

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Щёкина Вера Витальевна

Должность: Ректор

Дата подписания: 07.05.2021 09:44:20

Уникальный программный ключ:

a2232a55157e576517a8999f3190892af53989420420336ffbf573a434e57789



**МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное  
учреждение высшего образования**

**«Благовещенский государственный педагогический университет»**

**ПРОГРАММА ПОДГОТОВКИ СПЕЦИАЛИСТОВ  
СРЕДНЕГО ЗВЕНА**

**Рабочая программа учебной практики**

**УТВЕРЖДАЮ**

**И.о. декана физико-математического  
факультета ФГБОУ ВО «БГПУ»**

 **Т.А. Меределина**

**«29» декабря 2021 г**

**Рабочая программа учебной практики**

**УП.01.01 Учебная практика по модулю**

**Разработка модулей программного обеспечения для компьютерных систем**

**Программа подготовки специалистов среднего звена по специальности  
09.02.07 Информационные системы и программирование**

**Квалификация выпускника  
Программист**

**Принята на заседании кафедры  
информатики и методики преподавания информатики  
(протокол № 5 от «29» декабря 2021 г.)**

**Благовещенск 2021**

## СОДЕРЖАНИЕ

<b>1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА .....</b>	<b>3</b>
<b>2 ТЕМАТИЧЕСКИЙ ПЛАН И СОДЕРЖАНИЕ УЧЕБНОЙ ПРАКТИКИ .....</b>	<b>4</b>
<b>3 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ПРАКТИКИ .....</b>	<b>4</b>
<b>3.1 Требования к минимальному материально-техническому обеспечению .....</b>	<b>4</b>
<b>3.2 Информационное обеспечение обучения .....</b>	<b>5</b>
<b>4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....</b>	<b>6</b>
<b>5 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ПО ДИСЦИПЛИНЕ .....</b>	<b>7</b>
<b>5 ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ .....</b>	<b>45</b>

## 1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

### 1.1 Цель и задачи учебной практики:

- формирование у обучающихся первоначальных умений и практического опыта в рамках профессиональных компетенций, соответствующих основному виду деятельности;
- формирование у обучающихся общих компетенций, соответствующих основному виду деятельности.

Учебная практика проводится при освоении студентами профессиональных компетенций в рамках профессиональных модулей и реализуются концентрированно в несколько периодов. При реализации образовательной программы СПО по специальности 09.02.07 Информационные системы и программирование предусматриваются 4 вида практики в части освоения квалификации «Программист» и основного вида профессиональной деятельности (ВПД):

- Разработка модулей программного обеспечения для компьютерных систем
- Осуществление интеграции программных модулей
- Сопровождение и обслуживание программного обеспечения компьютерных систем
- Разработка, администрирование и защита баз данных

В результате прохождения учебной практики по модулю «Разработка модулей программного обеспечения для компьютерных систем» обучающийся приобретает опыт практической деятельности в области связи, информационных и коммуникационных технологий. Обучающийся в ходе освоения практики должен получить практический опыт:

- разработки кода программного продукта на основе готовой спецификации на уровне модуля;
- использования инструментальных средств на этапе отладки программного продукта;
- проведения тестирования программного модуля по определенному сценарию;
- использования инструментальных средств на этапе отладки программного продукта;
- разработки мобильных приложений.

### 1.2 Место практики в структуре ППССЗ:

УП.01.01 Учебная практика по специальности 09.02.07 Информационные системы и программирование, квалификация «Программист» входит в профессиональный модуль ПМ.01. Разработка модулей программного обеспечения для компьютерных систем.

### 1.3 Практика направлена на формирование следующих компетенций:

Общие компетенции:

Код компетенции	Наименование результата обучения
ОК 09	Использовать информационные технологии в профессиональной деятельности

Профессиональные компетенции:

Основной вид деятельности	Код и наименование компетенции
Разработка модулей программного обеспечения для компьютерных систем	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
	ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ

#### 1.4 Количество часов на освоение программы учебной практики:

Всего 108 часов.

### 2 ТЕМАТИЧЕСКИЙ ПЛАН И СОДЕРЖАНИЕ УЧЕБНОЙ ПРАКТИКИ

Код ПК	Наименование ПК	Виды работ, обеспечивающих формирование ПК	Объем часов
ПК 1.1.	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	Выполнять разработку спецификаций отдельных компонент. Разработка компонент проектной и технической документации с использованием графических языков спецификаций.	18
ПК 1.2.	Разрабатывать программные модули в соответствии с техническим заданием	Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.	18
ПК 1.3.	Выполнять отладку программных модулей с использованием специализированных программных средств	Отладка текстов программ в среде Visual Studio на языке C++, C#.	18
ПК 1.4.	Выполнять тестирование программных модулей.	Осуществлять тестирование отдельных модулей программы.	18
ПК 1.5.	Осуществлять рефакторинг и оптимизацию программного кода	Улучшение и оптимизация программного кода отдельных модулей программы.	18
ПК 1.6.	Разрабатывать модули программного обеспечения для мобильных платформ	Осуществлять разработку модуля программного продукта для мобильных платформ.	18
	<b>ИТОГО</b>		<b>108</b>

### 3 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ПРАКТИКИ

#### 3.1 Требования к минимальному материально-техническому обеспечению

Для реализации программы учебной практики предусмотрены следующие специальные помещения: компьютерный класс.

22 посадочных места. Учебная аудитория для проведения всех видов учебных занятий, групповых и индивидуальных консультаций, текущего контроля, промежуточной аттестации и самостоятельной работы.

Комплект учебной мебели, компьютерные столы, аудиторная доска, компьютеры с установленным лицензионным программным обеспечением, мультимедийный проектор, экспозиционный экран, 11 персональных компьютеров.

Используемое программное обеспечение: Microsoft®WINEDUperDVC AllLng Upgrade/SoftwareAssurancePack Academic OLV 1License LevelE Platform 1Year; Microsoft®OfficeProPlusEducation AllLng License/SoftwareAssurancePack Academic OLV

1License LevelE Platform 1Year; Dr.Web Security Suite; Java Runtime Environment; Calculate Linux.

### 3.2 Информационное обеспечение обучения

#### Литература

1. Гниденко, И. Г. Технология разработки программного обеспечения: учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. – Москва: Издательство Юрайт, 2020. – 235 с. – (Профессиональное образование). – ISBN 978-5-534-05047-9. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: <https://urait.ru/bcode/453640>

2. Казанский, А. А. Программирование на Visual C# : учебное пособие для среднего профессионального образования / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2020. — 192 с. — (Профессиональное образование). — ISBN 978-5-534-14130-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/467844>

3. Кувшинов, Д. Р. Основы программирования : учебное пособие для среднего профессионального образования / Д. Р. Кувшинов. — Москва : Издательство Юрайт, 2020. — 105 с. — (Профессиональное образование). — ISBN 978-5-534-07560-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/454780>

4. Огнева, М. В. Программирование на языке C++: практический курс: учебное пособие для среднего профессионального образования / М. В. Огнева, Е. В. Кудрина. – Москва : Издательство Юрайт, 2020. – 335 с. – (Профессиональное образование). – ISBN 978-5-534-05780-5. – Текст: электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/454231>

5. Подбельский, В. В. Программирование. Базовый курс C# : учебник для среднего профессионального образования / В. В. Подбельский. — Москва : Издательство Юрайт, 2020. — 369 с. — (Профессиональное образование). — ISBN 978-5-534-11467-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/456697>

6. Соколова, В. В. Разработка мобильных приложений : учебное пособие для среднего профессионального образования / В. В. Соколова. — Москва : Издательство Юрайт, 2021. — 175 с. — (Профессиональное образование). — ISBN 978-5-534-10680-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/475892>

7. Трофимов, В. В. Основы алгоритмизации и программирования: учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. – Москва: Издательство Юрайт, 2020. – 137 с. – (Профессиональное образование). – ISBN 978-5-534-07321-8. – Текст: электронный // ЭБС Юрайт [сайт]. – URL: <https://urait.ru/bcode/454452>

8. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для среднего профессионального образования / Д. Ю. Федоров. — 2-е изд. — Москва : Издательство Юрайт, 2020. — 161 с. — (Профессиональное образование). — ISBN 978-5-534-11961-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/454101>

9. Черпаков, И. В. Основы программирования : учебник и практикум для среднего профессионального образования / И. В. Черпаков. — Москва : Издательство Юрайт, 2020. — 219 с. — (Профессиональное образование). — ISBN 978-5-9916-9984-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/452182>

## Базы данных и информационно-справочные системы

1. Сайт Федеральной службы по интеллектуальной собственности, патентам и товарным знакам (Роспатента). – Режим доступа: <https://www.fips.ru>
2. Федеральный портал «Российское образование». – Режим доступа: <http://www.edu.ru>
3. Федеральный центр информационно-образовательных ресурсов. – Режим доступа: <http://srtv.fcior.edu.ru>

## Электронно-библиотечные ресурсы

1. ЭБС «Юрайт». – Режим доступа: <https://urait.ru>
2. Полпред (обзор СМИ). – Режим доступа: <https://polpred.com/news>

## 4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Контроль и оценка результатов освоения учебной практики осуществляется педагогом, реализующим учебную практику в форме дифференцированного зачета (результат выполнения комплексной практической работы).

