

Цель работы: закрепить практически знания о классах в программировании

Задание:

Вариант 1

Задание №1: Основы классов

Создайте класс "Студент" с атрибутами: имя, возраст, номер студенческого билета. Напишите метод для вывода информации о студенте.

Создайте несколько экземпляров класса "Студент" и вызовите метод для вывода информации о каждом из них.

Задание 2: Методы и свойства классов

Добавьте в класс "Студент" метод для изменения возраста.

Создайте свойство "средний балл" для класса "Студент" и реализуйте геттер и сеттер для этого свойства.

Протестируйте новые методы и свойства на созданных ранее экземплярах класса "Студент".

Задание 3: Наследование

Создайте базовый класс "Человек" с атрибутами: имя и возраст.

Унаследуйте класс "Студент" от класса "Человек". Переопределите метод вывода информации так, чтобы он включал информацию о студенческом билете.

Создайте экземпляр класса "Студент" и вызовите метод вывода информации.

Задание 4: Полиморфизм

Создайте еще один класс, например, "Преподаватель", унаследованный от "Человека", и добавьте атрибут "предмет".

Реализуйте полиморфный метод "информация" в базовом классе "Человек", который будет автоматически вызывать метод вывода информации для каждого класса (студент, преподаватель).

Создайте экземпляры классов "Студент" и "Преподаватель" и вызовите полиморфный метод "информация".

Задание 5: Композиция

Создайте класс "Группа", который будет содержать список студентов.

Реализуйте метод добавления студента в группу и метод вывода информации о всех студентах в группе.

Создайте экземпляр класса "Группа" и добавьте в нее созданных ранее студентов. Выведите информацию о студентах в группе.

Вариант 2

Задание 1: Основы классов и объектов

Создайте класс "Автомобиль" с атрибутами: марка, модель, год выпуска. Напишите метод для вывода информации об автомобиле.

Создайте несколько экземпляров класса "Автомобиль" и вызовите метод для вывода информации о каждом из них.

Задание 2: Методы и свойства классов

Добавьте в класс "Автомобиль" метод для изменения года выпуска.

Создайте свойство "пробег" для класса "Автомобиль" и реализуйте геттер и сеттер для этого свойства.

Протестируйте новые методы и свойства на созданных ранее экземплярах класса "Автомобиль".

Задание 3: Наследование

Создайте базовый класс "Транспортное средство" с атрибутами: масса и скорость.

Унаследуйте класс "Автомобиль" от класса "Транспортное средство". Переопределите метод вывода информации так, чтобы он включал информацию о марке, модели, годе выпуска, массе и скорости.

Создайте экземпляр класса "Автомобиль" и вызовите метод вывода информации.

Задание 4: Полиморфизм

Создайте еще один класс, например, "Грузовик", унаследованный от "Транспортного средства", и добавьте атрибут "грузоподъемность".

Реализуйте полиморфный метод "информация" в базовом классе "Транспортное средство", который будет автоматически вызывать метод вывода информации для каждого класса (автомобиль, грузовик).

Создайте экземпляры классов "Автомобиль" и "Грузовик" и вызовите полиморфный метод "информация".

Задание 5: Композиция

Создайте класс "Автопарк", который будет содержать список автомобилей.

Реализуйте метод добавления автомобиля в автопарк и метод вывода информации обо всех автомобилях в автопарке.

Создайте экземпляр класса "Автопарк" и добавьте в него созданные ранее автомобили. Выведите информацию об автомобилях в автопарке.

Вариант 3

Задание 1: Основы классов

Создайте класс "БанковскийСчет" с атрибутами: номер счета, владелец счета, баланс. Напишите метод для вывода информации о банковском счете.

Создайте несколько экземпляров класса "БанковскийСчет" и вызовите метод для вывода информации о каждом из них.

Задание 2: Методы и свойства классов

Добавьте в класс "БанковскийСчет" метод для внесения средств на счет.

Создайте свойство "кредитный лимит" для класса "БанковскийСчет" и реализуйте геттер и сеттер для этого свойства.

Протестируйте новые методы и свойства на созданных ранее экземплярах класса "БанковскийСчет".

Задание 3: Наследование

Создайте базовый класс "ФинансовыйАккаунт" с атрибутами: владелец аккаунта, баланс.

Унаследуйте класс "БанковскийСчет" от класса "ФинансовыйАккаунт". Переопределите метод вывода информации так, чтобы он включал информацию о номере счета.

Создайте экземпляр класса "БанковскийСчет" и вызовите метод вывода информации.

Задание 4: Полиморфизм

Создайте еще один класс, например, "КредитнаяКарта", унаследованный от "ФинансовогоАккаунта", и добавьте атрибут "кредитный лимит".

Реализуйте полиморфный метод "информация" в базовом классе "ФинансовыйАккаунт", который будет автоматически вызывать метод вывода информации для каждого класса (банковский счет, кредитная карта).

Создайте экземпляры классов "БанковскийСчет" и "КредитнаяКарта" и вызовите полиморфный метод "информация".

Задание 5: Композиция

Создайте класс "Клиент", который будет содержать список финансовых аккаунтов клиента.

