

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Б1.В.ДВ.02.01
(индекс дисциплины)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Программирование на Java (Джава) 1

(наименование дисциплины)

по направлению подготовки

09.03.03 Прикладная информатика

направленность (профиль)

Программирование и UX/UI-дизайн

Форма обучения: очная

Год набора: 2026

Общая трудоемкость: 5 ЗЕ

Распределение часов дисциплины по семестрам

Семестр	4	Итого
Форма контроля	экзамен	
Вид занятий		
Лекции	32	32
Лабораторные		
Практические	32	32
Руководство: курсовые работы (проекты) / РГР		
Промежуточная аттестация	0,35	0,35
Контактная работа	64,35	64,35
Самостоятельная работа	80	80
Контроль	35,65	35,65
Итого	180	180

Рабочую программу составил(и):

Доцент института цифровых технологий, канд. пед. наук Ерофеева Е. А.

(должность, ученое звание, степень, Фамилия И.О.)

Рецензирование рабочей программы дисциплины:



Отсутствует



Рецензент

(должность, ученое звание, степень, Фамилия И.О.)

Рабочая программа дисциплины составлена на основании ФГОС ВО и учебного плана направления подготовки

09.03.03 Прикладная информатика

Срок действия рабочей программы дисциплины до «31» августа 2030 г.

СОГЛАСОВАНО

Руководитель центра дизайна

«__» _____ 20__ г.

(подпись)

М.С. Кузьмина

(И.О. Фамилия)

УТВЕРЖДЕНО

На заседании института цифровых технологий

(протокол заседания № 1 от «05» сентября 2025 г.).

1. Цель освоения дисциплины

Цель освоения дисциплины – освоение технологии разработки, тестирования и развертывания программного обеспечения на платформе Java SE.

2. Место дисциплины в структуре ОПОП ВО

Дисциплины и практики, на освоении которых базируется данная дисциплина: Основы программирования, Объектно-ориентированное программирование.

Дисциплины и практики, для которых освоение данной дисциплины необходимо как предшествующее: Программирование на Java (Джава) 2, Производственная практика (технологическая (проектно-технологическая) практика) 1, Производственная практика (технологическая (проектно-технологическая) практика) 2, Производственная практика (технологическая (проектно-технологическая) практика) 3, Производственная практика (преддипломная практика).

3. Планируемые результаты обучения

Формируемые и контролируемые компетенции (код и наименование)	Индикаторы достижения компетенций (код и наименование)	Планируемые результаты обучения
ПК-5 Способен осуществлять выбор языка программирования и моделировать решение для реализации программного обеспечения	ПК-5.1 Знает технологии разработки и ведения баз данных	Знать:технологии моделирования программного обеспечения Уметь:применять технологии моделирования программного обеспечения Владеть:навыками моделирования программного обеспечения
	ПК-5.2 Умеет проектировать и разрабатывать базы данных, использовать их для поддержки информационного обеспечения решения прикладных задач	Знать:технологию разработки программного обеспечения на языках программирования Уметь:выбирать и моделировать решения по разработке программного обеспечения на языках программирования Владеть:навыками реализации программного обеспечения на языках программирования
	ПК-5.3 Владеет навыками ведения базы данных и поддержки информационного обеспечения решения прикладных задач	Знать:технологии моделирования решения для реализации программного обеспечения

Формируемые и контролируемые компетенции (код и наименование)	Индикаторы достижения компетенций (код и наименование)	Планируемые результаты обучения
		на языках программирования Уметь: моделировать решения для реализации программного обеспечения на языках программирования Владеть: инструментом моделирования решения для реализации программного обеспечения на языках программирования

4. Структура и содержание дисциплины

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля (наименование оценочного средства)
1 Введение в Java SE	Лек 1	Введение в Java	4	2	10	-	Отчет по практической работе 1
	Пр 1	Утилиты JDK	4	2		-	
	Пр 2	Утилиты JDK	4	2			
2 Основы синтаксиса Java	Лек 2	Основы синтаксиса Java.	4	2	10	-	Отчет по практической работе 2
	Лек 3	Основы синтаксиса Java.	4	2		-	
	Пр 3	Основы синтаксиса Java	4	2		-	
	Пр 4	Основы синтаксиса Java	4	2		-	
3. Объектно-ориентированное программирования в Java	Лек 4	Объектно- ориентированное программирования в Java	4	2	10	-	Отчет по практической работе 3
	Лек 5	Объектно-ориентированное программирование в Java	4	2		-	
	Пр 5	Объектно-ориентированное программирование в Java	4	2		-	
	Пр 6	Объектно-ориентированное программирование в Java	4	2		-	
4. Интерфейсы	Лек 6	Интерфейсы	4	2	12	-	Отчет по практической работе 4
	Лек 7	Интерфейсы	4	2		-	
	Пр 7	Интерфейсы	4	2		-	
	Пр 8	Интерфейсы	4	2			
5. Обработка строк на Java	Лек 8	Обработка строк на Java	4	2	12	-	Отчет по практической работе 5
	Лек 9	Обработка строк на Java	4	2		-	
	Пр 9	Обработка строк на Java	4	2		-	
	Пр 10	Обработка строк на Java	4	2			