Результаты обучения (освоенные профессиональные компетенции)	Основные показатели оценки результата	Формы и методы контроля и оценки результатов прохождения учебной практики
ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	<ul style="list-style-type: none"><li>– точное определение функций, структуры и основных спецификаций программных компонент;</li><li>– определение этапов подготовки и разработки программы;</li><li>– точное определение команд языка C++, C#.</li><li>– точное определение видов проектной документации и правил оформления их в соответствии с ГОСТ;</li><li>– построение диаграммы сущность-связь.</li></ul>	Текущий контроль в форме практической работы. Промежуточная аттестация по освоению ПК 1.1.
ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием	<ul style="list-style-type: none"><li>– представление программы в машинах кодах;</li><li>– управление процессом ассемблирования и формирования листинга;</li><li>– получение результатов работы программы в машинных кодах;</li><li>– определение размера памяти и использование регистров.</li></ul>	Текущий контроль в форме практической работы. Промежуточная аттестация по освоению ПК 1.2.
ПК.1.3. Выполнять отладку программных модулей с использованием специализированных программных средств	<ul style="list-style-type: none"><li>– компоновка программы на разных языках;</li><li>– генерация собственных уникальных объектных модулей;</li><li>– компиляция программы и сборка исполняемого модуля.</li></ul>	Текущий контроль в форме практической работы. Промежуточная аттестация по освоению ПК 1.3.

ПК 1.4. Выполнять тестирование программных модулей	– анализ результатов тестирования.	Текущий контроль в форме практической работы. Промежуточная аттестация по освоению ПК 1.4.
ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода	– реализация абстрактных типов данных; – разработка и использование шаблонов; – реализация иерархии объектов на основе механизмов наследования; – организация хранения и обработка массивов данных средствами библиотеки STL; – реализация графического пользовательского интерфейса.	Текущий контроль в форме практической работы. Промежуточная аттестация по освоению ПК 1.5.
ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ	– точное определение функций, структуры и основных спецификаций программного компонента мобильного приложения; – определение этапов подготовки и разработки программы для мобильной платформы.	Текущий контроль в форме практической работы. Промежуточная аттестация по освоению ПК 1.6.

Формы и методы контроля и оценки результатов обучения должны позволять проверять у обучающихся и развитие общих компетенций и обеспечивающих их умений.

<b>Результаты обучения (освоенные общие компетенции)</b>	<b>Формы и методы контроля и оценки результатов прохождения учебной практики</b>
ОК 09. Использовать информационные технологии в профессиональной деятельности	Наблюдение за деятельностью обучающегося в процессе выполнения комплексной практической работы. Промежуточная аттестация по освоению ОК 09.

## **5 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ПО ДИСЦИПЛИНЕ**

### **Вопрос 1**

Какой этап жизненного цикла программного обеспечения начинается с определения потребностей пользователей?

1. Выявление требований
2. Проектирование
3. Разработка
4. Тестирование
5. Развертывание
6. Поддержка

### **Вопрос 2**

Какой этап жизненного цикла программного обеспечения включает в себя создание спецификаций программного обеспечения?

1. Выявление требований
2. Проектирование
3. Разработка
4. Тестирование
5. Развертывание
6. Поддержка

### **Вопрос 3**

Какой этап жизненного цикла программного обеспечения включает в себя написание и компиляцию кода программного обеспечения?

1. Выявление требований
2. Проектирование
3. Разработка
4. Тестирование
5. Развертывание
6. Поддержка

### **Вопрос 4**

Какой этап жизненного цикла программного обеспечения включает в себя проверку программного обеспечения на наличие ошибок и несоответствий?

1. Выявление требований
2. Проектирование
3. Разработка
4. Тестирование
5. Развертывание
6. Поддержка

### **Вопрос 5**

Какой этап жизненного цикла программного обеспечения включает в себя установку программного обеспечения на целевые системы?

1. Выявление требований
2. Проектирование
3. Разработка
4. Тестирование
5. Развертывание
6. Поддержка

### **Вопрос 6**

Какой этап жизненного цикла программного обеспечения включает в себя решение проблем и предоставление обновлений программного обеспечения?

1. Выявление требований
2. Проектирование
3. Разработка
4. Тестирование
5. Развертывание
6. Поддержка

### **Вопрос 7**

Какая модель жизненного цикла программного обеспечения является наиболее распространенной?

1. Водопад
2. Итеративная
3. Адаптивная

### **Вопрос 8**

Какая модель жизненного цикла программного обеспечения позволяет вносить изменения в программное обеспечение на более поздних этапах жизненного цикла?

1. Водопад
2. Итеративная
3. Адаптивная

### **Вопрос 9**

Какая модель жизненного цикла программного обеспечения позволяет разрабатывать программное обеспечение в параллельном режиме?

1. Водопад
2. Итеративная
3. Адаптивная

### **Вопрос 10**

Какой из следующих этапов жизненного цикла программного обеспечения является наименее формальным?

1. Выявление требований
2. Проектирование
3. Разработка
4. Тестирование
5. Развертывание
6. Поддержка

### **Вопрос 11**

Какой из следующих этапов жизненного цикла программного обеспечения является наиболее дорогостоящим?

1. Выявление требований
2. Проектирование
3. Разработка
4. Тестирование
5. Развертывание
6. Поддержка

### **Вопрос 12**

Какой из следующих этапов жизненного цикла программного обеспечения является наиболее важным для обеспечения качества программного обеспечения?

1. Выявление требований
2. Проектирование
3. Разработка
4. Тестирование
5. Развертывание
6. Поддержка

### **Задание 13**

Какие из следующих операторов являются структурными операторами?

1. if-else
2. while
3. for
4. boolean
5. goto

#### Задание 14

Какие из следующих утверждений являются верными в отношении структурного программирования?

1. Структурное программирование основано на иерархической структуре блоков.
2. Структурное программирование позволяет легко выявлять и исправлять ошибки.
3. Структурное программирование является наиболее эффективной парадигмой программирования.
4. Структурное программирование позволяет легко модифицировать программное обеспечение.

#### Задание 15

Какие из следующих утверждений являются верными в отношении структурного программирования?

1. Структурное программирование основано на использовании только трех основных структур управления: последовательного, ветвления и цикла.
2. Структурное программирование позволяет создавать программы, которые являются более понятными, читаемыми и поддерживаемыми.
3. Структурное программирование позволяет избежать использования рекурсии в программах.
4. Структурное программирование позволяет создавать программы, которые являются более эффективными с точки зрения использования памяти и процессорного времени.

#### Задание 16

Дан код программы на Python, считающей количество неотрицательных чисел в списке, дополните пропущенные места, чтобы он работал верно

```
def count_positives(arr):  
    count = 0  
      
    if num >= 0:  
        count += 1  
    return count
```

```
arr = [1, 2, 3, -4, -5]  
print(count_positives(arr))
```

#### Задание 17

Сопоставьте определения понятиям

Понятие	Определение
1. Класс	1. Атрибут класса или объекта, который хранит значение
2. Объект	2. Действие, которое может выполнять объект
3. Поле	3. Абстрактная модель объекта, которая определяет его свойства и поведение
4. Метод	4. Отношение между классами, при котором один класс (наследник) получает свойства и поведение от другого класса (родителя)
5. Наследование	5. Экземпляр класса, который имеет конкретные значения свойств и может выполнять действия, определенные методами

	класса
6. Абстракция	6. Скрытие деталей реализации класса от пользователя
7. Полиморфизм	7. Возможность вызывать один и тот же метод для объектов разных классов, при этом результат вызова может быть разным

### Задание 18

Сопоставьте ключевые слова Java их назначению:

1. class	1. Наследование
2. new	2. Наследование интерфейса
3. this	3. Определение класса
4. super	4. Ссылка на текущий объект
5. extends	5. Ссылка на родительский класс
6. implements	6. Определение интерфейса
7. interface	7. Создание объекта

### Задание 19

Сопоставьте типы данных Java с их описанием:

1 int	1 Числовой тип данных с плавающей запятой с повышенной точностью
2 float	2 Базовый класс для всех классов в Java
3 double	3 Числовой тип данных с плавающей запятой
4 char	4 Строковый тип данных
5 boolean	5 Целочисленный тип данных
6 String	6 Символьный тип данных
7 Object	7 Логический тип данных

### Задание 20

Сопоставьте модификатор доступа и его описание

1 public	1 Доступ к методу разрешен только из класса, в котором определен метод.
2 protected	2 Доступ к методу разрешен из классов, которые находятся в том же пакете, что и класс, в котором определен метод.
3 friendly	3 Доступ к методу разрешен из любого места программы.
4 private	4 Доступ к методу разрешен из классов, которые наследуют от класса, в котором определен метод.