Реализуйте метод добавления финансового аккаунта в список клиента и метод вывода информации обо всех аккаунтах клиента.

Создайте экземпляр класса "Клиент" и добавьте в него созданные ранее банковский счет и кредитную карту. Выведите информацию об аккаунтах клиента.

**Критерии оценки выполнения практических заданий.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе практики, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет по итогам окончания практики студентом не подготовлен.

**Тема 5. Событийно-управляемое программирование**

1. Что такое событийно-управляемое программирование и чем оно отличается от других парадигм программирования?
2. Какое основное предназначение событий в контексте событийно-управляемого программирования?
3. Что такое обработчик событий и какова его роль в системе?
4. Объясните концепцию цикла обработки событий (event loop) и как он реализуется в различных языках программирования.
5. Какие преимущества и недостатки имеет событийно-управляемое программирование по сравнению с другими моделями?
6. Как реализуется модель "подписчик-издатель" (publish-subscribe) в событийно-управляемом программировании?
7. Какие существуют методы предотвращения блокировок и повышения производительности в событийно-управляемых системах?
8. Как управляются асинхронные события и каковы основные подходы к их обработке?
9. Приведите примеры использования событийно-управляемого программирования в современных веб-приложениях.
10. Как тестировать программное обеспечение, использующее событийно-управляемую архитектуру?

**Критерии оценки устного опроса**

* оценка **«отлично»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
* оценка **«хорошо»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
* оценка **«удовлетворительно»** выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;

оценка **«неудовлетворительно»** выставляется студенту, если он не смог изложить суть основного вопроса

**Практическое занятие №5. Событийно-управляемое программирование**

ОК1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

ОК2 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

ОК5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК9 Пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

Задание 1: Знакомство с событийно-управляемым программированием

Цель: ознакомиться с основными концепциями событийно-управляемого программирования и его применением.

Создайте простое приложение на языке программирования, который вы предпочитаете (например, Python, JavaScript, C#), используя концепцию событийно-управляемого программирования.

В приложении определите несколько событий и соответствующие обработчики этих событий.

Реализуйте возможность подписываться на события и отписываться от них.

Протестируйте приложение, гарантируя, что события вызываются и обрабатываются корректно.

Задание 2: Реализация интерфейса событийной обработки веб-приложения

Цель: освоить создание интерфейса для веб-приложений с использованием событийно-управляемого программирования.

Создайте простое веб-приложение с использованием HTML, CSS и JavaScript.

Определите различные события, которые могут происходить в вашем веб-приложении (например, клик на кнопке, наведение мыши на элемент, отправка формы и т. д.).

Напишите JavaScript-код, который будет обрабатывать эти события.

Используйте функции обратного вызова (callback) для реализации событийной обработки.

Протестируйте веб-приложение, убедившись, что события обрабатываются корректно.

Задание 3: Реализация событий в графическом интерфейсе пользователя

Цель: Практическое применение событийно-управляемого программирования в графическом интерфейсе пользователя (GUI).

Используя библиотеку для создания графического интерфейса (например, Tkinter для Python, JavaFX для Java), разработайте простое GUI-приложение.

Определите различные события, которые могут происходить в вашем GUI (например, нажатие кнопки, перемещение мыши, ввод текста и т. д.).

Реализуйте обработчики этих событий с использованием событийно-управляемого программирования.

Добавьте функциональность, которая позволит пользователю подписываться на события и отписываться от них.

Протестируйте GUI-приложение, убедившись, что события обрабатываются правильно и пользователь может взаимодействовать с интерфейсом.

Задание 4: Использование событийно-управляемого программирования в мобильном приложении

Цель: освоить создание мобильных приложений с использованием событийно-управляемого программирования.

Используя среду разработки для мобильных приложений (например, Android Studio для Android, Xcode для iOS), создайте простое мобильное приложение.

Определите различные события, которые могут происходить в вашем мобильном приложении (например, касание экрана, ввод текста, перемещение по экрану и т. д.).

Реализуйте обработчики этих событий с использованием событийно-управляемого программирования.

Обеспечьте возможность пользователю подписываться на события и отписываться от них.

Протестируйте мобильное приложение на эмуляторе или реальном устройстве, убедившись, что события обрабатываются корректно и пользователь может взаимодействовать с приложением.

**Критерии оценки выполнения практических заданий.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе практики, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет по итогам окончания практики студентом не подготовлен.

**Тема 6. Оптимизация и рефакторинг кода**

1. Что такое оптимизация кода и почему она важна в разработке программного обеспечения?
2. Какие основные типы оптимизации кода вы знаете, и когда их следует применять?
3. В чем разница между профилированием и оптимизацией кода?
4. Какие инструменты вы используете для анализа производительности кода?
5. Что такое рефакторинг кода и зачем его проводить?
6. Какие принципы руководят рефакторингом кода?
7. Какие типичные признаки указывают на необходимость рефакторинга кода?
8. Какие методы избегания дублирования кода вы используете при рефакторинге?
9. Какие могут быть негативные последствия при проведении оптимизации или рефакторинга кода?
10. Как измерить эффективность проведенного рефакторинга или оптимизации?