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля (наименование оценочного средства)
6. Работа с датой и временем на Java	Лек 10	Работа с датой и временем на Java	4	2		-	
	Лек 11	Работа с датой и временем на Java	4	2		-	
	Пр 11	Работа с датой и временем на Java	4	2	12	-	Отчет по практической работе 6
	Пр 12	Работа с датой и временем на Java	4	2		-	
7. Обобщенные типы	Лек 12	Обобщенные типы.	4	2		-	
	Лек 13	Обобщенные типы.	4	2		-	
	Пр 13	Обобщенные типы	4	2	12	-	Отчет по практической работе 7
	Пр 14	Обобщенные типы	4	2		-	
8. Java Collection Framework	Лек 14	Java Collection Framework 1	4	2		-	
	Лек 15	Java Collection Framework 1	4	2		-	
	Лек 16	Java Collection Framework 2	4	2		-	
	Пр 15	Java Collection Framework	4	2	12	-	Отчет по практической работе 8
	Пр 16	Java Collection Framework	4	2			
	Ср	Работа с теоретическим материалом	4	80			
	Псщ.		4		10		
	ПА	Промежуточная аттестация	4	0,35		-	
	Контроль	Экзамен	4	35,65	100		Итоговый тест
Итого:				180			

Схема расчета итогового балла

Текущий рейтинг + Результат итогового теста и все делится на 2

5. Образовательные технологии

В рамках изучения дисциплины предусмотрено использование следующих образовательных технологий:

- технология традиционного обучения;
- интерактивные технологии: учебные дискуссии (применяются во всех модулях по итогам выполнения работ).

Технологии традиционного обучения - организация учебного процесса в вузе, основанная на лекционных и практических формах обучения: объяснительно-иллюстративное обучение. Данная технология применяется во всех модулях курса.

Технология интерактивного обучения - организация учебного процесса, которая предполагает максимальную активность обучающихся в процессе формирования ключевых компетенций. На учебной дискуссии обучающиеся представляют результат выполнения заданной работы. Проводится дискуссия по применённым решениям, обсуждается эффективность и архитектура кода.

6. Методические указания по освоению дисциплины

6.1. Рекомендации по подготовке к практическим занятиям

Обучающимся следует:

- при подготовке к занятиям обязательно использовать не только учебную литературу, но и другие источники;
- в начале занятий задать преподавателю вопросы по материалу, вызвавшему затруднения в его понимании и освоении при решении задач, заданных для самостоятельного решения;
- на занятии доводить каждую задачу до окончательного решения, демонстрировать понимание проведенных расчетов (анализов, ситуаций), в случае затруднений обращаться к преподавателю.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если обучающийся видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

6.2. Рекомендации по подготовке к экзамену

Подготовка к экзамену способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к экзамену, обучающийся ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене обучающийся демонстрирует то, что он приобрел в процессе обучения по конкретной учебной дисциплине.

На консультации перед экзаменом обучающиеся должны быть ознакомлены с основными требованиями и получить ответы на возникающие в процессе подготовки вопросы.

Необходимо ориентировать обучающихся на систематическую подготовку к занятиям в течение семестра, что позволит использовать время экзаменационной сессии для систематизации знаний.

7. Оценочные средства

7.1. Паспорт оценочных средств

Семестр	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
4	ПК-5	Тестовые задания Вопросы к экзамену Отчеты по практическим занятиям

7.2. Типовые задания или иные материалы, необходимые для текущего контроля

7.2.1. Примеры тестовых заданий

(наименование оценочного средства)

Типовые примеры заданий

1. Технология Java представляет собой:
 - ☒ высокоуровневый объектно-ориентированный язык программирования
 - ☒ программную платформу
 - ☐ низкоуровневый язык программирования
 - ☐ язык сценариев
2. Что такое JVM (Java Virtual Machine)?
 - ☒ виртуальная машина Java
 - ☐ компилятор
 - ☐ набор специальных библиотек
 - ☐ техническое обеспечение
3. Исходные файлы (.java) содержат:
 - ☒ исходный код программ на Java
 - ☐ скомпилированные Java- программ - байтовые коды, определенные спецификацией Java
 - ☐ потоковое представление текущего состояния объектов
 - ☐ наборы файлов, которые могут быть представлены в упакованном виде
4. Файлы классов (.class) содержат:
 - ☒ скомпилированные Java- программ - байтовые коды, определенные спецификацией Java
 - ☐ исходный код программ на Java
 - ☐ потоковое представление текущего состояния объектов
 - ☐ наборы файлов, которые могут быть представлены в упакованном виде
5. Файлы сериализации (.ser) содержат:
 - ☒ потоковое представление текущего состояния

- исходный код программ на Java
- скомпилированные Java- программ - байтовые коды, определенные спецификацией Java
- наборы файлов, которые могут быть представлены в упакованном виде

6. Файлы архивов (.jar) содержат:

- потоковое представление текущего состояния
- исходный код программ на Java
- скомпилированные Java- программ - байтовые коды, определенные спецификацией Java
- ⊙ наборы файлов, которые могут быть представлены в упакованном виде

7. Java Development Kit (JDK) -это:

- виртуальная машина Java
- исходный код программ на Java
- не является компонентой Java
- ⊙ набор библиотек и утилит для разработки программ на Java

8. Назначение утилиты javac?

