

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Щёкина Вера Витальевна

Должность: Ректор

Дата подписания: 19.07.2021 08:56:34

Уникальный программный ключ:

a2232a55157e57651a48999f190892af53989420420336ffbf573a434e57789



**МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное  
учреждение высшего образования**

**«Благовещенский государственный педагогический университет»**

**ПРОГРАММА ПОДГОТОВКИ СПЕЦИАЛИСТОВ  
СРЕДНЕГО ЗВЕНА**

**Рабочая программа дисциплины**

**УТВЕРЖДАЮ**

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

 **Т.А. Мерделина**

**«29» декабря 2021 г**

**Рабочая программа учебной дисциплины**

**МДК.01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ**

**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 ДИСЦИПЛИНЫ .....</b> | <b>4</b>  |
| <b>3 УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ .....</b>             | <b>6</b>  |
| <b>4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....</b>  | <b>7</b>  |
| <b>5 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ПО ДИСЦИПЛИНЕ .....</b>         | <b>8</b>  |
| <b>6 ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ .....</b>                       | <b>20</b> |

## 1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

**1.1 Цель дисциплины:** освоение студентами основного вида деятельности «Разработка модулей программного обеспечения для компьютерных систем» и соответствующих ей профессиональных компетенций.

**1.2 Место дисциплины в структуре ООП:**

Учебная дисциплина «Разработка программных модулей» (МДК.01.01) принадлежит к профессиональному циклу и основному виду деятельности «Разработка модулей программного обеспечения для компьютерных систем» (ПМ.01).

**1.3 Дисциплина направлена на формирование следующих компетенций:**

- 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 1.5 Осуществлять рефакторинг и оптимизацию программного кода.

**1.4 Перечень планируемых результатов обучения.** В результате изучения дисциплины обучающийся должен

**уметь:**

- распознавать задачу и/или проблему в профессиональном контексте;
- анализировать задачу и/или проблему и выделять её составные части;
- определять этапы решения задачи;
- выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы;
- осуществлять разработку кода программного модуля на языке высокого уровня;
- создавать программу по разработанному алгоритму как отдельный модуль;
- осуществлять разработку кода программного модуля на современных языках программирования;
- уметь выполнять оптимизацию и рефакторинг программного кода;

**знать:**

- основные этапы разработки программного обеспечения;
- основные принципы технологии структурного и объектно-ориентированного программирования;
- способы оптимизации и приемы рефакторинга.

**иметь практический опыт в:**

- разработке кода программного продукта на основе готовой спецификации на уровне модуля.

**1.5 Общая трудоемкость** дисциплины «Разработка программных модулей» составляет 258 ч. максимальной учебной нагрузки обучающегося, в том числе: обязательной аудиторной учебной нагрузки обучающегося – 220 часов; самостоятельной работы обучающегося – 30 часов.

Программа предусматривает изучение материала на лекциях и лабораторных. Предусмотрена самостоятельная работа обучающихся по темам и разделам. Программа предусматривает использование в образовательном процессе активных и интерактивных форм проведения занятий в сочетании с внеаудиторной работой. Проверка знаний осуществляется фронтально, индивидуально.

## 1.6 Объем дисциплины и виды учебной деятельности

| Вид учебной работы   | Объем часов |
|--|-------------|
| <b>Максимальная учебная нагрузка (всего)</b>                       | <b>258</b>  |
| <b>Обязательная аудиторная учебная нагрузка (всего)</b>            | <b>220</b>  |
| в том числе:   |             |
| - лекции   | 112         |
| - лабораторные занятия   | 108         |
| <b>Самостоятельная работа обучающегося (всего)</b>                 | <b>30</b>   |
| <b>Консультации</b>  | <b>2</b>    |
| <b>Промежуточная аттестация: дифференцированный зачет, экзамен</b> | <b>6</b>    |