### Задание 21

Какой паттерн проектирования используется для обеспечения возможности расширения функциональности класса без изменения его исходного кода?

1. Singleton
2. Factory Method
3. Decorator
4. Adapter

### Задание 22

Какой паттерн проектирования используется для создания иерархии классов, основанной на отношениях «один ко многим»?

1. Singleton

2. Factory Method
3. Strategy
4. Composite

### **Задание 23**

Какой паттерн проектирования используется для обеспечения возможности взаимозаменяемости различных алгоритмов?

1. Singleton
2. Factory Method
3. Strategy
4. Adapter

### **Задание 24**

Какой паттерн проектирования используется для создания объекта, который может использоваться в качестве замены для другого объекта?

1. Singleton
2. Factory Method
3. Facade
4. Adapter

### **Задание 25**

Какой паттерн проектирования используется для создания объекта, который предоставляет простой интерфейс к сложной системе?

1. Facade
2. Singleton
3. Factory Method
4. Observer

### **Задание 26**

Какой паттерн проектирования используется для обеспечения возможности создания объектов в контексте, который управляет их жизненным циклом?

1. Singleton
2. Builder
3. Factory Method
4. Prototype

### **Задание 27**

Какой паттерн проектирования используется для обеспечения возможности совместного использования объектов, не раскрывая их реализации?

1. Singleton
2. Factory Method
3. Facade
4. Flyweight

### **Задание 28**

Какой паттерн проектирования используется для обеспечения возможности уведомления подписчиков об изменениях состояния объекта?

1. Singleton
2. Factory Method
3. Command
4. Observer

### Задание 29

Какой паттерн проектирования используется для обеспечения возможности выполнения задачи в фоновом режиме?

1. Command
2. Singleton
3. Factory Method
4. State

### Задание 29

Какой паттерн проектирования используется для обеспечения возможности управления поведением объекта в зависимости от его состояния?

1. Singleton
2. Factory Method
3. State
4. Iterator

### Задание 31

Дан код на Python. Перепишите его так, чтобы не использовалась рекурсия:

```
def factorial(n):
```

```
    if n == 0:
```

```
        return 1
```

```
    else:
```

```
        return n * factorial(n - 1)
```

### Задание 32

Дан код на Python, который осуществляет переворот строки. Перепишите код таким образом, чтобы не использовались специфические языковые особенности, то есть при помощи посимвольной обработки строк:

```
def reverse(s):
```

```
    return s[::-1]
```

### Задание 33

Перепишите код так, чтобы использовался цикл while:

```
for i in range(1, 10):
```

```
    if i % 2 == 0:
```

```
        print(i)
```

### Задание 34

Дан код на Python, рекурсивно вычисляющий факториал. Оцените сложность алгоритма по времени и памяти

```
def factorial(n):
```

```
    if n == 0:
```

```
        return 1
```

```
    else:
```

```
        return n * factorial(n - 1)
```

### Задание 35

Какие оптимизации можно применить к этому коду для повышения его производительности?

```
public class MyClass {  
    public void foo() {  
        for (int i = 0; i < 1000; i++) {
```

```

        int j = 0;
        while (j < 1000) {
            j++;
        }
    }
}

```

### Задание 36

Какой рефакторинг можно применить к предложенному коду для улучшения читабельности и ремонтпригодности?

```

public class MyClass {
    public void foo() {
        int[] numbers = new int[1000];
        for (int i = 0; i < numbers.length; i++) {
            numbers[i] = i;
        }
        int sum = 0;
        for (int i = 0; i < numbers.length; i++) {
            sum += numbers[i];
        }
        System.out.println(sum);
    }
}

```

### Задание 37

Каким образом дополнить класс, чтобы он реализовывал окно, устанавливал его заголовки и размер?

```

public class MyFrame extends JFrame {
    public MyFrame() {
        // ...
    }
}

```

### Задание 38

Каким образом можно обработать событие нажатия на кнопку javax.swing.JButton?

### Задание 39

Создайте приложение на Java, которое будет выполнять следующие действия:

1. Подключаться к базе данных MySQL.
2. Создавать новую таблицу с именем users.
3. Вставлять записи в таблицу users.
4. Обновлять записи в таблице users.
5. Удалять записи из таблицы users.

### Задание 40

Напишите запрос, создающий таблицу в базе данных со следующими свойствами:

- users - это имя таблицы, которую мы хотим создать.
- id - это имя первого столбца. Тип данных этого столбца - INT, и он является первичным ключом.
- name - это имя второго столбца. Тип данных этого столбца - VARCHAR(255), и он не имеет ограничений.

**Система оценивания заданий для проведения диагностической работы  
по дисциплине МДК.01.01 Разработка программных модулей**

За правильный ответ на задания 1- 30 ставится 1 балл, за неверный ответ или его отсутствие – 0 баллов.

Номер задания	Правильный ответ
1	1
2	2
3	3
4	4
5	5
6	6
7	1
8	2
9	3
10	1
11	3
12	4
13	1, 2, 3
14	1, 2, 4
15	1, 2
16	for num in arr:
17	1-3, 2-5, 3-1, 4-2, 5-4, 6-6, 7-7
18	1-3, 2-7, 3-4, 4-5, 5-1, 6-2, 7-6
19	1-5, 2-3, 3-1, 4-6, 5-7, 6-4, 7-2
20	1-3, 2-4, 3-2, 4-1
21	3
22	4
23	3
24	4
25	1
26	2
27	4
28	4
29	1
30	3

### Критерии оценивания выполнения заданий с развернутым ответом

#### Задание 31

Правильный ответ:

```
def factorial(n):
    result = 1
    for i in range(1, n + 1):
        result *= i
    return result
```

Ключевой момент - это использование цикла вместо рекурсивного вызова. В этом случае начисляется 1 балл. Конкретный способ организации цикла, имена переменных не важны.

#### Задание 32

Правильный ответ:

```
def reverse(s):
```

```
result = ""
for i in range(len(s) - 1, -1, -1):
    result += s[i]
return result
```

Конкретная реализация цикла может быть несколько иной, важна демонстрация способности работы с символами, строками и циклами. За верное решение начисляется 1 балл.

### Задание 33

Правильный ответ:

```
i = 1
while i <= 10:
    if i % 2 == 0:
        print(i)
    i += 1
```

Обучающийся должен продемонстрировать владение синтаксисом в отношении описания цикла `while`, должен не забыть объявить переменную-счетчик до описания цикла. Конкретные имена переменных не важны. В этом случае начисляется 1 балл.

### Задание 34

Решение:

Время:

Алгоритм рекурсивно вычисляет факториал числа  $n$ . На каждой итерации алгоритма выполняется умножение, которое занимает  $O(1)$  времени. Кроме того, на каждой итерации происходит вызов алгоритма с меньшим аргументом. Количество таких вызовов равно  $n$ . Таким образом, общее время работы алгоритма можно оценить как  $O(n)$ .

Память:

Алгоритм использует рекурсию, поэтому в памяти хранятся данные обо всех вызовах алгоритма. Количество таких вызовов равно  $n$ . Таким образом, сложность алгоритма по памяти можно оценить как  $O(n)$ .

Ответ:

Сложность алгоритма по времени и памяти  $O(n)$ .

В случае верной оценки по времени начисляется 1 балл, в случае верной оценки по памяти начисляется 1 балл.

### Задание 35

- Использовать цикл `for` вместо цикла `while`. В данном случае цикл `for` будет более производительным, поскольку он не требует проверки условия на каждом шаге.

- Использовать переменную  $i$  в качестве счетчика цикла `for`. В данном случае это позволит избежать необходимости инициализации переменной  $j$  на каждом шаге цикла.

- Использовать умножение вместо сложения для вычисления значения счетчика цикла. В данном случае это позволит сократить количество операций, выполняемых на каждом шаге цикла.

Начисляем 1 балл за каждый правильный ответ в рамках задания. Максимальный балл – 3, минимальный – 0.

### Задание 36

- Вынести инициализацию массива `numbers` в отдельный метод. Это позволит сделать код более читабельным и улучшить его понимание.

- Вынести вычисление суммы в отдельный метод. Это позволит сделать код более модульным и улучшить его ремонтпригодность.

Начисляем 1 балл за каждый правильный ответ в рамках задания. Максимальный балл – 2, минимальный – 0.

### Задание 37

- Добавить в класс конструктор, который будет принимать в качестве параметра размер окна. Это позволит задать размер окна при создании экземпляра класса.
- Добавить в класс метод setTitle(), который будет устанавливать заголовок окна. Это позволит задать заголовок окна в любое время.
- Добавить в класс метод setVisible(), который будет отображать или скрывать окно. Это позволит управлять видимостью окна.