- утилита для запуска в режиме командной строки откомпилированных программ-приложений
- утилита для запуска на исполнение и отладку апплетов без браузера. При этом не гарантируется работоспособность отлаженного апплета в браузере
- отладчик программ, написанных на языке Java
- ⊙ компилятор в режиме командной строки для программ, написанных на языке Java

9. Что будет, если ввести команду

jar -cfv jar.jar Hello.java Hello.class

- запустится класс Hello из jar-архива
- файл Hello.java скомпилируется и будет создан файл Hello.class
- ⊙ будет создан jar архив с файлами Hello.java и Hello.class
- ошибка (Error)

10. Параметр -cp позволяет?

- ☒ указать путь, по которому можно найти классы для компиляции
- ☒ указать путь, по которому можно найти классы для запуска JVM
- ☐ вывести версию компилятора
- ☐ вывести версию исходного кода

11. Что произойдет при использовании данной команды
java HelloWorld

- компиляция jar файла «HelloWorld»
- ⊙ запустится байт-код «HelloWorld»
- компиляция исходного Java-кода «HelloWorld» в байт-код
- ошибка

12. Что выполняет данная команда javac hellofromfirst.java?

- ⊙ компиляция исходного Java-кода с именем «hellofromfirst» в байт-код с таким же названием
- компиляция исходного Java-кода с именем «hellofromfirst» в jar с таким же названием

- компиляция исходного Java-кода с именем «hellofromfirst» в манифест с таким же названием
- компиляция исходного Java-кода с именем «hellofromfirst» в байт-код с другим названием

13. Базовый пакет java.lang?

- поддержка вычислений с целыми числами произвольной длины а так же с числами с плавающей точкой произвольной точности
- содержит важные классы для работы со структурами данных (работа с событиями и датами)
- ⊙ содержит базовые классы языка java
- поддержка базовых средств ввода/вывода

14. Базовый пакет java.io?

- поддержка вычислений с целыми числами произвольной длины а так же с числами с плавающей точкой произвольной точности
- содержит важные классы для работы со структурами данных (работа с событиями и датами)
- содержит базовые классы языка java
- ⊙ поддержка базовых средств ввода/вывода

15. Пакет (package) в Java это?

- ⊙ контейнер, который используется для того, чтобы изолировать имена классов
- базовый класс языка java
- базовый метод языка java
- интерфейс

16. Выберите правильный вариант значений переменных a,b,c,d чтобы результат выражения $a \&\&(b \parallel c)^d$ являлся ложью?

- a=false, b=false, c=false, d=true
- ⊙ все ответы верны
- a=true, b=true, c=false, d=true
- a=false, b=true, c=false, d=true

17. Выберите правильный вариант значений переменных a,b,c,d чтобы результат выражения $a \&\&b \parallel c^d$ являлся истиной?

- a=false, b=false, c=true, d=false
- ⊙ все ответы верны
- a=true, b=true, c=false, d=true
- a=false, b=false, c=false, d=true

18. Что означает данный оператор >> ?

- оператор беззнакового правого побитового сдвига.
- оператор левого побитового сдвига
- знак больше
- ⊙ оператор правого побитового сдвига с сохранением знака отрицательного числа

19. Что означает данный оператор >>> ?

- ⊙ оператор беззнакового правого побитового сдвига.

- знак больше
- оператор левого побитового сдвига
- оператор правого побитового сдвига с сохранением знака отрицательного числа

20. Что означает данный оператор << ?

- оператор беззнакового правого побитового сдвига.
- оператор беззнакового левого побитового сдвига
- знак меньше
- Ⓐ оператор левого побитового сдвига

21. Какое значение примет переменная result после выполнения программы?

```
int[] m = {4, 8, 10, 16, 12};
int result = 0;
for(int i=0; i<4; i++) {
    result += m[i];
}
System.out.println(result);
```

- 0
- 4
- 50
- Ⓐ 38

22. Укажите правильный синтаксис тернарного оператора в Java

- Ⓐ логическое_условие ? выражение1 : выражение2;
- if логическое_условие ? выражение1 : выражение2;
- if логическое_условие then выражение1 else выражение2;
- do ... while

23. В каких случаях применяют оператор switch?

- Ⓐ к известному числу возможных ситуаций выбора
- применяют, когда число ситуаций выбора не известно
- когда происходит дублирование значений выбора
- данный оператор в программировании не используется

24. Дан код.

```
public class Main {
    static int firstNum = 10;
    public static void main(String[] args) {
        String res = Integer.toHexString(firstNum++);
        System.out.println("postfix: " + res);
    }
}
```

Какой результат будет выведен после компиляции?

- Ⓐ postfix: a
- postfix: b
- postfix: 10
- postfix: 11

25. Дан код.

```
public class Main {  
    static int firstNum = 10;  
    public static void main(String[] args) {  
        String res = Integer.toHexString(++firstNum);  
        System.out.println("postfix: " + res);  
    }  
}
```

Какой результат будет выведен после компиляции?