## 2 ТЕМАТИЧЕСКИЙ ПЛАН И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

| Наименование разделов и тем                              | Содержание учебного материала, лабораторные работы, самостоятельная работа обучающегося  | Объем в часах |
|--|--|---------------|
| <b>Тема 1. Жизненный цикл ПО</b>                         | <i>Лекционные занятия:</i><br>Понятие ЖЦ ПО. Этапы ЖЦ ПО.  | 2             |
| <b>Тема 2. Структурное программирование</b>              | <i>Лекционные занятия:</i><br>Технология структурного программирования.<br>Инструментальные средства оформления и документирования алгоритмов программ.  | 8             |
|  | <i>Лабораторные занятия:</i><br>Оценка сложности алгоритма: классификация, классы алгоритмов, неразрешимые задачи.<br>Оценка сложности рекурсивных алгоритмов.<br>Оценка сложности эвристических алгоритмов.   | 18            |
|  | <i>Самостоятельная работа обучающихся:</i><br>Оценка сложности алгоритмов  | 4             |
| <b>Тема 3. Объектно-ориентированное программирование</b> | <i>Лекционные занятия:</i><br>Основные принципы объектно-ориентированного программирования. Классы: основные понятия.<br>Перегрузка методов.<br>Операции класса.<br>Иерархия классов.<br>Синтаксис интерфейсов.<br>Интерфейсы и наследование.<br>Структуры.<br>Делегаты.<br>Регулярные выражения.<br>Коллекции. Параметризованные классы.<br>Указатели<br>Операции со списками | 24            |

|   |   |    |
|---|---|----|
|   | <p><i>Лабораторные занятия:</i><br/> Работа с классами.<br/> Перегрузка методов.<br/> Определение операций в классе.<br/> Создание наследованных классов.<br/> Работа с объектами через интерфейсы.<br/> Использование стандартных интерфейсов.<br/> Работа с типом данных структура.<br/> Коллекции. Параметризованные классы.<br/> Использование регулярных выражений.<br/> Операции со списками.</p> | 20 |
|   | <p><i>Самостоятельная работа обучающихся:</i><br/> Синтаксис описания классов, интерфейсов, структур. Коллекции. Параметризованные классы.</p>  | 4  |
| <b>Тема 4. Паттерны проектирования</b>                | <p><i>Лекционные занятия:</i><br/> Назначение и виды паттернов.<br/> Основные шаблоны.<br/> Порождающие шаблоны.<br/> Структурные шаблоны.<br/> Поведенческие шаблоны.</p>  | 30 |
|   | <p><i>Лабораторные занятия:</i><br/> Использование основных шаблонов.<br/> Использование порождающих шаблонов.<br/> Использование структурных шаблонов.<br/> Использование поведенческих шаблонов.</p>  | 12 |
|   | <p><i>Самостоятельная работа обучающихся:</i><br/> Паттерны проектирования</p>  | 6  |
|   |   |    |
| <b>Тема 5. Событийно-управляемое программирование</b> | <p><i>Лекционные занятия:</i><br/> Событийно-управляемое программирование.<br/> Элементы управления. Диалоговые окна. Обработчики событий.<br/> Введение в графику.</p>   | 20 |
|   | <p><i>Лабораторные занятия:</i><br/> Разработка приложения с использованием текстовых компонентов.<br/> Разработка приложения с несколькими формами.<br/> Разработка приложения с не визуальными компонентами.<br/> Разработка игрового приложения.<br/> Разработка приложения с анимацией.</p>   | 8  |
|   | <p><i>Самостоятельная работа обучающихся:</i><br/> Событийно-управляемое программирование</p>   | 4  |
|   |   |    |
| <b>Тема 6. Оптимизация и рефакторинг кода</b>         | <p><i>Лекционные занятия:</i><br/> Методы оптимизации программного кода.<br/> Цели и методы рефакторинга.</p>   | 4  |
|   | <p><i>Лабораторные занятия:</i><br/> Оптимизация и рефакторинг кода.</p>  | 20 |
|   | <p><i>Самостоятельная работа обучающихся:</i><br/> Оптимизация кода</p>   | 4  |
| <b>Тема 7. Разработка</b>                             | <p><i>Лекционные занятия:</i></p>   | 8  |