Допускается ответ в виде переписанного кода класса:

```
public class MyFrame extends JFrame {
    public MyFrame(int width, int height) {
        super(width, height);
    }
    public void setTitle(String title) {
        super.setTitle(title);
    }
    public void setVisible(boolean visible) {
        super.setVisible(visible);
    }
}
```

Следует отслеживать каждую деталь, свидетельствующую об осознанном проектировании оконного интерфейса. Поскольку цели проектирования могут быть разные, то и действия с размерами, положением и функциями окна будут разные. Важно отметить, что обучающий действует с пониманием внутреннего устройства класса JFrame и наиболее часто используемых методов. В этом случае начисляется 1 балл.

### Задание 38

- Описать класс слушатель события, унаследованный от ActionListener, переопределить в нем метод actionPerformed и прописать в нем логику работы кнопки.
- Прикрепить к кнопке этот класс с помощью метода addActionListener. Допускается прикрепление анонимного класса слушателя на этом этапе.

Начисляем по 1 баллу за каждый верный ответ. Максимальная оценка: 2 балла, минимальная – 0 баллов.

### Задание 39

```
import java.sql.*;

public class DatabaseApp {

    private static final String DB_URL = "jdbc:mysql://localhost:3306/mydb";
    private static final String DB_USERNAME = "root";
    private static final String DB_PASSWORD = "password";

    public static void main(String[] args) throws SQLException {
        // Подключаемся к базе данных
        Connection connection = DriverManager.getConnection(DB_URL,
DB_USERNAME, DB_PASSWORD);
        // Создаем новую таблицу
        Statement statement = connection.createStatement();
```

```

statement.execute("CREATE TABLE users (id INT NOT NULL AUTO_INCREMENT, name VARCHAR(255), age INT, PRIMARY KEY (id));

// Вставляем записи
statement.execute("INSERT INTO users (name, age) VALUES ('John Doe', 30)");
statement.execute("INSERT INTO users (name, age) VALUES ('Jane Doe', 25)");

// Обновляем записи
statement.execute("UPDATE users SET age = 35 WHERE name = 'John Doe'");

// Удаляем записи
statement.execute("DELETE FROM users WHERE name = 'Jane Doe'");

// Закрываем соединение
connection.close();
}
}

```

Отслеживаем последовательное выполнение всех 5 пунктов задания и выставаем за каждое выполненное задание 1 балл. Максимальная оценка – 5, минимальная – 0.

**Задание 40**

```

CREATE TABLE users (
    id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(255),
);

```

Отслеживаем корректность типов данных. В случае верного описания выставаем 1 балл.

Формируемая компетенция	Показатели освоения компетенций
<p><b>ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.</b></p>	<p><b>Знать:</b></p> <ul style="list-style-type: none"> <li>• основные принципы отладки и тестирования программных продуктов.</li> </ul> <p><b>Уметь:</b></p> <ul style="list-style-type: none"> <li>• выполнять отладку и тестирование программы на уровне модуля.</li> </ul> <p><b>Владеть:</b></p> <ul style="list-style-type: none"> <li>• использованием инструментальных средств на этапе отладки программного продукта.</li> </ul>

**Задание 1.**

Какие из перечисленных этапов не относятся к процессу отладки программного обеспечения?

- а) определение ошибки
- б) изоляция ошибки
- в) отсеивание несущественных ошибок
- г) тестирование нового функционала
- д) устранение ошибки
- е) проверка работоспособности исправленной программы
- ж) разработка дополнительных функций
- з) версионирование и автоматизация сборки проекта

Ответ: г, ж, з

### Задание 2.

Что не является методом отладки программного обеспечения?

- а) индукция
- б) обратное прослеживание
- в) трансляция
- г) ручное тестирование
- д) дедукция
- е) умозаключение

Ответ: в, е

### Задание 3.

Установите соответствие между названием и функцией инструмента среды программирования:

- |                 |   |
|-----------------|---|
| 1. редактор     | а) создание и изменение исходных символьных файлов, содержащих программу                                  |
| 2. транслятор   | б) формирование исполняемого файла  |
| 3. библиотекарь | в) перевод программы с языка программирования на язык машинных кодов                                      |
| 4. компоновщик  | г) поддержка совокупности объектных файлов с подпрограммами, типами данных                                |
| 5. загрузчик    | д) копирование исполняемого файла с диска в память и инициализация компьютера перед выполнением программы |

Ответ: 1а, 2в, 3г, 4б, 5д

### Задание 4.

Установите соответствие между термином и его определением:

- |                 |   |
|-----------------|---|
| 1. отладка      | а) процесс поиска ошибок                    |
| 2. тестирование | б) процесс локализации и исправления ошибок |

Ответ: 1б, 2а

### Задание 5.

Установите соответствие:

- |                      |  |
|----------------------|--|
| 1. ошибки компиляции | а) ошибки, обнаруженные компоновщиком при объединении модулей программы  |
| 2. ошибки компоновки | б) ошибки, фиксируемые компилятором при выполнении синтаксического и частично семантического анализа программы   |
| 3. ошибки выполнения | в) ошибки, обнаруженные операционной системой, аппаратными средствами или пользователем при выполнении программы |

Ответ: 1б, 2а, 3в

### Задание 6.

Установите соответствие:

- |                          |   |
|--------------------------|---|
| 1. синтаксическая ошибка | а) происходит во время исполнения программы                         |
| 2. ошибка исполнения     | б) происходит, когда программа ведет себя не так, как было задумано |

3. логическая ошибка

в) происходит, когда встречается код, который не соответствует правилам языка программирования

Ответ: 1в, 2а, 3б

#### Задание 7.

Как называется процесс поиска и исправления ошибок или неполадок в исходном коде какого-либо программного обеспечения?

Ответ: отладка

#### Задание 8.

Как называется процесс выполнения программы с целью обнаружения ошибки?

Ответ: тестирование

#### Задание 9.

Приведите обобщенный алгоритм отладки программного обеспечения (ПО).

Ответ:

<b>Содержание верного ответа</b> (допускаются иные формулировки ответа, не искажающие его смысла)	
1. Выявление проблемы: определение симптомов проблемы, сбор информации о ПО и системе. 2. Выбор инструмента отладки: выбор подходящего отладчика или другого инструмента для работы с ПО. 3. Установка и настройка инструмента отладки на компьютере. 4. Запуск отладки: запуск инструмента отладки, подключение его к ПО и остановка выполнения ПО в точке возникновения проблемы. 5. Локализация ошибки: исследование состояния ПО в момент остановки, определение возможных причин проблемы на основе данных отладчика и анализа кода ПО. 6. Устранение ошибки: исправление ошибочного кода, тестирование исправленного ПО на различных тестовых данных. 7. Завершение отладки: проверка работоспособности ПО после устранения проблемы, подтверждение отсутствия других ошибок, запись результатов отладки в журнал.	
<b>Указания по оцениванию</b>	<b>Баллы</b>
В соответствии с требованиями задания правильно приведены верные шаги алгоритма при отсутствии дополнительных шагов, содержащих неточности/ошибки	2
В соответствии с требованиями задания правильно приведено более половины верных шагов при отсутствии либо наличии дополнительных шагов, содержащих неточности/ошибки.	1
Все иные ситуации, включая отсутствие данного в явном виде ответа на вопрос	0
<i>Максимальный балл</i>	2

#### Задание 10.

Перечислите не менее пяти компонентов, входящих в состав интегрированной среды разработки.

Ответ:

<b>Содержание верного ответа</b> (допускаются иные формулировки ответа, не искажающие его смысла)
1. Редактор исходного кода. 2. Компилятор / интерпретатор.

3. Отладчик. 4. Менеджер проектов. 5. Система управления версиями. 6. Средства автоматизации сборки и тестирования. 7. Интеграция с внешними инструментами.	
<b>Указания по оцениванию</b>	<b>Баллы</b>
В соответствии с требованиями конкретного задания правильно приведены пять или более верных пунктов при отсутствии дополнительных пунктов, содержащих неточности/ошибки	3
В соответствии с требованиями конкретного задания правильно приведены три-четыре пункта в при отсутствии дополнительных пунктов, содержащих неточности/ошибки	2
В соответствии с требованиями конкретного задания правильно приведены только два или три пункта при отсутствии дополнительных пунктов, содержащих неточности/ошибки. ИЛИ В соответствии с требованиями конкретного задания правильно приведены три-четыре пунктов при наличии одного или более дополнительных пунктов, содержащих неточности/ошибки	1
Все иные ситуации, включая отсутствие данного в явном виде ответа на вопрос	0
<i>Максимальный балл</i>	<i>3</i>

<b>Формируемая компетенция</b>	<b>Показатели освоения компетенций</b>
<b>ПК 1.4. Выполнять тестирование программных модулей.</b>	<p><b>Знать:</b></p> <ul style="list-style-type: none"> <li>• основные принципы отладки и тестирования программных продуктов.</li> </ul> <p><b>Уметь:</b></p> <ul style="list-style-type: none"> <li>• выполнять отладку и тестирование программы на уровне модуля;</li> <li>• оформлять документацию на программные средства.</li> </ul> <p><b>Владеть:</b></p> <ul style="list-style-type: none"> <li>• проведением тестирования программного модуля по определенному сценарию.</li> </ul>

### **Задание 1.**

Какие существуют разновидности структурных критериев?