- ☐ postfix: a
- ☒ postfix: b
- ☐ postfix: 10
- ☐ postfix: 11

26. Дан код.

```
public class Main {  
    static int firstNum = 10;  
    public static void main(String[] args) {  
        String res = Integer.toHexString(firstNum--);  
        System.out.println("postfix: " + res);  
    }  
}
```

Какой результат будет выведен после компиляции?

- ☒ postfix: a
- ☐ postfix: b
- ☐ postfix: 10
- ☐ postfix: 11

27. Дан код.

```
public class Main {  
    static int firstNum = 10;  
    public static void main(String[] args) {  
        String res = Integer.toHexString(--firstNum);  
        System.out.println("postfix: " + res);  
    }  
}
```

Какой результат будет выведен после компиляции?

- ☐ postfix: a
- ☐ postfix: b
- ☐ postfix: 10
- ☐ postfix: 11
- ☐ postfix: 9

28. Дан код.

```
int a= 10;  
int b= 10;  
String res = Integer.toHexString(a&b);  
System.out.println(res);
```

Какой результат будет выведен после компиляции

- ☒ a
- ☐ 1100

- 1011
- 10

29. Дан код.

```
int a= 10;  
int b= 11;  
String res = Integer.toHexString(a|b);  
System.out.println(res);
```

Какой результат будет выведен после компиляции

- ☒ b
- a
- 1001
- 1011
- 9

30. Дан код.

```
int a= 10;  
int b= 11;  
String res = Integer.toHexString(a^b);  
System.out.println(res);
```

Какой результат будет выведен после компиляции

- ☒ 1
- a
- 1001
- 1011
- b

31. Дан код.

```
int a= 10;  
String res = Integer.toHexString(a<<1);  
System.out.println(res);
```

Какой результат будет выведен после компиляции

- b
- a
- 10
- 20
- ☒ 14

32. К стилям (парадигмам) программирования относят:

- ☒ императивный
- ☒ функциональный
- ☐ итерационный
- ☐ линейный

33. К стилям (парадигмам) программирования относят:

- ☒ объектно-ориентированный
- ☒ процедурный (императивный)
- ☐ ветвящийся
- ☐ многопоточный

34. Основным преимуществом объектно-ориентированного программирования перед императивным является

- ☒ сокращение количества межмодульных вызовов и уменьшение объемов информации, передаваемой между модулями
- ☐ отказ от указателей
- ☐ использование специальных библиотек
- ☐ повышенные требования к аппаратному обеспечению

35. Статический аспект объектно-ориентированной декомпозиции программной системы предполагает наличие следующих отношений между классами

- ☒ ассоциация
- ☒ зависимость
- ☐ сцепление
- ☐ делегирование

36. К принципам ООП относят

- ☒ иерархия
- ☒ инкапсуляция
- ☐ динамическое распределение памяти
- ☐ переопределение методов

37. К принципам ООП относят

- ☒ полиморфизм
- ☒ инкапсуляция
- ☐ статическое распределение памяти
- ☐ перегрузка методов

38. К принципам ООП относят

- ☒ иерархия
- ☒ абстрагирование
- ☐ применение интерфейсов
- ☐ применение статических членов класса

39. К принципам ООП относят

- ☒ модульность
- ☒ иерархия
- ☐ динамическое распределение памяти
- ☐ приведение типов

40. К принципам ООП относят

- ☒ типизация
- ☒ параллелизм
- ☐ статическое распределение памяти
- ☐ использование указателей

Краткое описание и регламент выполнения

К тестам допускаются все обучающиеся.

По результатам итогового теста обучающийся может набрать максимально 100 баллов.

Комплект заданий для итогового теста

ОМ закрытого типа

Задание 1

Выберите один правильный вариант ответа.

Дан код. Какое действие описывает код?

```
import java.io.File;
public class Program {
    public static void main(String[] args) {
        File dir = new File("C://SomeDir");
        if(dir.isDirectory()){
            for(File item : dir.listFiles()){
                if(item.isDirectory()){
                    System.out.println(item.getName() + " \t folder");
                }
            }
        } else{
            System.out.println(item.getName() + "\t file");
        }
    }
}
```

- a. Запись текста в файл из определенного каталога
- b. Чтение текста из файла внутри каталога
- c. Получение всех подкаталогов и файлов в определенном каталоге
- d. Получение всех файлов в каталоге

Правильный ответ: c

Задание 2

Выберите один правильный вариант ответа.

Дан фрагмент кода. Каким будет результат?

```
private static String fileName = "C://blog/a.txt";
public static void main(String[] args)
    throws FileNotFoundException {
    FileWorker.delete(fileName);
}
```

- a. Создание файла
- b. Определение объекта для каталога
- c. Переименование файла
- d. Удаление файла

Правильный ответ: d

Задание 3

Выберите несколько правильных вариантов ответа.

Дан код. Какое действие описывает код?

```
import java.io.*;
public class Program {
    public static void main(String[] args) {
        String text = "Hello world!";
        try(FileOutputStream fos=new FileOutputStream("C://SomeDir//notes.txt"))
        {
            byte[] buffer = text.getBytes();
            fos.write(buffer, 0, buffer.length);
        }
        catch(IOException ex){
            System.out.println(ex.getMessage());
        }
        System.out.println("The file has been written");
    }
}
```

- a. Запись строки в файл
- b. Чтение строки из файла
- c. Считывание в промежуточный буфер из массива символов
- d. Запись в промежуточный буфер из массива символов

Правильный ответ: а.