|  |   |            |
|--|---|------------|
| <b>пользовательского интерфейса</b>                                | Правила разработки интерфейсов пользователя   |            |
|  | <i>Лабораторные занятия:</i><br>Разработка интерфейса пользователя  | 10         |
|  | <i>Самостоятельная работа обучающихся:</i><br>Проектирование интерфейса пользователя  | 4          |
| <b>Тема 8. Основы ADO.Net</b>                                      | <i>Лекционные занятия:</i><br>Работа с базами данных.<br>Доступ к данным.<br>Создание таблицы, работа с записями.<br>Способы создания команд. | 16         |
|  | <i>Лабораторные занятия:</i><br>Создание приложения с БД.<br>Создание запросов к БД.<br>Создание хранимых процедур.                           | 20         |
|  | <i>Самостоятельная работа обучающихся:</i><br>Основы построения БД. Доступ к БД из приложения.  | 4          |
| <b>Консультации</b>  |   | <b>2</b>   |
| <b>Промежуточная аттестация: дифференцированный зачет, экзамен</b> |   | <b>2+4</b> |
| <b>Всего:</b>  |   | <b>258</b> |

### 3 УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ

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

Реализация учебной дисциплины требует наличия компьютерного класса – 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

##### Основная литература

1. Гордеев, С. И. Организация баз данных в 2 ч. Часть 1 : учебник для среднего профессионального образования / С. И. Гордеев, В. Н. Волошина. – 2-е изд., испр. и доп. – Москва : Издательство Юрайт, 2021. – 310 с. – (Профессиональное образование). – ISBN 978-5-534-11626-7. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/476351>

2. Гордеев, С. И. Организация баз данных в 2 ч. Часть 2 : учебник для среднего профессионального образования / С. И. Гордеев, В. Н. Волошина. – 2-е изд., испр. и доп. – Москва : Издательство Юрайт, 2021. – 513 с. – (Профессиональное образование). – ISBN 978-5-534-11625-0. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/476352>

3. 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, 2021. – 235 с. – (Профессиональное образование). – ISBN 978-5-534-05047-9. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/472502>

4. Кувшинов, Д. Р. Основы программирования : учебное пособие для среднего профессионального образования / Д. Р. Кувшинов. – Москва : Издательство Юрайт, 2021. – 105 с. – (Профессиональное образование). – ISBN 978-5-534-07560-1. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/473687>

5. Огнева, М. В. Программирование на языке C++: практический курс : учебное пособие для среднего профессионального образования / М. В. Огнева, Е. В. Кудрина. – Москва : Издательство Юрайт, 2021. – 335 с. – (Профессиональное образование). – ISBN 978-5-534-05780-5. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/473118>

#### Дополнительная литература

1. Нестеров, С. А. Базы данных : учебник и практикум для среднего профессионального образования / С. А. Нестеров. – Москва : Издательство Юрайт, 2021. – 230 с. – (Профессиональное образование). – ISBN 978-5-534-11629-8. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/476348>

2. 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>

3. Стружкин, Н. П. Базы данных: проектирование : учебник для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. – Москва : Издательство Юрайт, 2021. – 477 с. – (Профессиональное образование). – ISBN 978-5-534-11635-9. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/476340>

#### Базы данных и информационно-справочные системы

1. Национальная платформа «Открытое образование». – Режим доступа: <https://openedu.ru>

2. Образовательная платформа «Лекториум». – Режим доступа: <https://www.lektorium.tv/>

3. Электронный фонд нормативно-технической и нормативно-правовой документации Консорциума «Кодекс». – Режим доступа: <https://docs.cntd.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 дисциплины осуществляется преподавателем в процессе проведения лекционных занятий, лабораторных занятий, тестирования, собеседований, а также выполнения обучающимися лабораторных работ.

