

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) 2

(наименование дисциплины)

по направлению подготовки
09.03.03 Прикладная информатика

направленность (профиль)
Автоматизация бизнес-процессов и проектирование ИТ-решений

Форма обучения: заочная

Год набора: 2024

Общая трудоемкость: 5 ЗЕ

Распределение часов дисциплины по семестрам

Семестр	6	Итого
Форма контроля	экзамен	
Вид занятий		
Лекции	4	4
Лабораторные		
Практические	-	-
Руководство: курсовые работы (проекты) / РГР		
Промежуточная аттестация	0.35	0.35
Контактная работа	4.35	4.35
Самостоятельная работа	167	167
Контроль	8.65	8.65
Итого	180	180

Рабочую программу составил(и):

доцент института цифровых технологий, к.п.н., Ерофеева Е.А.

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

(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 2031 г.

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, Алгоритмы и структуры данных, Практикум по программированию на Java (Джава) 1.

Дисциплины и практики, для которых освоение данной дисциплины необходимо как предшествующее: – Производственная практика (технологическая (проектно-технологическая) практика), 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 ведение базы данных и поддержку информационного обеспечения решения прикладных задач	ПК-5.1 Знает технологии разработки и ведения баз данных	Знать:технологии разработки и ведения баз данных Уметь:применять технологии разработки и ведения баз данных Владеть:навыками применения технологий разработки и ведения баз данных
	ПК-5.2 Умеет проектировать и разрабатывать базы данных, использовать их для поддержки информационного обеспечения решения прикладных задач	Знать:принципы проектирования и разработки базы данных Уметь:проектировать и разрабатывать базы данных, использовать их для поддержки информационного обеспечения решения прикладных задач Владеть:навыками проектирования и разработки базы данных, использования их для поддержки информационного обеспечения решения прикладных задач
	ПК-5.3 Владеет навыками ведения базы данных и поддержки информационного обеспечения решения прикладных задач	Знать:принципы эксплуатации баз данных, поддержки информационного обеспечения решения прикладных задач Уметь:осуществлять поддержку информационного обеспечения решения прикладных задач

		Владеть:навыками эксплуатации баз данных, поддержки информационного обеспечения решения прикладных задач
--	--	--

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. Библиотеки ввода вывода	Лек 1	Java I/O и java NIO	6	2		-	
	Ср	Java I/O и java NIO	6	11	6	-	Отчет по практической работе 1
2. Обработка исключений	Ср	Обработка исключений	6	11		-	
	Ср	Обработка исключений	6	11	6		Отчет по практической работе 2
3. Лямбда-выражения	Ср	Лямбда выражения	6	11			
	Ср	Лямбда выражения	6	11	8		Отчет по практической работе 3
4. Регулярные выражения	Ср	Регулярные выражения	6	11			
	Ср	Регулярные выражения	6	11	8		Отчет по практической работе 4
5. Библиотека JDBC	Лек 5	JAVA DATABASES	6	2			
	Ср	JAVA DATABASES	6	11	8		Отчет по практической работе 5
6. Работа с XML	Ср	Работа с XML на Java	6	11			
	Ср	Работа с XML на Java	6	11	8		Отчет по практической работе 6
7. Паттерны проектирования Java	Ср	Паттерны проектирования Java	6	11			
	Ср	Паттерны проектирования Java	6	11	8		Отчет по практической работе 7
8. Многопоточность и синхронизация в Java	Ср	Многопоточность и синхронизация	6	11		-	
	Ср	Многопоточность и синхронизация	6	11	8	-	Отчет по практической работе 8
	Ср	Работа с теоретическим материалом	6	12			
	ПА	Промежуточная аттестация	6	0,35		-	
	Контроль	Экзамен	6	8,65	40		Итоговый тест
Итого:				180			

Схема расчета итогового балла

Текущий рейтинг + Результат итогового теста

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
6	ПК-5	<i>Вопросы к экзамену Отчеты по практическим занятиям</i>

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

7.2.1. Примеры тестовых заданий

(наименование оценочного средства)

1. Какой класс используется для чтения текстовых данных из файла в Java?
A) FileInputStream
B) FileReader
C) BufferedInputStream
D) Scanner
Правильный ответ: B) FileReader
2. Какой класс позволяет буферизовать ввод и повысить производительность чтения?
A) BufferedReader
B) InputStreamReader
C) DataInputStream
D) RandomAccessFile
Правильный ответ: A) BufferedReader
3. Как правильно закрыть FileWriter в Java?
A) fw.close();
B) fw.shutdown();
C) fw.finalize();
D) fw.dispose();
Правильный ответ: A) fw.close();
4. Какой интерфейс используется для работы с файловыми путями в Java NIO?
A) Path
B) FilePath
C) FileLocation
D) FileSystem
Правильный ответ: A) Path
5. Какой метод Files позволяет записать строку в файл?
A) Files.write()
B) Files.save()
C) Files.store()
D) Files.append()

Правильный ответ: A) Files.write()

6. Какой метод `BufferedReader` считывает одну строку?

- A) `readLine()`
- B) `nextLine()`
- C) `read()`
- D) `readString()`

Правильный ответ: A) `readLine()`

7. Какой класс используется для случайного доступа к файлу?