**Критерии оценки устного опроса**

* оценка **«отлично»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
* оценка **«хорошо»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
* оценка **«удовлетворительно»** выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;

оценка **«неудовлетворительно»** выставляется студенту, если он не смог изложить суть основного вопроса

**Практическое занятие №6. Оптимизация и рефакторинг кода**

ОК1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

ОК2 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

ОК5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК9 Пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

Цель работы: закрепить знания о оптимизации и рефакторинге.

**Задание:**

Задание 1: Анализ кода:

Задание: Проанализируйте предоставленный набор кода и выявите участки, которые могут быть оптимизированы или улучшены.

def bubble\_sort(arr):

n = len(arr)

for i in range(n):

for j in range(0, n-i-1):

if arr[j] > arr[j+1]:

arr[j], arr[j+1] = arr[j+1], arr[j]

returnarr

# Неоптимизированный код для сортировки выбором

defselection\_sort(arr):

n = len(arr)

for i in range(n):

min\_index = i

for j in range(i+1, n):

if arr[j] < arr[min\_index]:

min\_index = j

arr[i], arr[min\_index] = arr[min\_index], arr[i]

returnarr

# Неоптимизированный код для поиска максимального элемента в списке

def find\_max(arr):

max\_val = arr[0]

for num in arr:

if num > max\_val:

max\_val = num

returnmax\_val

Задание 2: Идентификация узких мест:

Задание: выявите участки кода, которые являются узкими местами в плане производительности, и предложите способы их оптимизации.

def factorial(n):

result = 1

for i in range(1, n+1):

result \*= i

returnresult

# Неоптимизированный код для суммирования элементов списка

def sum\_elements(arr):

total = 0

for num in arr:

total += num

returntotal

# Неоптимизированный код для поиска простых чисел

def find\_primes(n):

primes = []

for num in range(2, n):

if all(num % i != 0 for i in range(2, int(num \*\* 0.5) + 1)):

primes.append(num)

return primes

Задание 3: Рефакторингкода:

Задание: Проанализируйте участки вашего кода из предыдущих работ так, чтобы он был более читаемым, поддерживаемым и эффективным.

**Критерии оценки выполнения практических заданий.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе практики, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет по итогам окончания практики студентом не подготовлен.

**Тема 7. Проектирования пользовательских интерфейсов**

1. Что такое пользовательский интерфейс (UI) и какую роль он играет в программном обеспечении?
2. Какие основные принципы проектирования пользовательского интерфейса вы знаете?
3. Какие инструменты вы используете для создания прототипов пользовательского интерфейса?
4. Что такое пользовательский опыт (UX) и почему он важен при разработке пользовательского интерфейса?
5. Какие методы существуют для тестирования пользовательского интерфейса?
6. Какие факторы следует учитывать при выборе цветовой схемы для пользовательского интерфейса?
7. Какие принципы эргономики и удобства использования помогают создать удачный пользовательский интерфейс?
8. Какие тренды в дизайне пользовательского интерфейса сейчас наиболее актуальны?
9. Какие методы доступности следует учитывать при разработке пользовательского интерфейса для людей с ограниченными возможностями?
10. Какие основные компоненты и шаблоны вы используете при разработке пользовательского интерфейса?

**Критерии оценки устного опроса**

* оценка **«отлично»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
* оценка **«хорошо»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
* оценка **«удовлетворительно»** выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;

оценка **«неудовлетворительно»** выставляется студенту, если он не смог изложить суть основного вопроса

**Практическое занятие №7.** Проектирования пользовательских интерфейсов

ОК1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

ОК2 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

ОК5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК9 Пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

Цель занятия

Научиться проектировать пользовательские интерфейсы с использованием инструмента Figma, а также понять основные принципы UX/UI дизайна.

Задания

Изучение Figma: Ознакомление с интерфейсом и основными возможностями Figma.

Основы UX/UI дизайна: Понимание принципов пользовательского опыта (UX) и пользовательского интерфейса (UI).

Создание прототипа: Проектирование прототипа простого веб-приложения.

Работа с компонентами: Создание и использование компонентов в Figma.

Презентация и обратная связь: Презентация созданных прототипов и получение обратной связи.

Ход работы

Изучение Figma

Зарегистрируйтесь на сайте Figma и установите необходимые программы или используйте веб-версию.

Ознакомьтесь с интерфейсом Figma, пройдите встроенные учебные материалы и туториалы.

Основы UX/UI дизайна

Прочитайте основные материалы по UX/UI дизайну.

Ознакомьтесь с принципами хорошего пользовательского опыта (UX), такими как удобство использования, интуитивность, и эстетика.

Изучите основы UI-дизайна, такие как цветовая палитра, типографика, иерархия элементов и компоновка.

Создание прототипа

Создайте новый проект в Figma.

Разработайте прототип простого веб-приложения, например, "Список задач" (To-DoList). Прототип должен включать:

Главный экран со списком задач.

Экран добавления новой задачи.

Экран редактирования задачи.

Обратите внимание на пользовательский опыт, стараясь сделать интерфейс интуитивно понятным и удобным.