| Результаты обучения (освоенные умения, усвоенные знания, практический опыт)  | Формы и методы контроля результатов обучения         |
|--|--|
| <b>Умения:</b><br>– осуществлять разработку кода программного модуля на языке высокого уровня;<br>– создавать программу по разработанному алгоритму как отдельный модуль;<br>– осуществлять разработку кода про- | Тест<br><br>Собеседование<br><br>Лабораторная работа |

|  |  |
|--|--|
| <p>граммного модуля на современных языках программирования;</p> <p>– уметь выполнять оптимизацию и рефакторинг программного кода.</p> <p><b>Знания:</b></p> <p>– основные этапы разработки программного обеспечения;</p> <p>– основные принципы технологии структурного и объектно-ориентированного программирования;</p> <p>– способы оптимизации и приемы рефакторинга.</p> <p><b>Иметь практический опыт в:</b></p> <p>– разработке кода программного продукта на основе готовой спецификации на уровне модуля.</p> |  |
|--|--|

## 5 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ПО ДИСЦИПЛИНЕ

| <b>Формируемая компетенция</b>   | <b>Показатели освоения компетенций</b>  |
|--|---|
| <p>ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.</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> <li>• актуальная нормативно-правовая база в области документирования алгоритмов.</li> </ul> <p><b>Уметь:</b></p> <ul style="list-style-type: none"> <li>• формировать алгоритмы разработки программных модулей в соответствии с техническим заданием;</li> <li>• оформлять документацию на программные средства;</li> <li>• оценка сложности алгоритма.</li> </ul> <p><b>Владеть:</b></p> <ul style="list-style-type: none"> <li>• разрабатывать алгоритм решения поставленной задачи и реализовывать его средствами автоматизированного проектирования.</li> </ul> |

**Задание 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

**Задание 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

**Задание 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

**Задание 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

**Задание 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

**Задание 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

**Задание 7.** Какая модель жизненного цикла программного обеспечения является наиболее распространенной?

1. Водопад
2. Итеративная
3. Адаптивная

**Задание 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

**Задание 9.** Какая модель жизненного цикла программного обеспечения позволяет разрабатывать программное обеспечение в параллельном режиме?

1. Водопад
2. Итеративная
3. Адаптивная

**Задание 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

**Задание 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

**Задание 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.

| Формируемая компетенция   | Показатели освоения компетенций   |
|---|---|
| <p><b>ПК 1.2.</b> Разрабатывать программные модули в соответствии с техническим заданием.</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> <li>• 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> <li>• осуществлять разработку кода программного модуля на языках низкого уровня и высокого уровня в том числе для мобильных платформ.</li> </ul> <p><b>Владеть:</b></p> <ul style="list-style-type: none"> <li>• разрабатывать код программного продукта на основе готовой спецификации на уровне модуля;</li> <li>• разрабатывать мобильные приложения.</li> </ul> |

**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
    [REDACTED]
    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

| <b>Формируемая компетенция</b>   | <b>Показатели освоения компетенций</b>   |
|--|--|
| <b>ПК 1.5.</b> Осуществлять рефакторинг и оптимизацию программного кода. | <p><b>Знать:</b></p> <ul style="list-style-type: none"> <li>• способы оптимизации и приемы рефакторинга;</li> <li>• инструментальные средства анализа алгоритма.</li> <li>• методы организации рефакторинга и оптимизации кода;</li> <li>• принципы работы с системой контроля версий.</li> </ul> <p><b>Уметь:</b></p> <ul style="list-style-type: none"> <li>• выполнять оптимизацию и рефакторинг программного кода;</li> <li>• работать с системой контроля версий.</li> </ul> <p><b>Владеть:</b></p> <ul style="list-style-type: none"> <li>• анализом алгоритмов, в том числе с применением инструментальных средств;</li> <li>• рефакторингом и оптимизацией программного кода.</li> </ul> |

**Задание 30.** 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 заголовок 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 AU-
TO_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.

**Составитель:** Ягелло А.А., старший преподаватель кафедры информатики и методики преподавания информатики

## **6 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.).