Задание 4

Выберите несколько правильных вариантов ответа.

Дан код. Выберите все правильные ответы, при которых код скомпилируется, если вставить их в строку 1: (Выберите 2 варианта ответа)

```
public class OverrideThrowsTest {
    public static void main(String[] args)
        // 1
    {
        A a = new A();
        a.method();
        A ab = new B();
        ab.method();
        B b = new B();
        b.method();
    }
}

class A {
    public void method() throws IOException {}
}

class B extends A {
    public void method() throws FileNotFoundException {}
}
```

- a. throws IOException
- b. throws Exception
- c. throw Exception
- d. throws FileNotFoundException

Правильный ответ: а, b

Задание 5

Выберите один правильный вариант ответа.

Дан код. Какой результат будет выведен на экран?

```
class A implements Cloneable{ //1
    public int i=10; }
class B extends A implements Cloneable{
    public int i=20;
    @Override
    public B clone() throws CloneNotSupportedException {
        B cloneA = (B) super.clone(); //2
        cloneA.i = 15;
        return cloneA; } }
public class MyClass {
    public static void main(String[] args) throws CloneNotSupportedException {
        B b = new B();
        A a = b.clone(); //3
        System.out.println(a.i); } }
```

- a. 15

- в. Ошибка компиляции в строке 2
- с. 10
- д. Ошибка выполнения в строке 2

Правильный ответ: с.

ОМ открытого типа

Задание 6

Дайте развернутый ответ.

Представьте реализацию алгоритма пузырьковой сортировки в Java-коде?

Правильный ответ:

```
3 public class Solution {
4     public static void main(String[] args) {
5         int[] testArr = new int[]{6, 3, 8, 2, 6, 9, 4, 11, 1};
6         bubbleSort( array: testArr);
7         for (int i : testArr) {
8             System.out.println( x: i);
9         }
10    }
11
12    public static void bubbleSort(int[] array) {
13        for(int i = array.length - 1; i > 1; i--) {
14            for (int j = 0; j < i; j++) { //
15                if (array[j] > array[j+1]) {
16                    int temp = array[j];
17                    array[j] = array[j+1];
18                    array[j+1] = temp;
19                }
20            }
21        }
22    }
23 }
```

Задание 7

Дайте развернутый ответ.

Представьте реализацию алгоритма сортировка методом выбора на java.

Правильный ответ:

```

2  public class Solution {
3      public static void main(String[] args) {
4          int[] testArr = new int[]{6, 3, 8, 2, 6, 9, 4, 11, 1};
5          sortBySelect(array: testArr);
6          for (int i : testArr) {
7              System.out.println(x:i);
8          }
9      }
10     public static void sortBySelect(int[] array) {
11
12         for (int i = 0; i < array.length-1; i++) { // внешний обобщенный цикл
13             int min = i;
14             for (int j = i + 1; j < array.length; j++) { // обобщенный цикл, но с учетом с сортированных
15                 if (array[j] < array[min]) {
16                     min = j;
17                 }
18             }
19             int temp = array[i]; // вставка отсортированного числа, в положенную ему ячейку
20             array[i] = array[min];
21             array[min] = temp;
22         }
23     }
24 }

```

Задание 8

Дайте развернутый ответ.

Что такое паттерны проектирования?

Правильный ответ:

Шаблон проектирования или паттерн (англ. design pattern) в разработке программного обеспечения — повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста.

Задание 9

Дайте развернутый ответ.

Дайте определение понятию паттерн «фабричный метод»?

Правильный ответ:

Фабричный метод — это порождающий паттерн проектирования, который определяет общий интерфейс для создания объектов в суперклассе, позволяя подклассам изменять тип создаваемых объектов

Задание 10

Дайте развернутый ответ.

Дайте определение понятию шаблону проектирования синглтон?

Синглтон - это порождающий паттерн проектирования, который гарантирует, что у класса есть только один экземпляр, и предоставляет к нему глобальную точку доступа.

7.2.2. Пример практической работы

Практическая работа 1. Основные утилиты Java

Цель работы: ознакомиться с основными утилитами Java, научиться компилировать, выполнять программы, используя утилиты `javac`, `java`, научиться создавать, модифицировать архивы Java, запускать программы из `jar`-файлов.