Работа с компонентами

Создайте компоненты для повторяющихся элементов интерфейса, таких как кнопки, поля ввода, карточки задач и т.д.

Используйте созданные компоненты для построения экранов приложения, что облегчит их редактирование и обновление.

Пример создания компонента кнопки:

Нарисуйте кнопку в Figma с текстом "Добавить задачу".

Выберите кнопку и нажмите "CreateComponent" (Создать компонент).

Назовите компонент "Button/AddTask".

Теперь вы можете использовать этот компонент на всех экранах, где требуется кнопка "Добавить задачу".

Презентация и обратная связь

Подготовьте короткую презентацию своего прототипа. Опишите, как пользователи будут взаимодействовать с вашим приложением.

Представьте свой прототип остальным студентам и преподавателю.

Получите обратную связь по своему дизайну и обсудите возможные улучшения.

Заключение

На данном практическом занятии студенты научатся использовать Figma для проектирования пользовательских интерфейсов, а также ознакомятся с основами UX/UI дизайна. Это важные навыки для создания удобных и привлекательных приложений.

**Критерии оценки выполнения практических заданий.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе практики, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет по итогам окончания практики студентом не подготовлен.

**Тема 8. Разработка пользовательского интерфейса.**

1. Что такое пользовательский интерфейс (UI) и какую роль он играет в программном обеспечении?
2. Какие основные принципы проектирования пользовательского интерфейса вы знаете?
3. Какие инструменты вы используете для создания прототипов пользовательского интерфейса?
4. Что такое пользовательский опыт (UX) и почему он важен при разработке пользовательского интерфейса?
5. Какие методы существуют для тестирования пользовательского интерфейса?
6. Какие факторы следует учитывать при выборе цветовой схемы для пользовательского интерфейса?
7. Какие принципы эргономики и удобства использования помогают создать удачный пользовательский интерфейс?
8. Какие тренды в дизайне пользовательского интерфейса сейчас наиболее актуальны?
9. Какие методы доступности следует учитывать при разработке пользовательского интерфейса для людей с ограниченными возможностями?
10. Какие основные компоненты и шаблоны вы используете при разработке пользовательского интерфейса?

**Критерии оценки устного опроса**

* оценка **«отлично»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
* оценка **«хорошо»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
* оценка **«удовлетворительно»** выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;

оценка **«неудовлетворительно»** выставляется студенту, если он не смог изложить суть основного вопроса

**Практическое занятие №9. Разработка** пользовательского интерфейса.

ОК1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

ОК2 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

ОК5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК9 Пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

Цель занятия

Научиться создавать пользовательские интерфейсы с использованием VisualStudio и. NETFramework, а также ознакомиться с основами проектирования интерфейсов для настольных приложений.

Задания

Изучение VisualStudio: Ознакомление с интерфейсом и основными возможностями среды разработки.

Основы проектирования пользовательского интерфейса: Понимание принципов UX/UI дизайна.

Создание простого приложения: Разработка настольного приложения с несколькими экранами.

Работа с элементами управления: Изучение и использование стандартных элементов управления.

Презентация и обратная связь: Презентация разработанного приложения и получение обратной связи.

Ход работы

Изучение VisualStudio

Установите VisualStudioCommunityEdition, если она еще не установлена. Скачать VisualStudio

Ознакомьтесь с интерфейсом VisualStudio, пройдите встроенные учебные материалы и туториалы.

Основы проектирования пользовательского интерфейса

Прочитайте основные материалы по UX/UI дизайну.

Изучите принципы проектирования удобных и интуитивно понятных интерфейсов.

Ознакомьтесь с основами создания настольных приложений на платформе .NET.

Создание простого приложения

Создайте новый проект в VisualStudio, выбрав шаблон "WindowsFormsApp (. NETFramework)".

Разработайте приложение "Список задач" (To-DoList). Приложение должно включать:

Главный экран со списком задач.

Экран добавления новой задачи.

Экран редактирования задачи.

Обратите внимание на удобство использования и эстетический вид интерфейса.

Работа с элементами управления

Изучите и используйте стандартные элементы управления WindowsForms, такие как кнопки, текстовые поля, списки, метки и панели.

Добавьте в проект элементы управления для создания и редактирования задач.

Пример кода для главного экрана:

csharp

using System;

using System.Windows.Forms;

namespace ToDoListApp

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

}

private void btnAddTask\_Click(object sender, EventArgs e)

{

AddTaskForm addTaskForm = new AddTaskForm();

addTaskForm.ShowDialog();

}

}

}

Создание экранов для добавления и редактирования задач

Создайте форму для добавления новой задачи (AddTaskForm) и форму для редактирования задачи (EditTaskForm).

Добавьте соответствующие элементы управления и напишите обработчики событий для сохранения и редактирования задач.

Пример кода для формы добавления задачи:

csharp

using System;

using System.Windows.Forms;

namespace ToDoListApp

{

public partial class AddTaskForm : Form

{

public AddTaskForm()

{

InitializeComponent();

}

private void btnSave\_Click(object sender, EventArgs e)

{

// Логика сохранения новой задачи

this.Close();

}

}

}

Презентация и обратная связь

Подготовьте короткую презентацию своего приложения. Опишите функционал и интерфейс.

