


Федеральное государственное образовательное бюджетное учреждение  
высшего образования

**«Финансовый университет при Правительстве Российской Федерации»  
(Финансовый университет)**

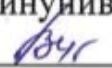
**Красноярский филиал Финуниверситета**

---

(наименование структурного подразделения)

СОГЛАСОВАНО  
ООО НПП "Авакс -Геосервис"  
Директор по инновациям  
  
В.В. Кретинин  
«04» сентябре 2025 г.



УТВЕРЖДАЮ  
Заместитель директора по  
учебно-методической работе  
Красноярского филиала  
Финуниверситета  
  
О.С. Вергейчик  
«04» сентябре 2025 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**  
по профессиональному модулю

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

---

(код, наименование)

**09.02.07 Информационные системы и программирование**

---

(код, наименование специальности)

Красноярск – 2025 г.

Фонд оценочных средств по профессиональному модулю разработан на основании федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 «Информационные системы и программирование»

Составители:

Дядичкина Валерия Витальевна, преподаватель

(фамилия, имя, отчество, наименование должности, квалификационной категории)

Гетт Полина Дмитриевна, преподаватель

(фамилия, имя, отчество, наименование должности, квалификационной категории)

Фонд оценочных средств по профессиональному модулю рассмотрен и рекомендован к утверждению на заседании предметной (цикловой) комиссии

Общепрофессиональных дисциплин

(наименование)

Протокол от «04» сентября 2025 г. № 1

Председатель предметной (цикловой)  
комиссии

  
(подпись)

О.А. Полтавец  
(инициалы, фамилия)

1. Паспорт фонда оценочных средств по профессиональному модулю «ПМ.01 Разработка модулей программного обеспечения для компьютерных систем»

(код, наименование)

09.02.07 «Информационные системы и программирование»

(код, наименование специальности)

Результаты обучения (знания, умения)	Общие и профессиональные компетенции	Междисциплинарный курс	Темы	Наименование оценочного средства	
				Текущий контроль	Промежуточная аттестация
Уметь: работать на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства. Знать: основные этапы разработки программного обеспечения; основные принципы	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09. ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6	МДК.01.01 Разработка программных модулей	Тема 1.1 Жизненный цикл ПО	Защита отчетов по практическим и лабораторным работам, опрос, тест.	Экзамен
			Тема 1.2 Структурное программирование		
			Тема 1.3 Объектно-ориентированное программирование		
			Тема 1.4 Паттерны проектирования		
			Тема 1.5 Событийно-управляемое программирование		
			Тема 1.6 Оптимизация и рефакторинг кода		
			Тема 1.7 Разработка пользовательского интерфейса.		
			Тема 1.8 Основы ADO.Net		
		МДК.01.02 Поддержка и тестирование программных	Тема 2.1 Основы тестирования программного	Защита отчетов по практическим и	

технологии структурного и объектно- ориентированног о программирован ия; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов		модулей	обеспечения	лабораторным работам, опрос, тест.	Дифференциро ванный зачет
			Тема 2.2 Процесс тестирования		
			Тема 2.3 Методы и техники тестирования		
			Тема 2.4 Функциональ ное тестирование		
			Тема 2.5 Нефункциона льное тестирование		
			Тема 2.6 Интеграционн ое и системное тестирование		
			Тема 2.7 Приемочное тестирование		
			Тема 2.8 Управление дефектами		
			Тема 2.9 Автоматизаци я тестирования		
			Тема 2.10 Тестирование API		
			Тема 2.11 Внедрение и эксплуатация ПО		
			Тема 2.12 Обслуживани е и развитие ПО		

		МДК.01.03 Разработка мобильных приложений	Тема 3.1 Введение в мобильную разработку	Защита отчетов по практическим и лабораторным работам, опрос, тест.	Экзамен
			Тема 3.2 Основы разработки под Android/iOS		
			Тема 3.3 Пользователь ский интерфейс мобильного приложения		
			Тема 3.4 Хранение данных		
			Тема 3.5 Работа с сетевыми данными		
			Тема 3.6 Используй е ресурсов и локализация		
			Тема 3.7 Работа с камерой и галереей		
			Тема 3.8 Геолокация и карты		
			Тема 3.9 Уведомления и фоновые процессы		

			Тема 3.10 Сенсоры устройства		
			Тема 3.11 Тестирование и отладка мобильных приложений		
			Тема 3.12 Публикация мобильных приложений		
			Тема 3.13 Обновление и поддержка		
		МДК.01.04 Системное программировани е	Тема 4.1 Сборка ПО и переменные окружения	Защита отчетов по практическим и лабораторным работам, опрос, тест.	Экзамен
			Тема 4.2 Системы сборки, библиотеки и Модель Клиент- Интерфейс- Сервер		
			Тема 4.3 Введение в Ассемблер		
			Тема 4.4 Низкоуровнев ый ввод- вывод и файловые дескрипторы		
			Тема 4.5 Файловые атрибуты, ссылки и прос-система		
			Тема 4.5 Каналы, сообщения и разделяемая память		
			Тема 4.6 Сокеты и		

			синхронизаци я		
			Тема 4.7 Сигналы		
			Тема 4.8 Процессы и потоки		
			Тема 4.9 Многопоточн ое программиров ание		
			Тема 4.10 Демоны и консольный ввод-вывод		
			Тема 4.11 Программиро вание графического интерфейса с помощью GTK+		

## 2. Формы промежуточной аттестации по профессиональному модулю

Элементы профессионального модуля	Форма промежуточной аттестации	
	Промежуточная аттестация	Промежуточный контроль
МДК.01.01 Разработка программных модулей	Экзамен	Оценка выполненных контрольных заданий внеаудиторной самостоятельной работы
МДК.01.02 Поддержка и тестирование программных модулей	Дифференцированный зачет	Оценка выполненных контрольных заданий внеаудиторной самостоятельной работы
МДК.01.03 Разработка мобильных приложений	Экзамен	Оценка выполненных контрольных заданий внеаудиторной самостоятельной работы
МДК.01.04 Системное программирование	Экзамен, защита курсовой работы	Оценка выполненных контрольных заданий внеаудиторной самостоятельной работы

УП	Дифференцированный зачет	Оценка практических умений, аттестационный лист студента по УП
ПП	Дифференцированный зачет	Оценка освоения профессиональных и динамики освоения общих компетенций, Аттестационный лист студента по ПП, характеристика профессиональной деятельности студента в период производственной практики
ПМ	Экзамен по модулю (квалификационный экзамен)	

### 3. Комплект оценочных средств

#### «МДК.01.01 Разработка программных модулей» Тема 1.1 Жизненный цикл ПО

Перечень вопросов по теме для устного опроса:

- 1) Что такое жизненный цикл программного обеспечения?
- 2) Перечислите фазы жизненного цикла ПО.
- 3) Перечислите основные модели жизненного цикла ПО.
- 4) Перечислите стратегии жизненного цикла ПО.

#### Тема 1.2 Структурное программирование

Перечень вопросов по теме для устного опроса:

- 1) Что такое структурное программирование и зачем оно нужно?
- 2) Какие преимущества даёт модульное программирование при разработке больших программных систем?
- 3) Какие три основных управляющих структуры используются в структурном программировании?
- 4) Чем отличается цикл while от цикла for?
- 5) Что делает оператор if в программе?