Задание:

1. Создайте текстовый файл `Test.txt` и запишите в него следующее:

```
public class Test {  
    public static void main(String[] args) {  
        System.out.println("Hello from Test!");  
    }  
}
```

2. Переименуйте файл в `Test.java` и запустите его на выполнение, запишите в отчет команды и приведите скриншот выполнения

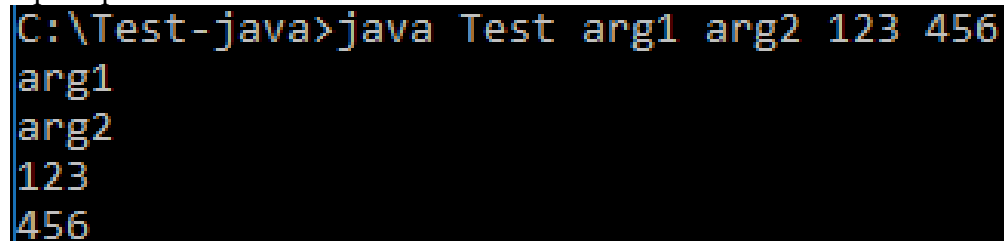
Решение: `java Test.java`

3. Скомпилируйте файл, запишите в отчет команды и приведите скриншот выполнения

Решение: `javac Test.java`

4. Перепишите код в файле таким образом, чтобы запускаемая программа выводила передаваемые в неё аргументы, запишите в отчет команды и приведите скриншот выполнения

Пример:



```
C:\Test-java>java Test arg1 arg2 123 456  
arg1  
arg2  
123  
456
```

Решение: в файле вместо `sout` пишем цикл, где обращаемся к элементам массива `args` через индекс и выводим их на консоль через `sout`

5. Создайте две копии файла `Test`. Переименуйте полученные файлы соответственно `Test1`, `Test2`, `Test3`. При запуске каждого из файлов, на консоль должно выводиться соответствующее сообщение:

Hello from Test1, Hello from Test2, ...

В текущей директории создайте каталог `bin`.

Скомпилируйте полученные файлы `java` таким образом, чтобы результат компиляции - файлы с расширением `class`, оказались в папке `bin`. Выполните это действие за одну команду, запишите в отчет команду и приведите скриншот выполнения

Решение: `javac -d bin *.java`

Возможные проблемы:

- Имя файла должно соответствовать имени класса
- Возможны проблемы с доступом, запустите `cmd` от администратора
- Не забудьте создать папку `bin` вручную, команда не создаёт папки автоматически

6. Удалите файлы с расширением `class` таким образом, чтобы они остались только в каталоге `bin`. Перейдите из текущей рабочей директории в любую другую внешнюю директорию. Из новой директории запустите на выполнение любой из файлов `Test`, запишите в отчет команды и приведите скриншот выполнения

Решение: `java -classpath Test-java\bin Test1`

7. Упакуйте полученные файлы class в архив jar таким образом, чтобы при запуске архива на консоль выводилось сообщение из программы Test2 (Hello from Test2), запишите в отчет команды и приведите скриншот выполнения

Решение: jar cfe app.jar Test2 *.class (находясь в каталоге bin)
java -jar app.jar

8. Из этого же архива (не создавая новый) запустите на выполнение программу Test1, запишите в отчет команды и приведите скриншот выполнения

Решение: java -cp app.jar Test1

9. В каталоге bin создайте 3 новых каталога – t1, t2, t3. В каждый из них поместите соответственно файлы Test. Затем упакуйте каталог bin в архив jar с сохранением структуры каталогов, запишите в отчет команды и приведите скриншот выполнения.

10. Заполните отчет, включающий следующую структуру:

- Название работы
- Цель работы
- Формулировка задания
- Исходный код с комментариями
- Вывод программы (если есть консольный вывод)
- Объяснение кода (что делает программа, какие интерфейсы использованы и почему)
- Выводы

Требования к оформлению

Отчет должен содержать подробное описание (включая иллюстрации). Отчёт по практическому занятию выполняется на страницах формата А4 в электронном виде.

При оформлении отчёта используется сквозная нумерация страниц, считая титульный лист первой страницей. Номер страницы на титульном листе не ставится. Номера страницы ставятся по центру сверху.

При оформлении отчёта соблюдать следующие требования:

- Для заголовков: полужирный шрифт, 14 пт, центрированный.
- Для основного текста: нежирный шрифт, 14 пт, выравнивание по ширине.
- Во всех случаях тип шрифта – Times New Roman, отступ абзаца 1.25 см, полуторный междустрочный интервал.
- Поля: левое – 2 см, правое, верхнее и нижнее – 1 см.

Процедура оценивания

Оценка выполненной практической работы проводится по следующим критериям:

1. Наличие всей существенной информации по работе
2. Точность и полнота предоставляемых сведений
3. Непротиворечивость приводимой информации
4. Правильность интерпретаций и выводов, которые сделаны по результатам работы
5. Степень достижения обучающимся поставленной цели
6. Обоснованность применяемого решения
7. Грамотность (содержательная) используемых формулировок

Критерии оценки за отчеты по практическим работам:

Формы текущего контроля	Критерии и нормы оценки
Отчеты по практическим работам 1 - 3	10 баллов – задание выполнено в полном объёме без замечаний 9 баллов – задание выполнено в полном объёме, присутствуют замечания 8 баллов – задание выполнено в объёме 70%, замечаний нет.