Представьте свое приложение остальным студентам и преподавателю.

Получите обратную связь по вашему проекту и обсудите возможные улучшения.

Заключение

На данном практическом занятии студенты научатся создавать пользовательские интерфейсы с использованием VisualStudio и. NETFramework. Это важные навыки для разработки настольных приложений, которые помогут создавать удобные и привлекательные приложения.

**Критерии оценки выполнения практических заданий.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе практики, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет по итогам окончания практики студентом не подготовлен.

**Тема 9. Разработка приложения. Работа с БД. Доступ к данным.**

1. Что такое база данных и для чего она используется?
2. Что такое таблица в базе данных и как она выглядит?
3. Что такое SQL и зачем он нужен при работе с базами данных?
4. Как добавить новую запись (строку) в таблицу базы данных?
5. Как найти и выбрать данные из таблицы базы данных?
6. Что такое связь между таблицами и зачем она нужна?
7. Как подключиться к базе данных из программы на примере любого языка программирования?
8. Что такое ORM и как он помогает работать с базой данных?
9. Как защитить базу данных от неправильных запросов и хакерских атак?
10. Как сохранять изменения в базе данных, чтобы они не потерялись?

**Критерии оценки устного опроса**

* оценка **«отлично»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
* оценка **«хорошо»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
* оценка **«удовлетворительно»** выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;

оценка **«неудовлетворительно»** выставляется студенту, если он не смог изложить суть основного вопроса

**Практическое занятие №9. Разработка приложения**

ОК1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

ОК2 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

ОК5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК9 Пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

Цель занятия

Научиться создавать настольные приложения с использованием языка программирования C# и платформы .NET, а также познакомиться с основами проектирования, разработки и тестирования приложений.

Задания

Изучение основ C# и .NET: Ознакомление с основными концепциями языка программирования C# и платформы .NET.

Проектирование приложения: Разработка плана и архитектуры приложения.

Реализация функционала: Написание кода для основного функционала приложения.

Работа с базой данных: Подключение и взаимодействие с базой данных.

Тестирование и отладка: Тестирование приложения и устранение ошибок.

Презентация и обратная связь: Презентация разработанного приложения и получение обратной связи.

Ход работы

Изучение основ C# и .NET

Ознакомьтесь с основными концепциями языка программирования C# и платформы .NET.

Пройдите короткие учебные материалы и туториалы по C# и .NET, чтобы понять синтаксис и основные возможности языка.

Проектирование приложения

Определите цель и функциональные требования приложения. Например, создайте приложение для управления списком задач (To-Do List).

Разработайте план и архитектуру приложения. Определите основные компоненты и модули приложения.

Реализация функционала

Создайте новый проект в Visual Studio, выбрав шаблон "Windows Forms App (.NET Framework)".

Разработайте основной функционал приложения:

Главный экран со списком задач.

Экран добавления новой задачи.

Экран редактирования задачи.

Возможность удаления задачи.

Пример кода для главного экрана:

csharp

Копироватькод

using System;

using System.Windows.Forms;

namespace ToDoListApp

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

}

private void btnAddTask\_Click(object sender, EventArgs e)

{

AddTaskForm addTaskForm = new AddTaskForm();

addTaskForm.ShowDialog();

}

private void btnEditTask\_Click(object sender, EventArgs e)

{

// Логика для редактирования выбранной задачи

}

private void btnDeleteTask\_Click(object sender, EventArgs e)

{

// Логика для удаления выбранной задачи

}

}

}

Создание экранов для добавления и редактирования задач

Создайте форму для добавления новой задачи (AddTaskForm) и форму для редактирования задачи (EditTaskForm).

Добавьте соответствующие элементы управления и напишите обработчики событий для сохранения и редактирования задач.

Пример кода для формы добавления задачи:

csharp

Копироватькод

using System;

using System.Windows.Forms;

namespace ToDoListApp

{

public partial class AddTaskForm : Form

{

public AddTaskForm()

{

InitializeComponent();

}

private void btnSave\_Click(object sender, EventArgs e)

{

// Логика сохранения новой задачи

this.Close();

}

}

}

Работа с базой данных

Создайте базу данных для хранения задач, используя SQL Server или другую СУБД.

Подключите приложение к базе данных и реализуйте CRUD операции (создание, чтение, обновление, удаление).

Пример кода для подключения к базе данных и выполнения операций:

csharp

Копироватькод

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace ToDoListApp

{

public partial class MainForm : Form

{

private string connectionString = "your\_connection\_string";

private void LoadTasks()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlDataAdapter adapter = new SqlDataAdapter("SELECT \* FROM Tasks", connection);

DataTable tasks = new DataTable();

adapter.Fill(tasks);

// Привязка данных к элементу управления, например, DataGridView

}

}

private void AddTask(string taskName)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand command = new SqlCommand("INSERT INTO Tasks (Name) VALUES (@Name)", connection);

command.Parameters.AddWithValue("@Name", taskName);

connection.Open();

command.ExecuteNonQuery();

}

}

// Аналогично добавьте методы для редактирования и удаления задач

}

}

Тестирование и отладка

Проведите тестирование приложения на наличие ошибок и багов.