#### Тема 1.3 Объектно-ориентированное программирование

Перечень вопросов по теме для устного опроса:

- 1) Что такое инкапсуляция?
- 2) Что такое наследование?
- 3) Что такое полиморфизм?

#### Тема 1.4 Паттерны проектирования

Перечень вопросов по теме для устного опроса:

- 1) Что такое паттерн проектирования и зачем он нужен в программировании?
- 2) На какие три основные группы делятся паттерны, и чем занимается каждая из них?
- 3) Какой паттерн помогает создать только один экземпляр класса и контролировать его доступ?
- 4) Как работает паттерн Observer, если представить его на примере подписки на канал в интернете?
- 5) Чем паттерн Strategy помогает менять поведение объекта, например, способ оплаты в интернет-магазине?

Практическая работа. Назначение и виды паттернов проектирования.

Цель работы: Понять назначение паттернов проектирования, изучить их основные виды.

**Паттерн проектирования** — это повторяемое решение типовой задачи в разработке программного обеспечения. Это не готовый код, а шаблон проектирования, описывающий, как структурировать классы и объекты для достижения гибкости, читаемости и поддерживаемости кода.

Паттерны делятся на три основные группы:

**Порождающие** - отвечают за создание объектов. Автоматизируют создание объектов в программе: вместо того чтобы создавать каждый объект вручную, разработчик использует готовый класс-конструктор.

**Структурные** - определяют, как компоновать объекты и классы в более крупные структуры. Помогают решать задачи, связанные с композицией объектов, агрегацией и наследованием.

**Поведенческие** - определяют, как объекты взаимодействуют между собой и каким образом они выполняют свои функции. Помогают управлять взаимодействием между объектами, управлять распределёнными системами и обеспечивать обратную связь в системе.

**Задание:** Создайте консольное приложение на C#, в котором будут реализованы данные паттерны.

Когда Вы запустите среду разработки, выберите опцию «Создание проекта»:

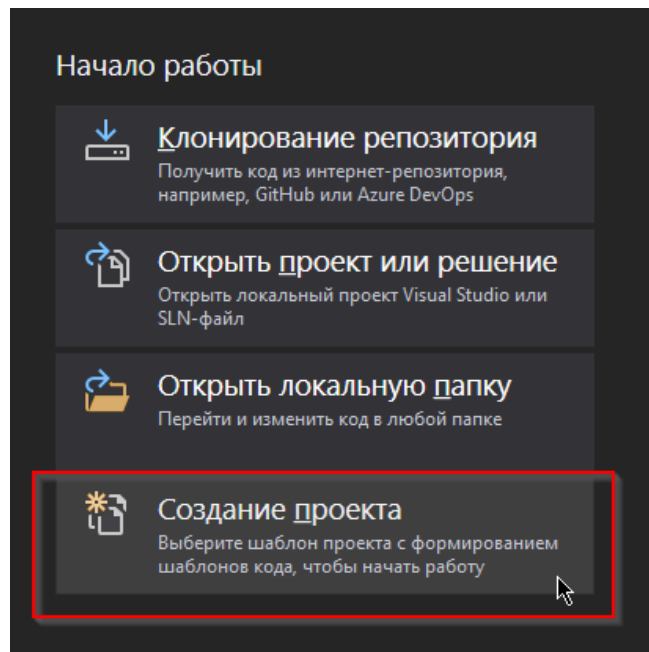


Рисунок 1 – Начало работы

В окне «Создание проекта» Вы увидите множество опций, в частности, первый выпадающий список, в котором перечислены доступные языки. Нам нужно выбрать C# из списка:

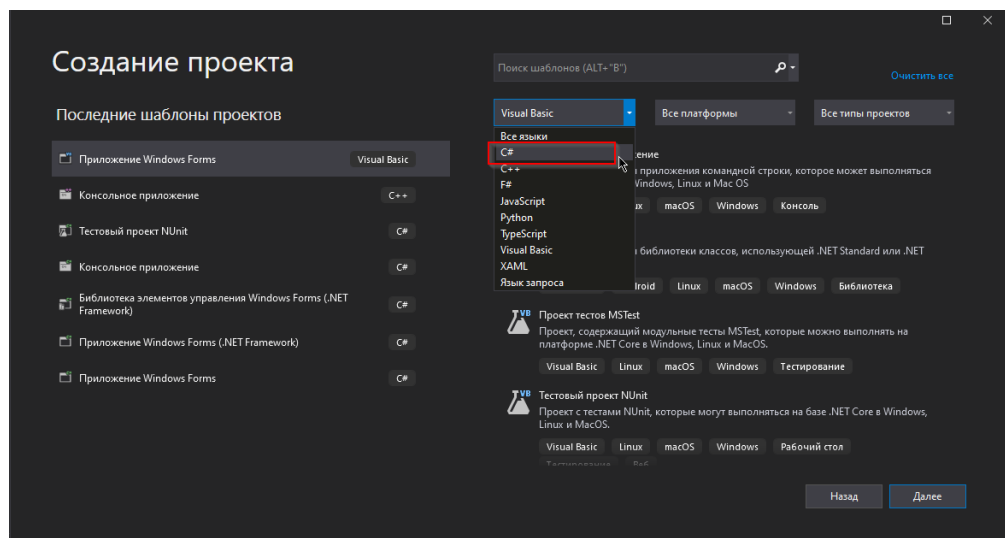


Рисунок 2 – Выбор языка

После выбора языка C# в качестве целевого для нашего проекта выбираем тип нашего будущего проекта – «Консольное приложение», как показано ниже на рисунке, и жмём «Далее»:

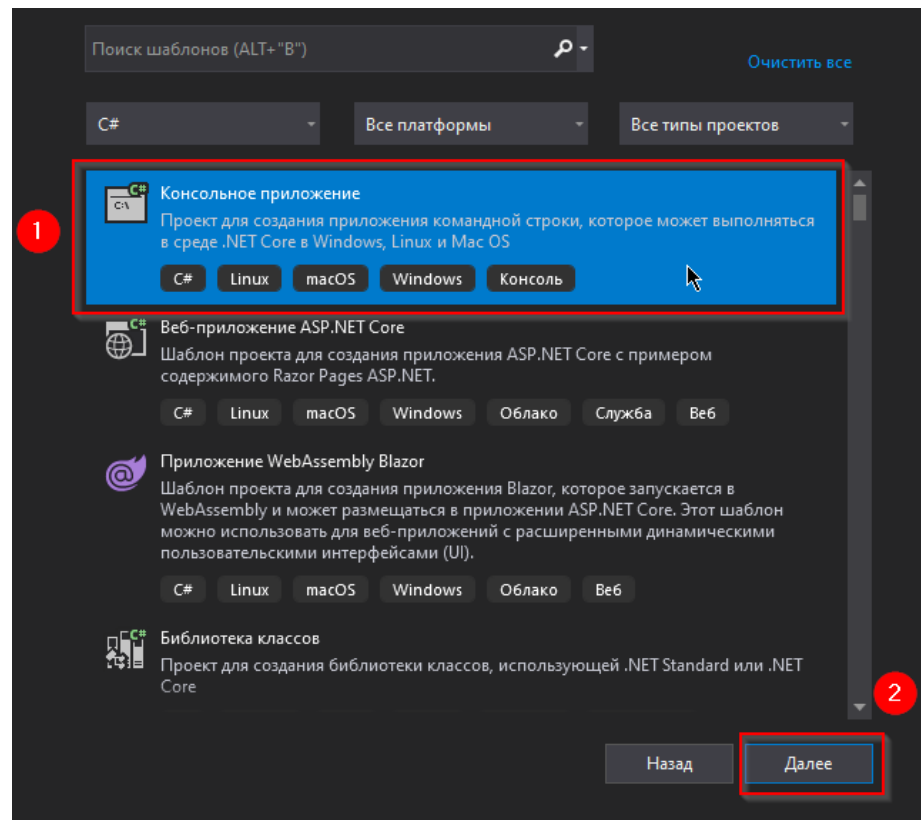


Рисунок 3 – Тип проекта

На следующем шаге нужно указать имя нашего будущего проекта.