- критерий тестирования ветвей
- критерий тестирования команд
- критерий тестирования путей
- критерий тестирования циклов

Ответ: а, б, в

### **Задание 2.**

Какие предъявляются требования к идеальному критерию тестирования?

- проверяемость
- достижимость

- в) полнота
  - г) достаточность
- Ответ: а, в, г

### Задание 3.

Какие два из перечисленных методов тестирования дают наиболее надежные результаты?

- а) статические методы
- б) интеграционное тестирование
- в) модульное тестирование
- г) системное тестирование в реальном окружении и реальном времени
- д) системное тестирование с моделируемым окружением

Ответ: г, д

### Задание 4.

Какие из указанных категорий относятся к типам тестовых примеров:

- а) неверные данные,
- б) надежность системы,
- в) реинициализация системы,
- г) граничные данные,
- д) допустимые данные,
- е) большие данные,
- ж) нештатные состояния среды выполнения

Ответ: а, в, г, д, ж

### Задание 5.

Установите соответствие:

- |                        |  |
|------------------------|--|
| 1 Постепенный отказ    | а) многократно возникающий, самоустраняющийся сбой одного и того же характера  |
| 2 Конструктивный отказ | б) отказ, возникновение которого вызвано другими отказами  |
| 3 Перемежающийся отказ | в) отказ, вызванный нарушением процесса проектирования и разработки системы  |
| 4 Явный отказ          | г) отказ, вызванный постепенным изменением одного из параметров системы или обрабатываемых системой данных           |
| 5 Зависимый отказ      | д) отказ, который обнаруживается сразу после его возникновения штатными средствами контроля состояния системы        |
| 6 Внезапный отказ      | е) отказ, вызванный резким скачкообразным изменением одного из параметров системы или обрабатываемых системой данных |

Ответ: 1г, 2в, 3а, 4д, 5б, 6е

### Задание 6.

Как реализуются динамические методы построения тестовых путей?

- а) поиск всех реализуемых путей
- б) наращивание начальных отрезков реализованных путей продолжающимися их фрагментами, чтобы увеличить покрытие
- в) построение пути методом удлинения за счет добавления дуг

Ответ: а, б

### Задание 7.

Какие существуют разновидности интеграционного тестирования?

- а) регрессионное тестирование

- б) восходящее тестирование
- в) нисходящее тестирование
- г) моноконтное тестирование

Ответ: б, в, г

#### **Задание 8.**

Установите соответствие между состоянием дефекта и его расшифровкой:

- |              |  |
|--------------|--|
| 1. verified  | а) дефект занесен в базу дефектов                                  |
| 2. resolved  | б) решение о замораживании активности по исправлению дефекта       |
| 3. new       | в) успешное исправление дефекта подтверждено инженером по качеству |
| 4. open      | г) дефект разработчиком исправлен                                  |
| 5. postponed | д) дефект зафиксирован за разработчиком для исправления            |

Ответ: 1в, 2г, 3а, 4д, 5б

#### **Задание 9.**

Какие классы частных критериев тестируемости известны?

- а) стохастические критерии
- б) структурные критерии
- в) функциональные критерии
- г) сценарные критерии
- д) мутационный критерий

Ответ: а, б, в, д

#### **Задание 10.**

При создании очередной версии программы была добавлена функция А, функция D была удалена, функция С – изменена, а функция U – оставлена без изменений. К какой группе относится тест, покрывающий только функцию D?

Ответ: устаревший тест

#### **Задание 11.**

На предыдущей версии программы тест 1 завершился в состоянии А, тест 2 – в состоянии В, а тест 3 – в состоянии С. На текущей версии программы тест 1 завершился в состоянии А, тест 2 – в состоянии С, а тест 3 – в состоянии D. На базе какого состояния наиболее целесообразна разработка новых тестов?

Ответ: состояние D

#### **Задание 12.**

На предыдущей версии программы тест 1 завершился в состоянии А, тест 2 – в состоянии В, а тест 3 – в состоянии С. На текущей версии программы тест 1 завершился в состоянии А, тест 2 – в состоянии С, а тест 3 – в состоянии D. На базе какого теста наиболее целесообразна разработка новых тестов?

Ответ: тест 3

#### **Задание 13.**

Как называется этап регрессионного тестирования, на котором проводится упорядочение тестов?

Ответ: отбор тестов

#### **Задание 14.**

Как называется этап регрессионного тестирования, на котором удаляются устаревшие тесты?

Ответ: обновление базы данных

**Задание 15.**

Как называется метод тестирования, при котором тестируемый вводит данные и анализирует результат, но он не знает, как именно работает программа?

Ответ: метод «черного ящика»

**Задание 16.**

Как называется метод тестирования, при котором тестируемый разрабатывает тесты, основываясь на знании исходного кода, к которому он имеет полный доступ?

Ответ: метод «белого ящика»

**Задание 17.**

Как называется тестирование программного обеспечения, которое оценивает надежность и устойчивость системы в условиях превышения пределов нормального функционирования?

Ответ: стресс-тестирование

**Задание 18.**

Проверка работоспособности программного обеспечения при длительном тестировании с ожидаемым уровнем нагрузки?

Ответ: тестирование стабильности

**Задание 19.**

Напишите тестовый сценарий для проверки требования «Система должна позволять пользователям регистрироваться».

Ответ:

<b>Содержание верного ответа</b> (допускаются иные формулировки ответа, не искажающие его смысла)	
Сценарий должен включать следующие шаги: <ul style="list-style-type: none"><li>– Пользователь открывает регистрационную форму на сайте.</li><li>– Пользователь вводит свои данные (имя, фамилия, адрес электронной почты, пароль).</li><li>– Пользователь нажимает кнопку «Регистрация».</li><li>– Система подтверждает успешную регистрацию и перенаправляет пользователя на главную страницу.</li></ul>	
<b>Указания по оцениванию</b>	<b>Баллы</b>
Дан правильный ответ на вопрос: указано необходимое количество требуемых объектов при отсутствии неверных позиций	1
Все иные ситуации, включая отсутствие данного в явном виде ответа на вопрос	0
<i>Максимальный балл</i>	<i>1</i>

**Задание 20.**

Напишите тестовый сценарий для проверки требования «Система должна позволять пользователям авторизоваться».

Ответ:

<b>Содержание верного ответа</b> (допускаются иные формулировки ответа, не искажающие его смысла)	
Сценарий должен включать следующие шаги:	

<ul style="list-style-type: none"> <li>– Пользователь открывает форму авторизации на сайте.</li> <li>– Пользователь вводит свой логин и пароль.</li> <li>– Пользователь нажимает кнопку «Войти».</li> <li>– Система подтверждает успешную авторизацию и перенаправляет пользователя на его личную страницу.</li> </ul>	
<b>Указания по оцениванию</b>	<b>Баллы</b>
Дан правильный ответ на вопрос: указано необходимое количество требуемых объектов при отсутствии неверных позиций	1
Все иные ситуации, включая отсутствие данного в явном виде ответа на вопрос	0
<i>Максимальный балл</i>	<i>1</i>

<b>Формируемая компетенция</b>	<b>Показатели освоения компетенций</b>
<b>ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода.</b>	<p><b>Знать:</b></p> <ul style="list-style-type: none"> <li>• способы оптимизации и приемы рефакторинга.</li> </ul> <p><b>Уметь:</b></p> <ul style="list-style-type: none"> <li>• выполнять оптимизацию и рефакторинг программного кода.</li> </ul> <p><b>Владеть:</b></p> <ul style="list-style-type: none"> <li>• использованием инструментальных средств на этапе отладки программного продукта.</li> </ul>

### **Задание 1.**

Укажите основные механизмы защитного программирования:

- а) ввод недопустимых значений,
- б) проверка допущений в критических точках,
- в) тестирование классов,
- г) обработка исключительных ситуаций,
- д) валидация.

Ответ: б, г

### **Задание 2.**

Какие методы используются для обеспечения соответствия разработанных программных модулей стандартам кодирования?

- а) проверка кода
- б) инжиниринг
- в) рециркуляция
- г) рефакторинг
- д) тестирование

Ответ: а, г, д

### **Задание 3.**

Выберите из предложенного списка критерии оптимизации программ.