Используйте встроенные инструменты Visual Studio для отладки кода и устранения ошибок.

Презентация и обратная связь

Подготовьте презентацию своего приложения. Опишите функционал и архитектуру.

Представьте свое приложение остальным студентам и преподавателю.

Получите обратную связь по вашему проекту и обсудите возможные улучшения.

Заключение

На данном практическом занятии студенты научатся создавать настольные приложения с использованием языка программирования C# и платформы .NET. Это важные навыки для разработки современных приложений

**Практическое занятие №10. Работа с Базой данных**

ОК1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

ОК2 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

ОК5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК9 Пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

Цель занятия

Научиться работать с базами данных в среде разработки Visual Studio, используя язык программирования C# и платформу .NET, а также ознакомиться с основами создания, подключения и выполнения операций с базами данных.

Задания

Изучение основ работы с базами данных в Visual Studio.

Создание базы данных и таблиц.

Подключение к базе данных из C# приложения.

Реализация CRUD операций (создание, чтение, обновление, удаление).

Работа с Entity Framework для упрощения работы с базами данных.

Тестирование и отладка приложения.

Презентация и обсуждение результатов.

Ход работы

Изучение основ работы с базами данных в Visual Studio

Ознакомьтесь с основами SQL и реляционных баз данных.

Изучите основные концепции работы с базами данных в Visual Studio.

Пройдите краткие учебные материалы по работе с базами данных в C# и .NET.

Создание базы данных и таблиц

Запустите SQL Server Management Studio (SSMS) или используйте встроенные инструменты Visual Studio для управления базами данных.

Создайте новую базу данных под названием ToDoListDB.

Создайте таблицу Tasks со следующими полями:

Id (int, Primary Key, Identity)

Name (nvarchar(100))

Description (nvarchar(255))

DueDate (datetime)

IsCompleted (bit)

Пример SQL-кода для создания таблицы:

sql

Копироватькод

CREATE DATABASE ToDoListDB;

GO

USE ToDoListDB;

GO

CREATE TABLE Tasks (

Id INT PRIMARY KEY IDENTITY,

Name NVARCHAR(100) NOT NULL,

Description NVARCHAR(255),

DueDate DATETIME,

IsCompleted BIT

);

Подключение к базе данных из C# приложения

Создайте новый проект в Visual Studio, выбрав шаблон "Windows Forms App (.NET Framework)".

Добавьте подключение к базе данных в файле App.config:

xml

Копироватькод

<configuration>

<connectionStrings>

<add name="ToDoListDB"

connectionString="Server=your\_server\_name;Database=ToDoListDB;Integrated Security=True;"

providerName="System.Data.SqlClient" />

</connectionStrings>

</configuration>

Реализация CRUD операций

Создайте класс Task для представления данных:

csharp

Копироватькод

public class Task

{

public int Id { get; set; }

public string Name { get; set; }

public string Description { get; set; }

public DateTime DueDate { get; set; }

public bool IsCompleted { get; set; }

}

Реализуйте методы для выполнения CRUD операций:

csharp

Копироватькод

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

public class TaskRepository

{

private string connectionString;

public TaskRepository(string connectionString)

{

this.connectionString = connectionString;

}

public void AddTask(Task task)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "INSERT INTO Tasks (Name, Description, DueDate, IsCompleted) VALUES (@Name, @Description, @DueDate, @IsCompleted)";

SqlCommand command = new SqlCommand(query, connection);

command.Parameters.AddWithValue("@Name", task.Name);

command.Parameters.AddWithValue("@Description", task.Description);

command.Parameters.AddWithValue("@DueDate", task.DueDate);

command.Parameters.AddWithValue("@IsCompleted", task.IsCompleted);

connection.Open();

command.ExecuteNonQuery();

}

}

public List<Task> GetTasks()

{

List<Task> tasks = new List<Task>();

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT \* FROM Tasks";

SqlCommand command = new SqlCommand(query, connection);

connection.Open();

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

Task task = new Task

{

Id = (int)reader["Id"],

Name = (string)reader["Name"],

Description = (string)reader["Description"],

DueDate = (DateTime)reader["DueDate"],

IsCompleted = (bool)reader["IsCompleted"]

};

tasks.Add(task);

}

}

return tasks;

}

public void UpdateTask(Task task)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "UPDATE Tasks SET Name = @Name, Description = @Description, DueDate = @DueDate, IsCompleted = @IsCompleted WHERE Id = @Id";

SqlCommand command = new SqlCommand(query, connection);

command.Parameters.AddWithValue("@Id", task.Id);

command.Parameters.AddWithValue("@Name", task.Name);

command.Parameters.AddWithValue("@Description", task.Description);

command.Parameters.AddWithValue("@DueDate", task.DueDate);

command.Parameters.AddWithValue("@IsCompleted", task.IsCompleted);

connection.Open();

command.ExecuteNonQuery();

}

}

public void DeleteTask(int taskId)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "DELETE FROM Tasks WHERE Id = @Id";

SqlCommand command = new SqlCommand(query, connection);

command.Parameters.AddWithValue("@Id", taskId);

connection.Open();

command.ExecuteNonQuery();

}

}

}

Работас Entity Framework

Добавьте Entity Framework впроектчерез NuGet Package Manager.