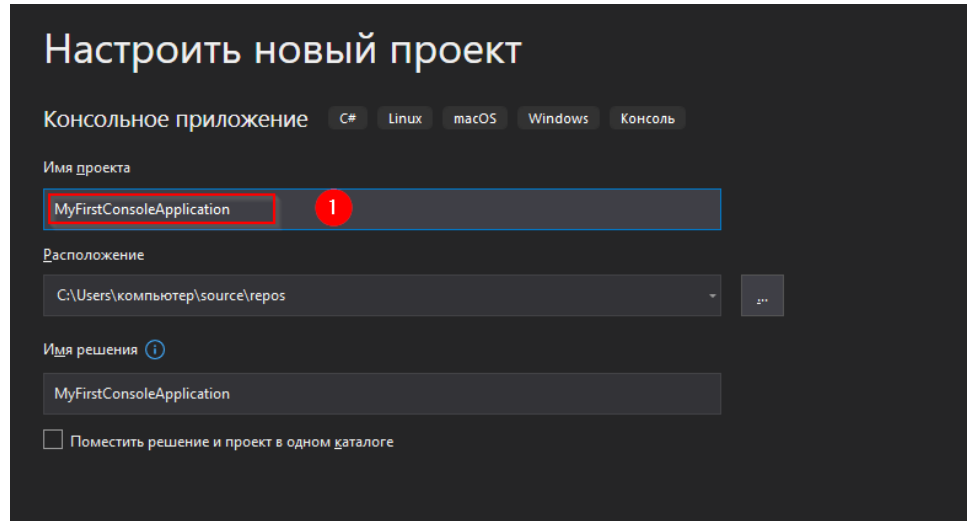


Рисунок 4 – Настройка имени проекта

Когда все настройки выполнены, нажимаем кнопку «Создать».

### 1. Порождающий паттерн: Singleton

Задача: Создать класс `Logger`, который будет записывать сообщения в консоль. Убедитесь, что в программе может существовать только один экземпляр логгера.

### 2. Структурный паттерн: Decorator

Задача: Реализовать систему заказа пиццы с возможностью добавления дополнительных ингредиентов (например, сыр, бекон) без изменения базового класса.

### 3. Поведенческий паттерн: Observer

Задача: Реализовать систему уведомлений о смене погоды. Когда температура меняется, все подписчики (например, термометр, уведомление на телефон) получают обновление.

## Тема 1.5 Событийно-управляемое программирование

Перечень вопросов по теме для устного опроса:

- 1) Что такое событийно-управляемое программирование
- 2) Какой элемент управления используется для ввода текста пользователем в однострочное поле?
- 3) Что такое диалоговое окно?
- 4) Что такое обработчик события, и как он связывается с элементом управления?
- 5) Перечислите основные элементы WPF.

## Тема 1.6 Оптимизация и рефакторинг кода

Перечень вопросов по теме для теста:

- 1) Какова основная цель рефакторинга кода?
  - а) Увеличить количество строк кода
  - б) Улучшить структуру и читаемость кода без изменения его поведения**
  - в) Переписать программу на другом языке
  - г) Скрыть ошибки от пользователя
- 2) Что такое оптимизация производительности программы?
  - а) Удаление всех комментариев
  - б) Ускорение работы программы и снижение потребления ресурсов**
  - в) Изменение интерфейса
  - г) Упрощение логики до минимального функционала
- 3) Какой из перечисленных методов является примером рефакторинга?
  - а) Добавление новой функции
  - б) Переименование метода для лучшего понимания его назначения**
  - в) Удаление тестов
  - г) Увеличение количества циклов
- 4) Для чего используется профилирование (profiling) в разработке?
  - а) Для автоматического написания кода
  - б) Для анализа производительности: времени выполнения, использования памяти**
  - в) Для компиляции программы
  - г) Для резервного копирования файлов

5) Какой инструмент в Visual Studio помогает автоматически переименовывать переменные и методы во всём проекте?

а) Компилятор

**б) Рефакторинг-меню (Refactor)**

в) Консоль управления

г) Обозреватель сервера

6) Почему важно проводить рефакторинг до оптимизации производительности?

а) Потому что оптимизация всегда делается первой

**б) Потому что чистый код проще анализировать и оптимизировать**

в) Потому что рефакторинг ускоряет программу

г) Это не важно — можно делать в любом порядке

7) Что может показать профилировщик при запуске приложения?

**а) Сколько времени выполняется каждый метод**

б) Какой цвет лучше использовать в интерфейсе

в) Сколько строк кода написано

г) Какой пользователь самый активный

8) Какой из следующих шагов относится к рефакторингу?

**а) Выделение повторяющегося кода в отдельный метод**

б) Добавление новой кнопки в интерфейс

в) Удаление всех комментариев

г) Замена одной библиотеки на другую без тестирования

9) Что такое «технический долг» в разработке ПО?

а) Деньги, которые должен заплатить разработчик

**б) Последствия быстрых, но плохих решений, которые потом нужно исправлять**

в) Задержка с выходом обновления

г) Ошибка в документации

10) Почему не стоит оптимизировать весь код сразу?

а) Потому что все методы одинаково важны

б) Потому что оптимизация — это вредно

**в) Потому что нужно сначала найти «узкие места» с помощью профилировщика**

г) Потому что код нельзя изменять после релиза

Задание: Рефакторинг кода.

Чтобы увидеть процесс рефакторинга в действии, модифицируем метод Main(), добавив в него следующий код:

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;
```

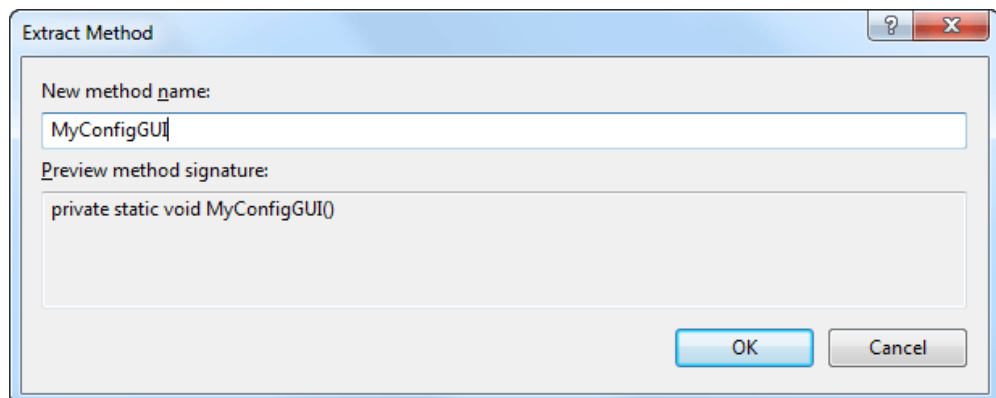
```
namespace ConsoleApplication1
```