	6 баллов – задание выполнено в объёме 70%, присутствуют замечания. 5 баллов – задание выполнено в объёме 50%, замечаний нет. 4 балла – задание выполнено в объёме 50%, присутствуют замечания. 3 балла - задание выполнено в объёме менее 50%, замечаний нет. 1 балл – задание выполнено в объёме менее 50%, присутствуют замечания. 0 баллов – задание не выполнено.
Отчеты по практическим работам 4 - 8	12 баллов – задание выполнено в полном объёме без замечаний 10 баллов – задание выполнено в полном объёме, присутствуют замечания 9 баллов – задание выполнено в объёме 70%, замечаний нет. 7 баллов – задание выполнено в объёме 70%, присутствуют замечания. 5 баллов – задание выполнено в объёме 50%, замечаний нет. 4 балла – задание выполнено в объёме 50%, присутствуют замечания. 3 балла - задание выполнено в объёме менее 50%, замечаний нет. 1 балл – задание выполнено в объёме менее 50%, присутствуют замечания. 0 баллов – задание не выполнено.

7.3. Оценочные средства для Промежуточной аттестации по итогам освоения дисциплины

7.3.1. Вопросы к промежуточной аттестации

Семестр 4

№ п/п	Вопросы к экзамену
1	Какие основные утилиты входят в состав JDK? Какова их роль в процессе разработки на Java?
2	Как работает компилятор javac? Каковы его основные параметры и возможности?
3	Что такое JAR-файлы? Как их создать, запаковать и выполнить с помощью jar?
4	Как использовать java для запуска Java-приложений? Какие ключевые параметры можно передавать?
5	Что такое javadoc и как генерировать документацию для Java-классов?
6	Как работает jdb и для чего он используется? Какие основные команды он поддерживает?
7	Как использовать jconsole и jvisualvm для мониторинга работы JVM?
8	Что такое javap и как он помогает анализировать байт-код?
9	Как jstat и jmap помогают диагностировать проблемы с производительностью в Java?
10	Как работает keytool и jarsigner? В каких ситуациях эти утилиты применяются?
11	Каковы основные принципы типизации в Java? Чем примитивные типы данных отличаются от ссылочных?
12	Что такое идентификаторы в Java? Какие правила их именования следует соблюдать?
13	Как работают операторы сравнения и логические операторы в Java? В чем их отличия?
14	Какие существуют управляющие конструкции в Java (if-else, switch, while, for)? В каких случаях их лучше использовать?
15	Каковы особенности использования тернарного оператора в Java? В каких ситуациях он полезен?
16	Как работают массивы в Java? Чем они отличаются от коллекций?

№ п/п	Вопросы к экзамену
17	Что такое перегрузка методов (method overloading)? Как Java определяет, какой метод вызвать?
18	Как работают varargs (аргументы переменной длины) в Java? Какие ограничения у них есть?
19	Как работает механизм static в Java? В каких случаях следует использовать статические методы и переменные?
20	Как в Java объявляются и используются константы? Чем final отличается от static final?
21	Каковы ключевые принципы ООП (инкапсуляция, наследование, полиморфизм, абстракция) и как они реализованы в Java?
22	Что такое классы и объекты в Java? Как они взаимодействуют друг с другом?
23	Как работает механизм наследования в Java? Какие ключевые слова используются для его реализации?
24	Какую роль играет ключевое слово super в Java? В каких случаях оно необходимо?
25	Что такое абстрактные классы? В чем их отличие от обычных классов? В каких ситуациях они применяются?
26	Что такое конструкторы в Java? Как они перегружаются? Чем отличается конструктор по умолчанию от пользовательского?
27	Как работает модификатор доступа protected? В каких случаях его использование оправдано?
28	Как реализовать в Java отношение HAS-A (композиция и агрегация)? Как это помогает в проектировании классов?
29	Что такое полиморфизм в Java? Какие существуют его разновидности и как они применяются на практике?
30	Как работает аннотация @Override? Почему ее использование является хорошей практикой?
31	Что такое интерфейсы в Java? Как они позволяют реализовать множественное наследование?
32	Каковы ключевые отличия между интерфейсами и абстрактными классами? В каких случаях лучше использовать каждый из них?
33	Что такое методы по умолчанию (default methods) в интерфейсах Java? В чем их преимущество?
34	Как интерфейсы могут наследовать друг друга? В каких случаях это полезно?
35	Что такое функциональные интерфейсы? Как они связаны с лямбда-выражениями?
36	Как в Java работает класс String? Почему строки являются неизменяемыми (immutable)?
37	Как работают классы StringBuilder и StringBuffer? В чем разница между ними?
38	Какие методы класса String используются для поиска и замены подстрок? Каковы их особенности?
39	Как происходит сравнение строк в Java? В чем разница между == и equals()?
40	Как эффективно объединять строки в Java, избегая проблем с производительностью?
41	Как работать с датами в Java 8 (java.time API)? Почему java.util.Date и Calendar считаются устаревшими?
42	Как использовать LocalDate, LocalTime и LocalDateTime? В чем их основные различия?
43	Как форматировать и парсить дату в Java? Какие классы для этого используются?