Создайте модель данных с использованием Entity Framework и настройте контекст базы данных.

Пример кода для работы с Entity Framework:

csharp

Копироватькод

using System.Data.Entity;

public class ToDoListContext : DbContext

{

public DbSet<Task> Tasks { get; set; }

public ToDoListContext() : base("name=ToDoListDB")

{

}

}

Тестирование и отладка

Проведите тестирование всех реализованных функций приложения.

Используйте отладчик Visual Studio для устранения выявленных ошибок.

**Практическое занятие №11. Доступ к данным.**

ОК1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

ОК2 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

ОК5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК9 Пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

Цель занятия

Научиться осуществлять доступ к данным в базе данных из приложения на C# с использованием среды разработки Visual Studio. Это включает подключение к базе данных, выполнение запросов и отображение данных.

Задания

Изучение способов подключения к базе данных в C#.

Создание подключения к базе данных.

Выполнение запросов к базе данных.

Отображение данных в приложении.

Реализация CRUD операций.

Тестирование и отладка приложения.

Ход работы

Изучение способов подключения к базе данных в C#

Прочитайте материалы о различных способах подключения к базе данных в C#, таких как ADO.NET и Entity Framework.

Ознакомьтесь с примерами использования ADO.NET и Entity Framework для доступа к данным.

Создание подключения к базе данных

Создайте новый проект в Visual Studio, выбрав шаблон "Windows Forms App (.NET Framework)".

Добавьте подключение к базе данных в файле App.config или web.config:

xml

Копироватькод

<configuration>

<connectionStrings>

<add name="ToDoListDB"

connectionString="Server=your\_server\_name;Database=ToDoListDB;Integrated Security=True;"

providerName="System.Data.SqlClient" />

</connectionStrings>

</configuration>

Выполнение запросов к базе данных

Создайте класс для работы с базой данных. Реализуйте методы для выполнения запросов к базе данных.

Пример кода для подключения к базе данных и выполнения запросов с использованием ADO.NET:

csharp

Копироватькод

using System;

using System.Data;

using System.Data.SqlClient;

public class DatabaseHelper

{

private string connectionString;

public DatabaseHelper(string connectionString)

{

this.connectionString = connectionString;

}

public DataTable ExecuteQuery(string query)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlDataAdapter adapter = new SqlDataAdapter(query, connection);

DataTable dataTable = new DataTable();

adapter.Fill(dataTable);

return dataTable;

}

}

public void ExecuteNonQuery(string query)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand command = new SqlCommand(query, connection);

connection.Open();

command.ExecuteNonQuery();

}

}

}

Отображениеданныхвприложении

Создайте форму для отображения данных, например, MainForm.

Добавьте элемент управления DataGridView для отображения данных из базы данных.

Напишите код для заполнения DataGridView данными из базы данных.

Пример кода для отображения данных в DataGridView:

csharp

Копироватькод

using System;

using System.Data;

using System.Windows.Forms;

public partial class MainForm : Form

{

private DatabaseHelper db;

public MainForm()

{

InitializeComponent();

db = new DatabaseHelper("your\_connection\_string");

LoadData();

}

private void LoadData()

{

string query = "SELECT \* FROM Tasks";

DataTable dataTable = db.ExecuteQuery(query);

dataGridView1.DataSource = dataTable;

}

}

Реализация CRUD операций

Добавьте формы для добавления, редактирования и удаления записей.

Реализуйте соответствующие методы в DatabaseHelper для выполнения операций INSERT, UPDATE и DELETE.

Пример кода для добавления записи:

csharp

Копироватькод

public void AddTask(string name, string description, DateTime dueDate, bool isCompleted)

{

string query = "INSERT INTO Tasks (Name, Description, DueDate, IsCompleted) VALUES (@Name, @Description, @DueDate, @IsCompleted)";

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand command = new SqlCommand(query, connection);

command.Parameters.AddWithValue("@Name", name);

command.Parameters.AddWithValue("@Description", description);

command.Parameters.AddWithValue("@DueDate", dueDate);

command.Parameters.AddWithValue("@IsCompleted", isCompleted);

connection.Open();

command.ExecuteNonQuery();

}

}

Примеркодадляобновлениязаписи:

csharp

Копироватькод

public void UpdateTask(int id, string name, string description, DateTime dueDate, bool isCompleted)

{

string query = "UPDATE Tasks SET Name = @Name, Description = @Description, DueDate = @DueDate, IsCompleted = @IsCompleted WHERE Id = @Id";

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand command = new SqlCommand(query, connection);

command.Parameters.AddWithValue("@Id", id);

command.Parameters.AddWithValue("@Name", name);

command.Parameters.AddWithValue("@Description", description);

command.Parameters.AddWithValue("@DueDate", dueDate);

command.Parameters.AddWithValue("@IsCompleted", isCompleted);

connection.Open();

command.ExecuteNonQuery();

}

}

Примеркодадляудалениязаписи:

csharp

Копироватькод

public void DeleteTask(int id)

{

string query = "DELETE FROM Tasks WHERE Id = @Id";

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand command = new SqlCommand(query, connection);

command.Parameters.AddWithValue("@Id", id);

connection.Open();

command.ExecuteNonQuery();

}

}

Тестированиеиотладкаприложения

Проверьте функциональность всех реализованных операций.