```

{
    class Program
    {
        static void Main(string[] args)
        {
            // Настраиваем консольный интерфейс (CUI)
            Console.Title = "Мое приложение";
            Console.ForegroundColor = ConsoleColor.Yellow;
            Console.BackgroundColor = ConsoleColor.Blue;
            Console.WriteLine("Привет, это мой проект!");
            Console.BackgroundColor = ConsoleColor.Black;
            Console.ReadLine();
        }
    }
}

```

В открывшемся далее окне назначьте новому методу имя MyConfigCUI():



После этого метод Main() станет вызывать новый только что сгенерированный метод MyConfigCUI(), внутри которого будет содержаться выделенный ранее код.

Напишите выводы проделанной работы и итоговую программу.

## Тема 1.7 Разработка пользовательского интерфейса.

Перечень вопросов по теме для устного опроса:

- 1) Что такое пользовательский интерфейс?
- 2) Назовите ключевые компоненты интерфейса пользователя.
- 3) Что такое Windows Forms?
- 4) Что такое WPF?
- 5) Назовите правила разработки интерфейсов пользователя.

## Тема 1.8 Основы ADO.Net

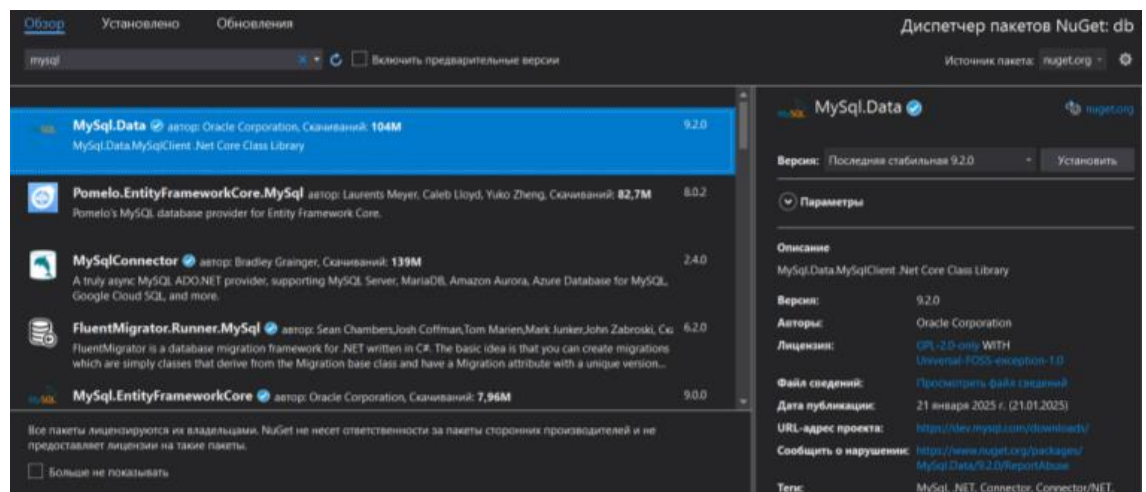
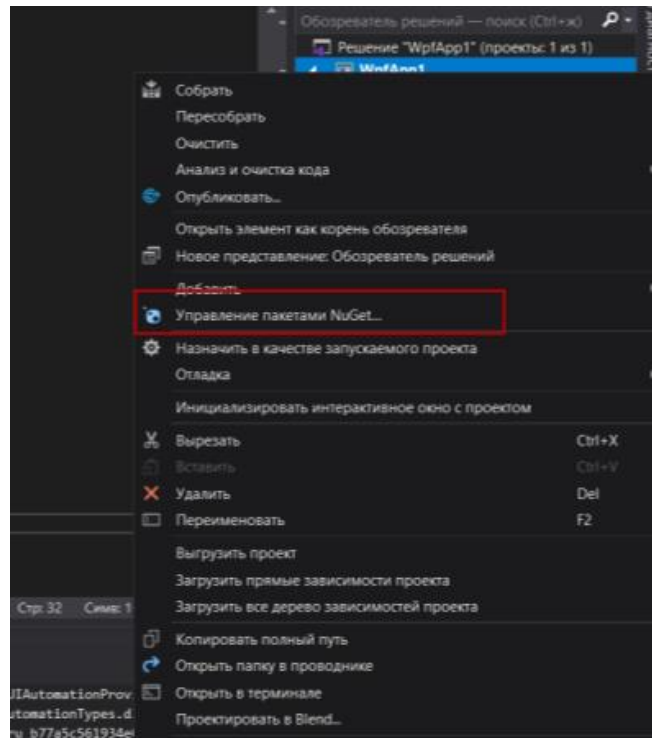
Перечень вопросов по теме для устного опроса:

- 1) По каким видам делятся запросы SQL.
- 2) Назовите команды, работающие со структурой БД.
- 3) Что такое MySQL?

Практическое задание:

Создадим простое приложение с выводом списка пациентов с возможность добавления и удаления.

1. Установим пакет mysqldata в наш проект.



2. Создадим класс модели нашей таблицы (ДЛЯ КАЖДОЙ ТАБЛИЦЫ СОЗДАЕТСЯ ОТДЕЛЬНЫЙ КЛАСС).

```
public int Id { get; set; }
```

```
public string FirstName { get; set; }  
public string LastName { get; set; }  
public string Phone { get; set; }  
public DateTime BirthDay { get; set; }  
public string Address { get; set; }
```

3. Создадим ОТДЕЛЬНЫЙ класс базы данных, в котором будут выполняться все запросы. Пропишем строку подключения.