- а) скорость работы
- б) объем используемой памяти
- в) размер программы
- г) объем места, занимаемого на диске

д) независимость модулей

Ответ: а, б, г

#### **Задание 4.**

Какие из перечисленных пунктов являются методами рефакторинга?

а) выделение локальной переменной

б) изменение сигнатуры метода

в) подъем поля

г) инкапсуляция поля

д) выделение класса

Ответ: а, б, в, г, д

#### **Задание 5.**

Какие из перечисленных пунктов являются наиболее очевидными причинами для рефакторинга кода?

а) отсутствие комментариев

б) дублирование кода

в) избыточные временные переменные

г) длинный список параметров

Ответ: б, в, г

#### **Задание 6.**

Установите соответствие между методами оптимизации и их определениями:

1 устранение избыточного кода

а) оптимизация, направленная на удаление лишних или неиспользуемых частей кода, которые не добавляют ценности для приложения

2 изменение порядка выполнения операций

б) процесс идентификации и удаления элементов кода, которые не выполняют полезных функций в программе

3 изъятие неиспользуемого кода

в) перераспределение операций таким образом, чтобы порядок их выполнения был более эффективным

4 объединение похожих операций

г) процесс объединения нескольких похожих или связанных операций в одну, что упрощает код и делает его более понятным

Ответ: 1б, 2в, 3а, 4г

#### **Задание 7.**

Как называется процесс преобразования программного кода, приводящий к улучшению его характеристик и повышению производительности программы?

Ответ: оптимизация

#### **Задание 8.**

Как называется переработка исходного кода программы, чтобы он стал более простым и понятным?

Ответ: рефакторинг

#### **Задание 9.**

Проведите рефакторинг приведенного программного кода.

```
def is_palindrome(word):
    word = word.lower()
    if word == word[::-1]:
        return True
    else:
        return False
```

Ответ:

<b>Содержание верного ответа</b> (допускаются иные формулировки ответа, не искажающие его смысла)	
<pre>def is_palindrome(word):     word = word.lower()     return word == word[::-1]</pre>	
<b>Указания по оцениванию</b>	<b>Баллы</b>
Рефакторинг проведен верно	1
Все иные ситуации, включая отсутствие данного в явном виде ответа на во-прос	0
<i>Максимальный балл</i>	<i>1</i>

### Задание 10.

Проведите рефакторинг приведенного программного кода.

```
def is_upper(word):
    word1 = word.upper()
    if word == word1:
        return True
    else:
        return False
```

Ответ:

<b>Содержание верного ответа</b> (допускаются иные формулировки ответа, не искажающие его смысла)	
<pre>def is_upper(word):     return word == word.upper()</pre>	
<b>Указания по оцениванию</b>	<b>Баллы</b>
Рефакторинг проведен верно	1
Все иные ситуации, включая отсутствие данного в явном виде ответа на во-прос	0
<i>Максимальный балл</i>	<i>1</i>

### Вопрос 1

Какой из следующих элементов является частью архитектуры Android?

1. Java
2. XML
3. Gradle
4. Google Play Store

### Вопрос 2

Какой из следующих методов используется для запуска приложения Android?

1. startActivity()
2. run()
3. main()
4. launch()

Вопрос 3

Какой из следующих элементов используется для описания макета приложения Android?

1. Activity
2. View
3. Layout
4. ViewGroup

Вопрос 4

Какой из следующих элементов используется для управления состоянием приложения Android?

1. Activity
2. ViewModel
3. Fragment
4. Repository

Вопрос 5

Какой из следующих элементов используется для доступа к данным из Интернета в приложении Android?

1. HttpURLConnection
2. OkHttp
3. Retrofit
4. Gson

Вопрос 6

Какой класс используется для представления кнопки в Android?

1. Button
2. View
3. Layout

Вопрос 7

Какой класс используется для представления текстового поля в Android?

1. View
2. Layout
3. EditText

Вопрос 8

Какой класс используется для представления изображения в Android?

1. ImageView
2. View
3. Layout

Вопрос 9

Какой класс используется для представления списка в Android?

1. View
2. ListView

### 3. Layout

#### Вопрос 10

Какой класс используется для представления слайдера в Android?

1. SeekBar
2. View
3. Layout

#### Вопрос 11

В каком порядке выполняются жизненные циклы Activity?

- 1 – onResume
- 2 – onStop
- 3 – onDestroy
- 4 – onPause
- 5 – onStart
- 6 – onCreate

#### Вопрос 12

Какие методы жизненного цикла Activity являются обязательными для переопределения?

1. onCreate()
2. onStart()
3. onResume()

#### Вопрос 13

Как называется класс, который предоставляет доступ к системным ресурсам приложения?

1. Activity
2. Application
3. Context
4. View

#### Вопрос 14

Какие методы класса Context используются для доступа к системным ресурсам?

1. getResources()
2. getSystemService()
3. getPackageName()

#### Вопрос 15

Как называется класс, который представляет собой визуальный компонент приложения?

1. Activity
2. Application
3. Context
4. View

#### Вопрос 16

Как называется класс, который представляет собой контейнер для визуальных компонентов?

1. Activity
2. Application
3. Layout

#### 4. View

#### Вопрос 17

Как называется класс, который представляет собой пользовательский интерфейс приложения?

1. Activity
2. Application
3. Layout
4. UI

#### Вопрос 18

Как называется класс, который представляет собой модель данных приложения?

1. Activity
2. Application
3. Model
4. Data

#### Вопрос 19

Как называется класс, который представляет собой контроллер приложения?

1. Activity
2. Controller
3. Application
4. Control

#### Вопрос 20

Какой тип тестирования применяется для проверки функциональности приложения?

1. Функциональное тестирование
2. Нефункциональное тестирование
3. Тестирование производительности
4. Тестирование безопасности

#### Вопрос 21

Какой тип тестирования применяется для проверки производительности приложения?

1. Функциональное тестирование
2. Нефункциональное тестирование
3. Тестирование производительности
4. Тестирование безопасности

#### Вопрос 22

Какой тип тестирования применяется для проверки безопасности приложения?

1. Функциональное тестирование
2. Нефункциональное тестирование
3. Тестирование производительности
4. Тестирование безопасности

#### Вопрос 23

Какие инструменты можно использовать для тестирования мобильного приложения на Java?

1. JUnit
2. Selenium

### 3. Appium

#### Вопрос 24

Как называется фреймворк для тестирования мобильных приложений на Java?

1. JUnit
2. Selenium
3. Appium
4. Espresso

#### Вопрос 25

Как называется инструмент для автоматизации тестирования мобильных приложений на Java?

1. JUnit
2. Selenium
3. Appium
4. Espresso

#### Вопрос 26

Как называется инструмент для тестирования производительности мобильных приложений на Java?

1. JUnit
2. Selenium
3. Appium
4. JMeter

#### Вопрос 27

Как называется инструмент для тестирования безопасности мобильных приложений на Java?

1. JUnit
2. OWASP ZAP
3. Selenium
4. Appium

#### Вопрос 28

Какая из следующих методологий тестирования мобильных приложений на Java является наиболее распространенной?

1. От черного ящика
2. От белого ящика
3. Смешанная

#### Вопрос 29

Какой из следующих методов тестирования мобильных приложений на Java является наиболее эффективным для выявления ошибок пользовательского интерфейса?

1. Тестирование вручную
2. Автоматизированное тестирование
3. Тестирование производительности

#### Вопрос 30

Какой из следующих типов ошибок мобильных приложений на Java является наиболее распространенным?

1. Ошибки пользовательского интерфейса
2. Ошибки логики приложения

### 3. Ошибки производительности

Вопрос 31

Какие существуют платформы для разработки мобильных приложений?

Вопрос 32

Какие существуют инструменты для разработки мобильных приложений?

Вопрос 33

Какие существуют основные компоненты мобильного приложения?

Вопрос 34

Какие существуют основные концепции разработки мобильных приложений?

Вопрос 35

Какие существуют методы для реализации пользовательского интерфейса мобильного приложения?

Вопрос 36

Какие существуют способы хранения данных мобильного приложения?

Вопрос 37

Какие существуют методы для распространения мобильных приложений?

Вопрос 38

Что такое Activity?

Вопрос 39

Что такое Service?

Вопрос 40

Что такое BroadcastReceiver?

Вопрос 41

Что такое ContentProvider?

#### **Система оценивания заданий для проведения диагностической работы по дисциплине МДК.01.03 Разработка мобильных приложений**

За правильный ответ на задания 1- 30 ставится 1 балл, за неверный ответ или его отсутствие – 0 баллов.