Используйте отладчик Visual Studio для устранения выявленных ошибок.

**Критерии оценки выполнения практических заданий.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе практики, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет по итогам окончания практики студентом не подготовлен.

**Тема 10. Отладка и тестирование программного обеспечения**

1. Что такое отладка (debugging) и зачем она нужна в процессе разработки программного обеспечения?
2. Какие существуют методы и инструменты для отладки программного кода?
3. Что такое точка останова (breakpoint) и как она используется при отладке?
4. Какие типичные ошибки и баги могут возникать в программном обеспечении?
5. Что такое тестирование программного обеспечения и почему оно важно?
6. Какие существуют виды тестирования (например, модульное, интеграционное, системное, приемочное)?
7. Как написать простейший модульный тест на примере любого языка программирования?
8. Что такое тестирование черного и белого ящика и в чем их различия?
9. Как автоматизировать тестирование и какие инструменты для этого используются?
10. Как анализировать и интерпретировать результаты тестирования для улучшения качества программного обеспечения?

**Критерии оценки устного опроса**

* оценка **«отлично»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
* оценка **«хорошо»** выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
* оценка **«удовлетворительно»** выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;

оценка **«неудовлетворительно»** выставляется студенту, если он не смог изложить суть основного вопроса

**Практическое занятие №12. Отладка и тестирование программного обеспечения**

ОК1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

ОК2 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

ОК5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК9 Пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

Цель занятия

Научить студентов правильно отлаживать и тестировать программное обеспечение для обеспечения его корректной работы и выявления ошибок.

Задания

Изучение основных принципов отладки и тестирования программного обеспечения.

Отладка приложения в среде Visual Studio.

Тестирование функциональности приложения.

Тестирование пользовательского интерфейса.

Оценка и устранение обнаруженных ошибок.

Презентация результатов и обсуждение опыта.

Ход работы

Изучение основных принципов отладки и тестирования ПО

Рассмотрите основные принципы отладки, включая поиск, анализ и исправление ошибок в коде.

Изучите различные методы тестирования программного обеспечения, такие как модульное тестирование, интеграционное тестирование, системное тестирование и т. д.

Познакомьтесь с инструментами и техниками для тестирования пользовательского интерфейса.

Отладка приложения в среде Visual Studio

Запустите проект в Visual Studio.

Используйте отладчик для пошагового выполнения кода, отслеживания значений переменных и выявления ошибок.

Проанализируйте сообщения об ошибках и используйте инструменты Visual Studio для их исправления.

Тестирование функциональности приложения

Разработайте набор тестов, охватывающих основную функциональность вашего приложения.

Запустите тесты и оцените результаты. Выявите обнаруженные ошибки и замечания.

Тестирование пользовательского интерфейса

Проведите тестирование пользовательского интерфейса, проверяя работоспособность элементов управления и взаимодействие с пользователем.

Зафиксируйте любые недочеты или несоответствия дизайну.

Оценка и устранение обнаруженных ошибок

Проведите анализ обнаруженных ошибок и проблем.

Используйте полученную информацию для внесения соответствующих изменений в код или дизайн приложения.

Повторите процесс тестирования и отладки для проверки исправлений.

**Критерии оценки выполнения практических заданий.**

**«5» «отлично»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
* при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

**«4» «хорошо»**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
* оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
* при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
* характеристики студента положительные, имеются отдельные замечания и рекомендации;

**«3» «удовлетворительно»**

* при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
* в отчете по практическим работам содержится описание видов выполняемых работ согласно программе практики, однако, записи неполные и неточные;
* отчет по практическим работам оформлен с отклонениями от установленных требований;
* при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет по итогам окончания практики студентом не подготовлен.

# **4.Контрольно-оценочные материалы для промежуточной аттестации по учебной дисциплине –зачет.**

**Зачёт**

* студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
* все работы выполнил правильно и уверенно;
* проявил активность в освоении новых практических умений;
* содержание отчета по практике полноценно отражает информацию по выполненным видам работ, установленных программой практики;
* оформление отчета по учебной практике полностью соответствует предъявляемым требованиям;
* при выполнении программы практики студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;
* программа учебной практики реализована полностью.

**Незачёт**

– студент не справился с запланированными видами работ;

– при выполнении программы практики студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет по итогам учебной практики студентом не подготовлен.

**Дополнения и изменения к комплекту ФОС на учебный год**

Дополнения и изменения к комплекту ФОС на \_\_\_\_\_\_\_\_\_\_ учебный год по дисциплине Наименование\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

В комплект ФО внесены следующие изменения:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дополнения и изменения в комплекте ФОС обсуждены на заседании ЦК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Протокол № \_\_\_\_\_\_ от «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_\_г.

Председатель ЦК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_