№ п/п	Вопросы к экзамену
44	Как работать с часовыми поясами в Java? Как ZonedDateTime помогает учитывать временные зоны?
45	Как вычислять разницу между датами? Какие классы помогают в этом (Duration, Period)?
46	Что такое обобщенные типы (Generics) в Java? Как они помогают писать более универсальный код?
47	Как создать обобщенный класс и обобщенный метод в Java?
48	Что такое ограниченные (bounded) типы в Generics? Как использовать extends и super в Generics?
49	Как работают wildcard (?) в Generics? В каких ситуациях их использование оправдано?
50	Как Java реализует механизмы стирания типов (type erasure)? Как это влияет на работу с Generics?
51	Какие интерфейсы входят в Java Collections Framework? В чем их ключевые отличия?
52	Как работает интерфейс List? В чем разница между ArrayList и LinkedList?
53	Как работает интерфейс Set? Как HashSet и TreeSet обеспечивают уникальность элементов?
54	Что такое Map? Как устроены HashMap и TreeMap? Какие задачи они решают?
55	Как правильно выбирать коллекцию для конкретной задачи? Какие факторы влияют на выбор?
56	Как происходит сортировка элементов в коллекциях? В чем разница между Comparable и Comparator?
57	Как работает Iterator? Какие существуют способы обхода коллекций?
58	Как устроены потокобезопасные коллекции (ConcurrentHashMap, CopyOnWriteArrayList)?
59	Что такое неизменяемые (immutable) коллекции? Как их создать?
60	Как PriorityQueue реализует структуру очереди с приоритетами?

7.3.2. Критерии и нормы оценки

Семестр	Форма проведения промежуточной аттестации	Критерии и нормы оценки	
	Экзамен (по накопительному рейтингу)	отлично	рейтинговый балл 85-100
		хорошо	рейтинговый балл 70-84
		удовлетворительно	рейтинговый балл 55-69
		неудовлетворительно	рейтинговый балл 0-54

8. Учебно-методическое и информационное обеспечение дисциплины

8.1. Обязательная литература

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно-методическое пособие, Практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
1	Монахов, В. В.	Язык Программирования Java и среда NetBeans	Учебное пособие	2021	ЭБС "IPRbooks"
2	Мухаметзянов, Р. Р.	Основы Программирования на Java	Учебное пособие	2017	ЭБС "IPRbooks"
3	Вязовик, Н. А.	Программирование на Java	Учебное пособие	2021	ЭБС «IPRBooks»

8.2. Дополнительная литература

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно-методическое пособие, Практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
1	Гаряева В. В	Решение задач с использованием пакетов Прикладных Программ	Учебное пособие	2017	ЭБС "IPRbooks"
2	Лоскутов В. И., Коробова И.Л.	Разработка информационных систем для Windows Store		2016	ЭБС "IPRbooks"
3	Стешин А. И.	Информационные системы в организации	Учебное пособие	2019	ЭБС "IPRbooks"
4	Голицына О. Л., Максимов Н.В., Попов И.И.	Информационные системы	Учебное пособие	2018	ЭБС "Znanium.com"

8.3. Перечень Профессиональных баз данных и информационных справочных систем

№ пп	Наименование	Ссылка
1	Springer Nature (Полнотекстовая коллекция журналов)	https://www.springernature.com/gp/products
2	Springer eBooks (Полнотекстовая коллекция электронных книг издательства Springer Nature)	https://link.springer.com/
3	«Кодекс»	https://kodeks.ru/
4	Техэксперт	https://cntd.ru/

8.4. Перечень Программного обеспечения

№ п/п	Наименование ПО	Реквизиты договора (дата, номер, срок действия)
1	Eclipse IDE 4.x (Eclipse Foundation)	Лицензия Eclipse Public License (EPL)
2	Apache NetBeans IDE 8.x	Лицензия Apache License 2.0
3	Code::Blocks 16.x (The Code::Blocks Team)	Лицензия GNU GPLv3
4	PostgreSQL 16	Лицензия PostgreSQL License (аналогичная BSD)
5	Oracle OpenJDK 21	Лицензия GNU GPLv2 с Classpath Exception

8.5. Описание материально-технической базы, необходимой для осуществления образовательного Процесса по дисциплине

№ п/п	Наименование оборудованных учебных кабинетов, лабораторий, мастерских и др. объектов для Проведения Практических и лабораторных занятий, помещений для самостоятельной работы обучающихся (номер аудитории)	Перечень основного оборудования
1	Компьютерный класс. Учебная аудитория для проведения занятий лекционного типа. Учебная аудитория для проведения занятий семинарского типа. Учебная аудитория для проведения лабораторных работ. Учебная аудитория для курсового проектирования (выполнения курсовых работ). Учебная аудитория для проведения групповых и индивидуальных консультаций Учебная аудитория для проведения занятий текущего контроля и промежуточной аттестации. (УЛК-407).	Компьютер (монитор Samsung Sync Master 943n 19", системный блок Intel (R) Core 2 Quad 2,40 GHz 1 Gb), столы лабораторные, стулья, доска 3-х секционная (меловая), стол преподавательский.

№ п/п	Наименование оборудованных учебных кабинетов, лабораторий, мастерских и др. объектов для Проведения Практических и лабораторных занятий, помещений для самостоятельной работы обучающихся (номер аудитории)	Перечень основного оборудования
2	Помещение для самостоятельной работы обучающихся (УЛК-105).	Стол, стулья, стеллажи (в т.ч. выставочные) с книгами, персональные компьютеры, мобильные рабочие места.
3	Помещение для самостоятельной работы обучающихся (УЛК-406).	Стол компьютерный, стулья, микрокомпьютеры raspberry pi 32 bit.