<b>Номер задания</b>	<b>Правильный ответ</b>
1	1,2,3,4
2	1
3	3
4	2
5	2
6	1
7	3
8	1
9	2
10	1
11	1-6, 2-5, 3-1, 4-4, 5-2, 6-3
12	1, 2, 3

13	3
14	1, 2, 3
15	4
16	3
17	4
18	3
19	2
20	1
21	3
22	4
23	1, 2, 3
24	4
25	3
26	4
27	2
28	1
29	2
30	1

### **Критерии оценивания выполнения заданий с развернутым ответом**

#### **Задание 31**

Правильный ответ:

Существуют следующие платформы для разработки мобильных приложений:

- Android - это платформа для разработки мобильных приложений для устройств под управлением операционной системы Android.
- iOS - это платформа для разработки мобильных приложений для устройств под управлением операционной системы iOS.
- Windows Phone - это платформа для разработки мобильных приложений для устройств под управлением операционной системы Windows Phone.
- BlackBerry - это платформа для разработки мобильных приложений для устройств под управлением операционной системы BlackBerry.

Обучающийся должен знать о перечисленных платформах. Допускается, если одну из них при ответе обучающийся забудет упомянуть. В этом случае начисляется 1 балл.

#### **Задание 32**

Правильный ответ:

Существуют следующие инструменты для разработки мобильных приложений:

- Android Studio - это интегрированная среда разработки (IDE) для разработки мобильных приложений для Android.
- Xcode - это IDE для разработки мобильных приложений для iOS.
- Visual Studio - это IDE для разработки мобильных приложений для Windows Phone.
- RAD Studio - это IDE для разработки мобильных приложений для BlackBerry.

Обучающийся должен знать о перечисленных IDE. Допускается, если одну из них при ответе обучающийся забудет упомянуть. В этом случае начисляется 1 балл.

#### **Задание 33**

Правильный ответ:

Основные компоненты мобильного приложения:

- Пользовательский интерфейс (UI) - это компонент, который отвечает за взаимодействие пользователя с приложением.
- Логика приложения - это компонент, который отвечает за обработку данных и выполнение действий приложения.
- Данные приложения - это компонент, который хранит данные приложения. Обучающийся должен знать все компоненты. В этом случае начисляется 1 балл.

#### **Задание 34**

Правильный ответ:

Основные концепции разработки мобильных приложений:

- Адаптивность - это способность приложения адаптироваться к различным размерам и разрешениям экранов.
- Оптимизация производительности - это способность приложения работать быстро и плавно.
- Безопасность - это способность приложения защитить данные пользователя. Обучающийся должен знать все концепции. В этом случае начисляется 1 балл.

#### **Задание 35**

Правильный ответ:

Существуют следующие методы для реализации пользовательского интерфейса мобильного приложения:

- Пользовательский интерфейс на основе XML - это метод, при котором пользовательский интерфейс приложения определяется в файлах XML.
- Пользовательский интерфейс на основе Java - это метод, при котором пользовательский интерфейс приложения определяется в Java-коде.
- Пользовательский интерфейс на основе фреймворка - это метод, при котором пользовательский интерфейс приложения определяется с помощью фреймворка для разработки мобильных приложений.

Обучающийся должен знать все способы. В этом случае начисляется 1 балл.

#### **Задание 36**

Правильный ответ:

Существуют следующие способы хранения данных мобильного приложения:

- Память устройства - это способ хранения данных, при котором данные хранятся в памяти устройства пользователя.
- Файлы - это способ хранения данных, при котором данные хранятся в файлах на устройстве пользователя.
- База данных - это способ хранения данных, при котором данные хранятся в базе данных на устройстве пользователя или в облаке.

Обучающийся должен знать все способы. В этом случае начисляется 1 балл.

#### **Задание 37**

Правильный ответ:

Существуют следующие методы для распространения мобильных приложений:

- Магазин приложений - это метод, при котором приложение публикуется в магазине приложений, таком как Google Play или App Store.
- Прямое распространение - это метод, при котором приложение распространяется напрямую пользователю через веб-сайт или электронную почту.

Обучающийся должен знать все способы. В этом случае начисляется 1 балл.

#### **Задание 38**

Правильный ответ:  
 Основной компонент мобильного приложения, который отвечает за отображение пользовательского интерфейса.  
 В случае верного ответа начисляется 1 балл.

**Задание 39**

Правильный ответ:  
 Фоновый компонент мобильного приложения, который выполняет задачи, не требующие постоянного взаимодействия с пользователем.  
 В случае верного ответа начисляется 1 балл.

**Задание 40**

Правильный ответ:  
 Компонент мобильного приложения, который получает уведомления о системных событиях.  
 В случае верного ответа начисляется 1 балл.

**Задание 41**

Правильный ответ:  
 Компонент мобильного приложения, который предоставляет доступ к данным другим приложениям.  
 В случае верного ответа начисляется 1 балл.

Формируемая компетенция	Показатели освоения компетенций
<p><b>ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.</b></p>	<p><b>Знать:</b></p> <ul style="list-style-type: none"> <li>• Основные этапы разработки программного обеспечения.</li> <li>• Основные принципы технологии структурного и объектно-ориентированного программирования.</li> </ul> <p><i>Дополнительно для квалификаций "Программист":</i></p> <ul style="list-style-type: none"> <li>• Знание API современных мобильных операционных систем.</li> </ul> <p><b>Уметь:</b></p> <ul style="list-style-type: none"> <li>• Создавать программу по разработанному алгоритму как отдельный модуль.</li> <li>• Оформлять документацию на программные средства.</li> </ul> <p><i>Дополнительно для квалификаций "Программист":</i></p> <ul style="list-style-type: none"> <li>• Осуществлять разработку кода программного модуля на языках низкого уровня и высокого уровней в том числе для мобильных платформ.</li> </ul> <p><b>Практические умения:</b></p> <ul style="list-style-type: none"> <li>• Разрабатывать код программного продукта на основе готовой спецификации на уровне модуля.</li> </ul> <p><i>Дополнительно для квалификаций "Программист":</i></p>

	<i>мис</i> "": <ul style="list-style-type: none"> <li>• Разрабатывать мобильные приложения.</li> </ul>
--	--

**Задание 1.** Перечислите ниже основные этапы разработки программного обеспечения.

---

Ответ:

- Анализ, составление требований к продукту.
- Планирование.
- Проектирование и дизайн.
- Разработка.
- Тестирование.
- Развертывание, эксплуатация.

**Задание 2.** Установите соответствие между названием принципа структурного программирования и его характеристикой.

1. Этот принцип предполагает строгий методический подход к программированию, придает творческому процессу определенную строгость и дисциплину	А. Принцип абстрактности
2. Этот принцип предполагает, что взаимосвязь между частями программы должна носить подчиненный характер.	Б. Принцип формальности
3. Этот принцип позволяет разработчику рассматривать программу в нужный момент без лишней детализации. Детализация увеличивается при переходе от верхнего уровня абстракции к нижнему.	В. Принцип модульности
4. В соответствии с этим принципом программа разделяется на отдельные законченные фрагменты, модули, которые просты по управлению и допускают независимую отладку и тестирование. В результате отдельные ветви программы могут создаваться разными группами программистов.	Г. Принцип иерархического упорядочения

Ответ: 1 – Б, 2 – Г, 3 – А, 4 – В.

**Задание 3.** Перечислите ниже основные принципы объектно-ориентированного программирования.

---

Ответ:

- Абстракция
- Инкапсуляция.
- Наследование.

- Полиморфизм.

**Задание 4.** Что такое API (определение)?

---

Ответ: **API** – это набор способов и правил, по которым различные программы общаются (взаимодействуют) между собой и обмениваются данными.

**Задание 5.** Что такое PowerShell?

1. База данных конфигурации загрузки
2. Язык сценариев для автоматизации процессов управления системами
3. Объект режима ядра

Ответ: 2.

**Задание 6.** Для чего используется NtDuplicateObject?

- a. Для дублирования описателей из одного процесса в другой;
- b. Для обмена с процессом подсистемы, когда возникающие ошибки не обрабатываются подключившимся отладчиком;
- c. Для обмена с отладчиком при передаче ему управления процессом после исключения.

Ответ: a.

**Задание 7.** Что из перечисленного не делает HAL?

- a. Изучает конфигурацию системы.
- b. Настраивает передачи данных в режиме DMA и управляет ею независимым от устройств образом.
- c. Предоставляет поддержку низкого уровня для двух классов механизмов синхронизации.

Ответ: c.

**Задание 8.** Что делает диспетчер кэширования (cache manager)?

- a. Оптимизирует производительность ввода-вывода в файловой системе.
- b. Реализует архитектуру виртуальной памяти с подкачкой по требованию.
- c. Обеспечивает работу сложных механизмов безопасности Windows.

Ответ: a.

**Задание 9.** Верно ли утверждение:

Каждый поток может выполнять определенное подмножество волокон, создавая связь типа «один ко многим» между потоками и волокнами (для упрощения синхронизации).

- a. Да.
- b. Нет.
- c. Затрудняюсь ответить.

Ответ: a.

**Задание 10.** Какой самый нижний уровень ядра?

- a. Гипервизор.
- b. NTOS.