```
public class DataBase  
{  
    MySqlConnection connection = new MySqlConnection("user=root;  
password=1234; port=3306; server=localhost; database=hospital");  
}
```

4. Пропишите разметку главного окна;
5. Пропишите разметку страницы пациентов;
6. Пропишите метод в классе базы данных для получения информации и пациентах;
7. Пропишите вывод в Список;
8. Добавьте кнопку для внесения в базу нового пациента;
9. Пропишите метод для добавления пациента в базу данных;
10. Создайте окно для добавления нового пациента;
11. Пропишите логику окна;
12. Пропишем логику в окне отображения пациентов;
13. На странице пациентов добавьте кнопку для удаления;
14. Пропишите метод для удаления в классе базы данных;
15. Пропишите логику для кнопки удаления;
16. Сделайте отчет проделанной работы.

## «МДК 01.02 Поддержка и тестирование программных модулей»

### Тема 2.1 Основы тестирования программного обеспечения

Перечень вопросов по теме для устного опроса:

1. Что такое тестирование программного обеспечения?
2. Какие цели преследует тестирование ПО?
3. Чем тестирование отличается от отладки?
4. Что такое дефект, ошибка и сбой?
5. Какие принципы тестирования вы знаете (перечислите не менее 5)?
6. Что такое качество ПО и какие характеристики его определяют?
7. Что означает утверждение «тестирование может показать наличие дефектов, но не их отсутствие»?

### Тема 2.2 Процесс тестирования

Перечень вопросов по теме для устного опроса:

1. Какие основные этапы включает процесс тестирования?
2. Что такое тест-план и какие разделы он содержит?
3. Что такое тестовая стратегия?

4. Для чего требуется анализ требований в тестировании?
5. Чем отличается тест-дизайн от выполнения тестов?
6. Что такое критерии входа и выхода?
7. Какие артефакты создаются на этапе завершения тестирования?

Задание: Разработайте структуру тест-плана для небольшого веб-приложения (логин-форма).

### Тема 2.3 Методы и техники тестирования

Перечень вопросов по теме для устного опроса:

1. Что такое методы «белого», «черного» и «серого» ящика?
2. Какие техники тест-дизайна относятся к тестированию «черного ящика»?
3. В чем заключается разбиение на классы эквивалентности?
4. Что такое анализ граничных значений?
5. Объясните технику «тестирование на основе состояний».
6. Какие методы применяются в тестировании «белого ящика»?
7. Что такое причинно-следственный граф?

Задание: Разработайте набор тестов по технике «анализ граничных значений» для проверки возраста пользователя (допустимый диапазон 18–60).

### Тема 2.4 Функциональное тестирование

Перечень вопросов по теме для устного опроса:

1. Что такое функциональное тестирование?
2. На каких уровнях может проводиться функциональное тестирование?
3. Чем функциональное тестирование отличается от нефункционального?
4. Что такое позитивные и негативные тесты?
5. Что проверяют при валидации данных?
6. Что такое тестовые сценарии и тест-кейсы?
7. Какие риски могут возникнуть при неполном функциональном тестировании?

Задание: Создайте 5 функциональных тест-кейсов для проверки работы формы регистрации.

### Тема 2.5 Нефункциональное тестирование

Перечень вопросов по теме для устного опроса:

1. Что такое нефункциональное тестирование?
2. Что включает в себя тестирование производительности?
3. Что такое нагрузочное, стрессовое и объемное тестирование?
4. Что означает термин «юзабилити»?
5. Чем безопасность отличается от надежности?
6. Как проверяется совместимость?
7. Что включает в себя тестирование установки (installation testing)?

Задание: Опишите 3 сценария проверки производительности для веб-сервиса (например, интернет-магазина).

## Тема 2.6 Интеграционное и системное тестирование

Перечень вопросов по теме для устного опроса:

1. Что такое интеграционное тестирование?
2. Какие существуют стратегии интеграции (top-down, bottom-up)?
3. Чем системное тестирование отличается от интеграционного?
4. Что такое тестовые заглушки (stubs) и драйверы (drivers)?
5. Какие риски могут возникать при интеграции модулей?
6. Что включает в себя системное тестирование?
7. На каком этапе проводится системное тестирование?

Задание: Составьте пример плана интеграционного тестирования для 3 модулей: «Авторизация → Профиль → Заказы».

## Тема 2.7 Приемочное тестирование

Перечень вопросов по теме для устного опроса:

1. Что такое приемочное тестирование?
2. Кто проводит приемочное тестирование?
3. Чем отличается  $\alpha$ -тестирование от  $\beta$ -тестирования?
4. Какие критерии могут быть использованы для приемки ПО?
5. Что такое UAT (User Acceptance Testing)?
6. Зачем нужны чек-листы приемочного тестирования?
7. Какие риски возникают при отсутствии приемочного тестирования?

Задание: Составьте чек-лист приемочного тестирования для простого мобильного приложения (например, «Список задач»).

## Тема 2.8 Управление дефектами

Перечень вопросов по теме для устного опроса:

1. Что такое дефект?
2. Какие статусы дефектов вы знаете?
3. Что такое жизненный цикл дефекта?
4. Какие атрибуты содержит баг-репорт?
5. Что такое приоритет и серьезность дефекта?
6. Какие ошибки чаще всего допускают при оформлении баг-репортов?
7. Что такое triage (расстановка приоритетов дефектов)?

Задание: Создайте 2 баг-репорта: один — критичный, второй — низкого приоритета (можно на вымышленном примере).

## Тема 2.9 Автоматизация тестирования

Перечень вопросов по теме для устного опроса:

1. Что такое автоматизация тестирования?
2. Какие задачи целесообразно автоматизировать?
3. Какие преимущества и недостатки у автоматизации?
4. Что такое тестовый фреймворк?

5. Что такое CI/CD?
6. Какие типы тестов наиболее эффективно автоматизируются?
7. Что такое стабильность автотестов?

Задание: Опишите, какие тесты вы бы автоматизировали в интернет-магазине и почему.

### Тема 2.10 Тестирование API

Перечень вопросов по теме для устного опроса:

1. Что такое API и какие существуют их виды?
2. Чем тестирование API отличается от тестирования UI?
3. Что такое HTTP-методы и какие из них используются при тестировании API?
4. Что такое запрос и ответ в API?
5. Какие форматы данных применяются в API (JSON, XML)?
6. Что проверяют при тестировании API?
7. Какие инструменты используются для API-тестирования?

Задание: Напишите 5 тестов для проверки REST-эндпоинта авторизации (login).

### Тема 2.11 Внедрение и эксплуатация ПО

Перечень вопросов по теме для устного опроса:

1. Что такое внедрение программного обеспечения?
2. Какие этапы включает процесс внедрения?
3. Какие риски могут возникать при запуске системы в эксплуатацию?
4. Что включает обучение пользователей?
5. Чем отличается тестовая среда от продуктивной?
6. Что такое мониторинг работы системы?
7. Почему важны инструкции по эксплуатации?

Задание: Составьте план внедрения небольшого приложения в организацию (5–7 пунктов).

### Тема 2.12 Обслуживание и развитие ПО

Перечень вопросов по теме для устного опроса:

1. Что включает обслуживание программного обеспечения?
2. Чем отличается корректирующее обслуживание от адаптивного?
3. Что такое эволюционное развитие ПО?
4. Зачем требуется регулярное обновление ПО?
5. Какие метрики важны при сопровождении?
6. Что такое SLA и зачем он нужен?
7. Какие виды изменений должны документироваться?

Задание: Опишите план сопровождения ПО после релиза (обновления, мониторинг, поддержка пользователей).