с. HAL.

Ответ: а.

**Задание 11.** Как называется один из механизмов синхронизации?

- а. Критической секцией.
- б. Потоки.
- с. Почтовые слоты.

Ответ: а.

**Задание 12.** Какие категории типов объектов «Ввод-вывод» режима ядра?

- а. Семафоры, мьютексы, события, порты ИРС, очереди ввода-вывода.
- б. Файлы, устройства, драйверы, таймеры.
- с. Задания, процессы, потоки, сегменты, маркеры.

Ответ: б.

**Задание 13.** Для чего в Windows реализован алгоритм DFSS?

- а. Не дает сессиям работать чрезмерно.
- б. Поддерживается несколько интерактивных сессий.
- с. Ускоряет процесс.

Ответ: б.

**Задание 14.** Напишите ниже, что называется процессом.

---

Ответ: Процесс – это программа или команда, выполняемая на компьютере.

**Задание 15.** Поток в многозадачной ОС может находиться в \_\_\_\_\_ состояниях (впишите число).

Ответ: 3-х.

**Задание 16.** Активное состояние потока, во время которого поток обладает всеми необходимыми ресурсами и непосредственно выполняется процессором называется \_\_\_\_\_.

Ответ: Выполнением.

**Задание 17.** Характеристика потока – это

- а. Параллельная работа, неблокирующие системные вызовы, прерывания.
- б. Параллельная работа, блокирующие системные вызовы.
- с. Отсутствие параллельной работы, блокирующие системные вызовы.

**Задание 17.** Поток называется \_\_\_\_\_.

Ответ: абстракция, используемая для чтения или записи файлов, сокетов и т.п. в единой манере или базовая единица, которой операционная система выделяет процессорное время.

**Задание 18.** Пассивное состояние потока, при котором поток заблокирован в связи с внешним по отношению к нему обстоятельством (имеет все требуемые для него ресурсы, который готов выполняться, однако процессор занят выполнением другого потока) называется \_\_\_\_\_.

Ответ: готовностью.

**Задание 19.** В UNIX системный вызов, который приказывает операционной системе завершить некоторые другие процессы, называется

- a. Kill.
- b. Terminate Processor.
- c. Fork.

Ответ: a.

**Задание 20.** В Windows для отображения списка запущенных процессов может использоваться

- a. Программа ps
- b. Диспетчер задач
- c. Команда top.

Ответ: b.

Формируемая компетенция	Показатели освоения компетенций
<p><b>ПК.1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.</b></p>	<p><b>Знать:</b></p> <ul style="list-style-type: none"> <li>• Основные принципы отладки и тестирования программных продуктов.</li> <li>• Инструментарий отладки программных продуктов.</li> </ul> <p><b>Уметь:</b></p> <ul style="list-style-type: none"> <li>• Выполнять отладку и тестирование программы на уровне модуля.</li> <li>• Оформлять документацию на программные средства.</li> </ul> <p><i>Дополнительно для квалификаций "Программист" и "Специалист по тестированию в области информационных технологий":</i></p> <ul style="list-style-type: none"> <li>• Применять инструментальные средства отладки программного обеспечения.</li> </ul> <p><b>Практические умения:</b></p> <ul style="list-style-type: none"> <li>• Использовать инструментальные средства на этапе отладки программного продукта.</li> <li>• Проводить тестирование программного модуля по определенному сценарию.</li> </ul>

**Задание 1.** Событие, приводящее к созданию процессов – это

- a. Выход при возникновении ошибки.
- b. Выполнение работающим процессом системного вызова, предназначенного для создания процесса.
- c. Возникновение фатальной ошибки.

Ответ: b.

**Задание 2.** Причина завершения процесса – это

- a. Запрос пользователя на создание нового процесса.
- b. Инициализация системы.
- c. Уничтожение другим процессом.

Ответ: с.

**Задание 3.** Какую информацию об управлении процессами ОС запишет в таблице процессов?

- a. Дескрипторы файлов.
- b. Указатель на информацию о сегменте данных.
- c. Приоритет.

Ответ: с.

**Задание 4.** Известно, что программа А выполняется в монопольном режиме за 10 минут, а программа В – за 20 минут, то есть при последовательном выполнении этих программ потребуется 30 минут. Если  $T$  — время выполнения обеих этих задач в режиме мультипрограммирования, то какое из неравенств, приведенных ниже, справедливо?

- a.  $0 < T < 30$
- b.  $T > 30$
- c.  $T < 20$

Ответ: а.

**Задание 5.** Пассивное состояние потока, находясь в котором, поток заблокирован по своим внутренним причинам (ждет осуществления некоторого события, например, завершения операции ввода-вывода, получения сообщения от другого потока или освобождения какого-либо необходимого ему ресурса) называется ...

Ответ: ожиданием.

**Задание 6.** Исполняемый экземпляр приложения и комплект ресурсов, отводящийся данному исполняемому приложению, называется ...

- a. Поток
- b. Процесс
- c. Прерыванием

Ответ: b.

**Задание 7.** Внутренняя составляющая процесса, которой операционная система выделяет процессорное время для выполнения кода называется ...

- a. Поток
- b. Процесс
- c. Прерыванием

Ответ: b.

**Задание 7.** В соответствии с алгоритмом квантования времени при планировании потоков смена потока происходит, если ...

- a. поток завершился и покинул систему
- b. поток перешел в состояние выполнения

с. смена потока никогда не будет происходить

Ответ: а.

**Задание 8.** Установите соответствие

1. Элементы, присущие каждому процессу, – это	А. Регистры
2. Элементы, присущие каждому потоку, – это	Б. Адресное пространство

Ответ: 1 – Б, 2 – А.

**Задание 9.** Для чего предназначен empty семафор?

---

Ответ: предназначен для подсчета количества пустых мест в буфере.

**Задание 10.** Подобная ситуация, когда два или более процесса считывают или записывают какие-нибудь общие данные, а окончательный результат зависит от того, какой процесс и когда именно выполняется, называется \_\_\_\_\_.

Ответ: состязательной ситуацией.

**Задание 11.** Во сколько раз во многих системах создание потоков осуществляется быстрее, чем создание процессов?

- а. 100 – 1000 раз
- б. 1000 – раз
- с. 10 – 100 раз

Ответ: с.

**Задание 12.** В процессе загрузки операционной системы происходит: ...

- а. копирование файлов операционной системы с гибкого диска на жёсткий диск.
- б. копирование файлов операционной системы с CD – диска на жёсткий диск.
- с. последовательная загрузка файлов операционной системы в оперативную память.

Ответ: с.

**Задание 13.** Характеристика потока – это

- а. параллельная работа, неблокирующие системные вызовы, прерывания
- б. параллельная работа, блокирующие системные вызовы
- с. отсутствие параллельной работы, блокирующие системные вызовы

Ответ: б.

**Задание 14.** Две операции с семафорами – это ...

- а. sleep и wakeup
- б. wait и signal
- с. down и up

Ответ: с.

**Задание 15.** Какой семафор предназначен для подсчета количества заполненных мест в буфере?

- a. empty
- b. mutex
- c. full

Ответ: с.

**Задание 16.** Передача сообщений – это ...

- a. коллекция переменных и структур данных, сгруппированных вместе в специальную разновидность модуля или пакета процедур;
- b. метод взаимодействия процессов, использующий два примитива send и receive;
- c. совместно используемая переменная, которая может находиться в одном из двух состояний: заблокированном или незаблокированном.

Ответ: b.

**Задание 17.**

1. Механизм синхронизации, предназначенный для групп процессов, называется ...	А. Семафор
2. Целочисленная переменная Дейкстра используемая для подсчета количества активизаций, отложенных на будущее, называется ...	Б. Мьютекс
3. Совместно используемая переменная, которая может находиться в одном из двух состояний: заблокированном или незаблокированном, называется ...	В. Фьютекс

Ответ: 1 – В, 2 – А, 3 – Б.

**Задание 18.** В какой среде с пользователями требуется приоритетность?

- a. Интерактивной
- b. Пакетной
- c. Реального времени

Ответ: а.

**Задание 19.** Как называется ситуация, при которой все программы бесконечно работают, но не могут ничего добиться?

- a. Голодание
- b. Приостановка
- c. Циклическое планирование

Ответ: а.

**Задание 20.** Системы реального времени обычно делятся на ...

- a. Гибкие и мягкие
- b. Жесткие и упругие
- c. Жесткие и гибкие

Ответ: с.

**Составитель:** Алутина Е.Ф., к.ф.-м.н., доцент кафедры информатики и методики преподавания информатики

## **5 ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ**

**Утверждение изменений и дополнений в РПД для реализации в 2023/2024 уч. г.**  
РПД обсуждена и одобрена для реализации в 2023/2024 уч. г. на заседании кафедры информатики и методики преподавания информатики (протокол №9 от 26 июня 2023 г.).