## «МДК 01.03 Разработка мобильных приложений»

### Тема 3.1 Введение в мобильную разработку

Перечень вопросов по теме для устного опроса:

1. Какие особенности отличают мобильную разработку от разработки для ПК?
2. Что такое мобильная платформа?
3. Какие популярные мобильные ОС существуют?
4. Какие языки программирования применяются для мобильной разработки?
5. Какие типы мобильных приложений существуют (native, hybrid, web)?
6. Что такое SDK?
7. Что включает жизненный цикл мобильного приложения?

Задание: Установите Android Studio или Xcode. Создайте пустой проект «HelloMobile» и запустите его на эмуляторе/симуляторе.

### Тема 3.2 Основы разработки под Android/iOS

Перечень вопросов по теме для устного опроса:

1. Какие языки используются для Android и для iOS?
2. Что такое Android Studio и Xcode?
3. Что такое Activity (Android) и ViewController (iOS)?
4. Как работает жизненный цикл Activity?
5. Что такое Info.plist в iOS и Manifest в Android?
6. Что такое эмулятор и симулятор?
7. Что такое Gradle и для чего он используется?

Задание: Создайте экран (Activity или ViewController) с одной кнопкой. При нажатии выводите сообщение в лог.

### Тема 3.3 Пользовательский интерфейс мобильного приложения

Перечень вопросов по теме для устного опроса:

1. Что такое UI и UX?
2. Что такое layout?
3. Какие элементы интерфейса чаще всего используются в мобильных приложениях?
4. Что такое адаптивный интерфейс?
5. Что означает принцип «mobile first»?
6. Какие отличия интерфейсов Android и iOS?
7. Что такое Material Design?

Задание: Создайте экран со списком (ListView/RecyclerView/SwiftUI List) и заполните его 10 элементами.

### Тема 3.4 Хранение данных

Перечень вопросов по теме для устного опроса:

1. Какие способы хранения данных существуют в мобильном приложении?

2. Что такое SharedPreferences/UserDefaults?
3. Что такое SQLite?
4. Что такое Room (Android) или CoreData (iOS)?
5. Чем отличается локальное хранение от удалённого?
6. Что такое сериализация данных?
7. Когда используется кэширование?

Задание: Создайте локальное хранилище (SharedPreferences, UserDefaults или SQLite). Сохраните строку по нажатию кнопки и выведите её при следующем запуске приложения.

### Тема 3.5 Работа с сетевыми данными

Перечень вопросов по теме для устного опроса:

1. Что такое API?
2. Что такое HTTP-запрос?
3. Какие методы HTTP используются чаще всего?
4. Что такое JSON?
5. Что такое асинхронные запросы?
6. Какие библиотеки используются для сетевых операций (Retrofit, Alamofire)?
7. Что такое обработка ошибок при сетевом запросе?

Задание: Сделайте GET-запрос к открытому API. Выведите имя первого пользователя на экран приложения.

### Тема 3.6 Использование ресурсов и локализация

Перечень вопросов по теме для устного опроса:

1. Что относится к ресурсам в мобильном приложении?
2. Что такое string resources?
3. Что такое локализация и зачем она нужна?
4. Как добавить несколько языков в приложение?
5. Какие типы ресурсов существуют (изображения, цвета, строки)?
6. Что такое DPI и зачем нужны разные размеры изображений?
7. Что такое тема (theme) приложения?

Задание: Добавьте в проект строки интерфейса на двух языках (RU/EN). Сделайте так, чтобы текст кнопки менялся в зависимости от языка устройства.

### Тема 3.7 Работа с камерой и галереей

Перечень вопросов по теме для устного опроса:

1. Какие разрешения необходимы для доступа к камере?
2. Как вызвать системную камеру из приложения?
3. Как открыть галерею для выбора изображения?
4. Что такое Intent (Android)?
5. Какие форматы изображений используются?
6. Какие риски связаны с обработкой фото?
7. Что такое runtime permissions?

Задание: Добавьте кнопку «Сделать фото». После снимка отображайте фото в ImageView (или UIImageView).

### Тема 3.8 Геолокация и карты

Перечень вопросов по теме для устного опроса:

1. Какие данные предоставляет GPS?
2. Что такое Location API?
3. Что такое геокодирование?
4. Какие сервисы карт используются в Android и iOS?
5. Что такое маркер на карте?
6. Какие разрешения нужны для работы с геолокацией?
7. Что такое трекинг передвижений?

Задание: Получите текущие координаты устройства и выведите их на экран. Добавьте карту и поставьте маркер в текущей точке.

### Тема 3.9 Уведомления и фоновые процессы

Перечень вопросов по теме для устного опроса:

1. Какие виды уведомлений существуют?
2. Что такое push-уведомления?
3. Что такое службы (Service) в Android?
4. Что такое фоновые задачи?
5. Что такое Firebase Cloud Messaging?
6. Чем отличаются локальные уведомления от удалённых?
7. Какие ограничения существуют для фоновых процессов?

Задание: Создайте локальное уведомление, которое срабатывает через 10 секунд после нажатия кнопки.

### Тема 3.10 Сенсоры устройства

Перечень вопросов по теме для устного опроса:

1. Какие датчики используются в смартфонах?
2. Что такое акселерометр?
3. Что такое гироскоп?
4. Что такое сенсор освещённости?
5. Что такое обработка событий сенсоров?
6. Какие разрешения требуются для сенсоров?
7. Как сенсоры используются в играх?

Задание: Используйте акселерометр и выводите на экран значения движения по осям X/Y/Z.

### Тема 3.11 Тестирование и отладка мобильных приложений

Перечень вопросов по теме для устного опроса:

1. Какие виды тестирования применяются к мобильным приложениям?
2. Что такое эмулятор устройства?
3. Как проводится тестирование на реальном устройстве?
4. Что такое логирование?

5. Что такое профилирование приложения?
6. Какие инструменты отладки есть в Android Studio/Xcode?
7. Что проверяют при тестировании UI?

Задание: Запустите приложение в режиме отладки. Выведите в лог значение переменной и поставьте breakpoint перед её изменением.

### Тема 3.12 Публикация мобильных приложений

Перечень вопросов по теме для устного опроса:

1. Какие магазины приложений существуют?
2. Что такое APK/AAB и IPA?
3. Какие требования предъявляют Google Play и App Store?
4. Что такое сертификаты и подпись приложений?
5. Что включает подготовка приложения к публикации?
6. Что такое описание приложения и метаданные?
7. Что такое модерация приложения?

Задание: Соберите релизную сборку приложения: Android: .apk или .aab. iOS: архив (Archive). Сохраните полученный файл в папку проекта.

### Тема 3.13 Обновление и поддержка

Перечень вопросов по теме для устного опроса:

1. Что такое hotfix?
2. Что такое версии приложения и семантическое версионирование?
3. Чем отличаются минорные и мажорные обновления?
4. Что входит в сопровождение приложения?
5. Что такое crash reports?
6. Почему важна обратная связь от пользователей?

Задание: Измените номер версии приложения (versionCode/versionName или CFBundleVersion/ShortVersion). Соберите новую сборку и проверьте, что номер версии обновился.

### «МДК.01.04 Системное программирование»

#### Тема 4.1 Сборка ПО и переменные окружения

Перечень вопросов по теме для устного опроса:

- 1) Какая команда в Ubuntu открывает текстовый редактор nano для редактирования файла hello.asm?
- 2) Как сохранить и выйти из редактора nano?
- 3) Что такое переменные окружения в Linux?
- 4) Как посмотреть значение переменной окружения HOME в терминале?

Задача: Откройте файл ~/.bashrc с помощью nano, добавьте строку export MY\_PROJECT=/home/user/project, сохраните файл и перезагрузите конфигурацию. Проверьте, доступна ли переменная командой echo \$MY\_PROJECT.

## Тема 4.2 Системы сборки, библиотеки и КИС

Перечень вопросов по теме для устного опроса:

- 1) Что такое статическая библиотека?
- 2) Какое расширение обычно имеет статическая библиотека в Linux?
- 3) Какое расширение обычно имеет динамическая библиотека в Linux?
- 4) Что означает модель Клиент-Интерфейс-Сервер (КИС)?

Задача: Создайте простой Makefile, который компилирует программу из двух файлов: main.c и utils.c, с выводом в app. Запустите сборку командой make.

## Тема 4.3 Введение в Ассемблер

Перечень вопросов по теме для устного опроса:

- 1) Что такое язык ассемблера?
- 2) Что такое регистры процессора?
- 3) Какая команда в ассемблере x86 используется для пересылки данных?
- 4) Что делает инструкция mov eax, 5?
- 5) Какая команда используется для сложения двух чисел в ассемблере?
- 6) Что такое системный вызов в ассемблере?
- 7) Какой регистр в Linux (x86) используется для указания номера системного вызова?
- 8) Для чего в ассемблерной программе нужно объявлять секцию .data?

Задача: Напишите программу на ассемблере (NASM), которая:

Выводит на экран строку: Hello, World

Завершает программу с кодом 0.

## Тема 4.4 Низкоуровневый ввод-вывод и файловые дескрипторы

Перечень вопросов по теме для устного опроса:

- 1) Что такое файловый дескриптор в Linux?
- 2) Какой стандартный файловый дескриптор соответствует выводу на экран (stdout)?
- 3) Какой системный вызов используется для открытия файла в Linux?
- 4) Какой флаг нужно передать в open(), чтобы создать файл?
- 5) Какой системный вызов используется для записи данных в файл?

## Тема 4.5 Файловые атрибуты, ссылки и rps-система

Перечень вопросов по теме для самостоятельной работы:

- 1) Какой символ в начале строки команды ls -l обозначает обычный файл?
  - а) d
  - б) l

- в) -  
г) с
- 2) Какая команда создаёт символическую ссылку link.txt на файл original.txt?
- а) ln original.txt link.txt
  - б) **ln -s original.txt link.txt**
  - в) link original.txt link.txt
  - г) symlink original.txt link.txt
- 3) Какой каталог в Linux содержит информацию о запущенных процессах в виде файлов?
- а) /tmp
  - б) /var
  - в) **/proc**
  - г) /sys
- 4) Что такое индексные дескрипторы?
- 5) Что такое жёсткие ссылки?
- 6) Что такое символические ссылки?

#### Тема 4.5 Каналы, сообщения и разделяемая память

Перечень вопросов по теме для устного опроса:

- 1) Что такое межпроцессное взаимодействие (IPC)?
- 2) Как создать неименованный канал в программе?
- 3) Какая команда в терминале создаёт именованный канал?
- 4) Какие системные вызовы используются для работы с разделяемой памятью в Linux?
- 5) Какой системный вызов используется для отправки сообщения в очередь?

Задача:

Используйте очередь сообщений для передачи структурированного сообщения между двумя процессами.

#### Тема 4.6 Сокеты и синхронизация

Перечень вопросов по теме для устного опроса:

- 1) Для чего используются семафоры в многопроцессных и многопоточных программах?
- 2) Что такое UNIX-сокеты (local socket)?
- 3) Где находится адрес UNIX-сокета?
- 4) Какой тип сокета используется для надёжной потоковой передачи данных в рамках одного хоста?
- 5) Какой протокол обеспечивает надёжную, упорядоченную передачу данных по сети?
- 6) Какой протокол передаёт данные без установления соединения и без гарантии доставки?

## Тема 4.7 Сигналы

Перечень вопросов по теме для устного опроса:

- 1) Что такое сигнал?
- 2) Какой системный вызов используется для установки обработчика сигнала?
- 3) Какой системный вызов отправляет сигнал процессу?
- 4) Какой сигнал посылается процессу при закрытии терминала?

Задача:

Запустите программу в одном терминале, а из другого отправьте ей SIGTERM с помощью команды: `kill -15 <PID>`. Убедитесь, что программа корректно завершается.

## Тема 4.8 Процессы и потоки

Перечень вопросов по теме для устного опроса:

- 1) Что такое процессы? Как их просмотреть?
- 2) Что такое потоки?
- 3) Что произойдёт, если дочерний процесс завершится, а родитель не вызовет `wait()`?
- 4) Какой системный вызов завершает процесс и возвращает статус родителю?

Задача: Напишите программу, которая:

Создаёт дочерний процесс с помощью `fork()`.

В дочернем процессе запускает команду `ls -l` с помощью `execl()`.

Родительский процесс ждёт завершения ребёнка через `wait()` и выводит: Дочерний процесс завершён.

## Тема 4.9 Многопоточное программирование

Перечень вопросов по теме для устного опроса:

- 1) Что значит многопоточное программирование?
  - 2) Что такое поток в многопоточном приложении?
  - 3) Чем потоки одного процесса отличаются от отдельных процессов?
  - 4) Как можно досрочно завершить поток?
- Задача: Напишите программу, в которой создаются два потока. Каждый выводит на экран: Привет от потока [номер].

## Тема 4.10 Демоны и консольный ввод-вывод

Перечень вопросов по теме для устного опроса:

- 1) Что такое демон?
- 2) Чем демон отличается от обычной консольной утилиты?
- 3) Какой системный вызов используется для создания дочернего процесса при запуске демона?

4) Может ли демон использовать сетевые сокеты?

Задача: Измените демон, чтобы он запускал TCP-сервер на порту 8080 и отвечал клиенту Hello from daemon.

#### Тема 4.11 Программирование графического интерфейса с помощью GTK+

Перечень вопросов по теме для устного опроса:

- 1) Что такое GTK+?
- 2) На каком языке преимущественно используется GTK+?
- 3) Какой виджет в GTK+ представляет собой главное окно приложения?
- 4) Какой сигнал в GTK+ генерируется при нажатии на кнопку?
- 5) Какие флаги компиляции нужно добавить при сборке программы с GTK+ в 64-битной системе?

Перечень примерных вопросов для подготовки к экзамену по МДК.01.01 Разработка программных модулей.

- 1) Понятие ЖЦ ПО.
- 2) Этапы ЖЦ ПО.
- 3) Модели жизненного цикла ПО.
- 4) Технология структурного программирования.
- 5) Модульность и декомпозиция программ.
- 6) Основные принципы объектно-ориентированного программирования. Классы: основные понятия.
- 7) Полиморфизм.
- 8) Инкапсуляция и модификаторы доступа.
- 9) Наследование.
- 10) Назначение и виды паттернов.
- 11) Событийно-управляемое программирование
- 12) Отладка кода.
- 13) Оптимизация кода.
- 14) Рефакторинг кода.
- 15) Работа с базами данных

Перечень примерных вопросов для подготовки к экзамену по МДК.01.03 Разработка мобильных приложений.

1. В каких случаях кроссплатформенные приложения становятся оптимальным выбором, и какие типы проектов лучше всего подходят для этого подхода
2. Опишите структуру многомерного массива в Java.
3. Почему нативные приложения лучше подходят для финансовых сервисов, например, банковских приложений?
4. Что такое SQLite
5. Чем веб-приложения удобны для пользователей с точки зрения обновлений

6. Какие методы отражают жизненный цикл Activity
7. Охарактеризовать гибридные приложения
8. Базовые элементы управления. TextView. EditText. Тип ввода текста.
9. Рассказать о видах мобильных приложений.
10. Каким образом происходит создание списков
11. Ресурсы приложения
12. Виды циклов в Java
13. Почему Java редко используется для создания приложений под iOS, несмотря на свою универсальность?
14. Описать работу и структуру цикла for в Java.
15. Какие основные преимущества Java делают её популярной для разработки кроссплатформенных приложений?
16. Класс View
17. Класс Intent
18. Определение жизненного цикла

Перечень примерных вопросов для подготовки к экзамену по МДК.01.04 Системное программирование

- 1) Объясните, как Вы понимаете термин системное программное обеспечение, сформулируете основные понятия.
- 2) Расскажите, как реализуются арифметические операции на Ассемблере.
- 3) Объясните, как Вы понимаете команды обработки строковых данных.
- 4) Перечислите и охарактеризуйте средства аппаратной поддержки функций ОС.
- 5) Объясните двоичное кодирование информации. Представление элементарных типов данных: натуральные числа, целые числа со знаком, числа с плавающей точкой.
- 6) Расскажите про компилятор ассемблер - программ, редактор связей (загрузчик).
- 7) Объясните общую структуру машинных команд.
- 8) Как организуется ввод - вывод и классифицируются внешние устройства.
- 9) Назовите и опишите константы, метки и условную компиляцию.
- 10) Перечислите и опишите этапы проектирования и выполнения программ
- 11) Опишите задачи, для решения которых используется язык ассемблера.
- 12) Объясните регистры eax, ebx, ecx, edx и их специальные свойства.
- 13) Расскажите про команды ADC и SBB.
- 14) Дайте определения резидентным программам.
- 15) Расскажите про команды MUL и IDIV.
- 16) Опишите уровни сложности резидентных программ
- 17) Расскажите про команды IMUL и DIV.
- 18) Опишите операнды Ассемблера

- 19) Опишите назначение простейших команд Ассемблера
- 20) Дайте характеристику языку ассемблера, определение, преимущества.

### 3. Критерии оценки

#### 1. Критерии оценки задач

При оценке задач учитываются все ошибки (грубые и негрубые) и недочёты.

Грубыми считаются ошибки:

- незнание определения основных понятий, правил, незнание формул, общепринятых символов обозначений величин, единиц их измерения;
- неумение применять знания, алгоритмы для решения задач;
- неправильное решение задания (пропуск действия, неправильный выбор действий, лишние действия);
- нерешенная до конца задача или пример;
- невыполненное задание;
- вычислительные ошибки, если они не являются опиской;
- логические ошибки.

К негрубым ошибкам следует отнести:

- неточность формулировок, определений, понятий, теорий, вызванная неполнотой охвата основных признаков определяемого понятия или заменой одного - двух из этих признаков второстепенными;
- неточность графика;
- нерациональный метод решения задачи или недостаточно продуманный план ответа (нарушение логики, подмена отдельных основных вопросов второстепенными);
- неумение решать задачи, выполнять задания в общем виде.

Недочётами являются:

- нерациональные приемы вычислений и преобразований;
- небрежное выполнение записей, чертежей, схем, графиков.

Примечание: за грамматические ошибки, допущенные в решении заданий, оценка не снижается. За неряшливо оформленное задание, несоблюдение правил каллиграфии оценка снижается на 1 балл, но не ниже «3».

#### Критерии оценки выполнения задач

Оценка уровня подготовки		Имеющийся результат
Балл (отметка)	Вербальный аналог	
5	Отлично	<ul style="list-style-type: none"> <li>- задача выполнена полностью.</li> <li>- в логических рассуждениях и обоснованиях нет пробелов и ошибок;</li> <li>- в решении нет математических ошибок (возможна одна неточность, описка, не являющаяся следствием незнания или непонимания учебного материала)</li> </ul>

4	Хорошо	<ul style="list-style-type: none"> <li>- задача выполнено полностью, но обоснования шагов решения недостаточны (если умения обосновывать рассуждения не являлись специальным объектом проверки);</li> <li>- допущена одна ошибка или два-три недочета в выкладках, чертежах или графиках (если эти виды работы не являлись специальным объектом проверки)</li> </ul>
3	Удовлетворительно	<ul style="list-style-type: none"> <li>- допущены более одной ошибки или более двух- трех недочетов в выкладках, чертежах или графика, но обучающийся владеет обязательными умениями по проверяемой теме</li> </ul>
2	Неудовлетворительно	<ul style="list-style-type: none"> <li>- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями по данной теме в полной мере;</li> <li>- выполненное задание показало полное отсутствие у обучающегося обязательных знаний, умений по проверяемой теме или значительная часть заданий выполнена не самостоятельно.</li> </ul>

## 2. Критерии оценки устного опроса:

Оценка «отлично» выставляется студенту, сформулировавшему полный и правильный ответ на вопрос, логично структурировавшему и изложившему материал. При этом студент должен показать знание специальной литературы. Для получения отличной оценки необходимо продемонстрировать умение обозначить проблемные вопросы в соответствующей области, проанализировать их и предложить варианты решений, дать исчерпывающие ответы на уточняющие и дополнительные вопросы.

Оценка «хорошо» выставляется студенту, который дал полный правильный ответ на вопрос, с соблюдением логики изложения материала, но допустил при ответе отдельные неточности, не имеющие принципиального характера. Оценка «хорошо» может выставляться студенту, недостаточно чётко и полно ответившему на уточняющие и дополнительные вопросы.

Оценка «удовлетворительно» выставляется студенту, показавшему неполные знания, допустившему ошибки и неточности при ответе на вопрос, продемонстрировавшему неумение логически выстроить материал ответа и сформулировать свою позицию по проблемным вопросам. При этом хотя бы по одному из заданий ошибки не должны иметь принципиального характера. Студент, ответ которого оценивается «удовлетворительно», должен опираться в своем ответе на учебную литературу.

Оценка «неудовлетворительно» выставляется студенту, если он не дал ответа на вопрос; дал неверные, содержащие фактические ошибки ответы на все вопросы; не смог ответить на дополнительные и уточняющие вопросы. Неудовлетворительная оценка выставляется студенту, отказавшемуся отвечать на вопросы семинара

### 3. Критерии оценки теста:

оценка «5» - правильных ответов 88–100%;

оценка «4» - правильных ответов 68–87%;

оценка «3» - правильных ответов 50–67%;

оценка «2» - правильных ответов < 50%.

### 4. Критерии оценки дифференцированного зачета:

Оценка «5» ставится, если:

- студент свободно применяет знания на практике;
- не допускает ошибок в воспроизведении изученного материала;
- студент выделяет главные положения в изученном материале и не затрудняется в ответах на видоизмененные вопросы;
- студент усваивает весь объем программного материала;
- материал оформлен аккуратно в соответствии с требованиями.

Оценка «4» ставится, если:

- студент знает весь изученный материал;
- отвечает без особых затруднений на вопросы преподавателя;
- студент умеет применять полученные знания на практике;
- в ответах не допускает серьезных ошибок, легко устраняет определенные неточности с помощью дополнительных вопросов преподавателя;
- материал оформлен недостаточно аккуратно и в соответствии с требованиями;

Оценка «3» ставится, если:

- студент обнаруживает освоение основного материала, но испытывает затруднения при его самостоятельном воспроизведении и требует дополнительных дополняющих вопросов преподавателя;
- предпочитает отвечать на вопросы воспроизводящего характера и испытывает затруднения при ответах на воспроизводящие вопросы;
- материал оформлен не аккуратно или не в соответствии с требованиями;

Оценка «2» ставится, если:

- у студента имеются отдельные представления об изучаемом материале, но все, же большая часть не усвоена;
- материал оформлен не в соответствии с требованиями.