- A) `RandomAccessFile`
- B) `FileReader`
- C) `DataInputStream`
- D) `BufferedReader`

Правильный ответ: A) `RandomAccessFile`

8. Какой оператор лучше всего использовать для корректного закрытия ресурсов?

- A) `try-catch`
- B) `finally`
- C) `try-with-resources`
- D) `synchronized`

○ **Правильный ответ: C) `try-with-resources`**

○

9. В чем отличие `FileInputStream` от `BufferedInputStream`?

- A) `BufferedInputStream` использует буфер для ускорения чтения
- B) `FileInputStream` работает только с текстовыми файлами
- C) `BufferedInputStream` может читать только строки
- D) `FileInputStream` не может работать с файлами

Правильный ответ: A) `BufferedInputStream` использует буфер для ускорения чтения

10. Какой метод `Files` позволяет скопировать файл?

- A) `Files.copy()`
- B) `Files duplicate()`
- C) `Files.clone()`
- D) `Files.replicate()`

Правильный ответ: A) `Files.copy()`

11. Какой тип исключения является непроверяемым (unchecked)?

- A) `IOException`
- B) `NumberFormatException`
- C) `SQLException`
- D) `FileNotFoundException`

Правильный ответ: B) `NumberFormatException`

12. Какой блок кода является обязательным в конструкции `try-catch-finally`?

- A) `try`
- B) `catch`
- C) `finally`
- D) Все три

Правильный ответ: A) `try`

13. Как можно создать собственное исключение в Java?

- A) Унаследовавшись от Exception
- B) Унаследовавшись от Throwable
- C) Реализовав интерфейс Error
- D) Используя throw new Error()

Правильный ответ: A) Унаследовавшись от Exception

14. Какое ключевое слово используется для выбрасывания исключения?

- A) throw
- B) throws
- C) new
- D) exception

Правильный ответ: A) throw

15. Что произойдет, если в блоке finally возникнет исключение?

- A) Исключение перезапишет предыдущее
- B) Будет проигнорировано
- C) Программа завершится аварийно
- D) Выполнится блок catch

Правильный ответ: A) Исключение перезапишет предыдущее

16. Можно ли обработать OutOfMemoryError?

- A) Нет
- B) Да, используя try-catch
- C) Да, но только с блоком finally
- D) Только если использовать многопоточность

Правильный ответ: B) Да, используя try-catch

17. Что означает throws в заголовке метода?

- A) Метод может выбросить исключение
- B) Метод перехватывает исключение
- C) Метод игнорирует исключение
- D) Метод не выбрасывает исключений

Правильный ответ: A) Метод может выбросить исключение

18. Можно ли выбросить несколько исключений в одном throw?

- A) Да
- B) Нет
- C) Только если они являются подклассами RuntimeException
- D) Только в Java 17+

Правильный ответ: B) Нет

19. Что делает try-with-resources?

- A) Автоматически закрывает ресурсы
- B) Перехватывает все исключения
- C) Заменяет try-catch
- D) Позволяет работать без catch

Правильный ответ: A) Автоматически закрывает ресурсы

20. Какой оператор используется для перехвата всех исключений?

- A) catch(Exception e)
- B) catch(Throwable t)
- C) catch(*)

D) catch(RuntimeException e)

Правильный ответ: B) catch(Throwable t)

21. Какой интерфейс необходим для работы с лямбда-выражениями?

A) FunctionalInterface

B) Runnable

C) Callable

D) Serializable

Правильный ответ: A) FunctionalInterface

22. Как объявить лямбда-выражение, принимающее два числа и возвращающее их сумму?

A) (a, b) -> a + b

B) (int a, int b) { return a + b; }

C) { a + b }

D) sum(a, b) -> a + b

Правильный ответ: A) (a, b) -> a + b

23. Какой пакет содержит встроенные функциональные интерфейсы Java?

A) java.util.function

B) java.lang.functional

C) java.functional

D) java.util.stream

Правильный ответ: A) java.util.function

24. Какой функциональный интерфейс используется для представления операции без аргументов, но возвращающей значение?

A) Supplier

B) Consumer

C) Function

D) BiFunction

Правильный ответ: A) Supplier

25. Можно ли передавать лямбда-выражение как аргумент метода?

A) Да

B) Нет

C) Только если оно возвращает значение

D) Только если оно не принимает параметры

Правильный ответ: A) Да

26. Какой интерфейс представляет операцию, принимающую два аргумента и возвращающую значение?

A) BiFunction

B) Function

C) Consumer

D) Predicate

Правильный ответ: A) BiFunction

27. Что возвращает лямбда-выражение () -> 42?

A) Значение 42

B) Ошибку компиляции

C) null

D) Ничего

Правильный ответ: A) Значение 42

28. Какой метод функционального интерфейса Consumer<T> необходимо реализовать?

- A) accept()
- B) apply()
- C) test()
- D) execute()

Правильный ответ: A) accept()

29. Можно ли использовать переменные из окружающей области видимости в лямбда-выражении?

- A) Да, но только final или effectively final
- B) Нет
- C) Только в статических методах
- D) Только если они объявлены в классе

Правильный ответ: A) Да, но только final или effectively final

30. Какой функциональный интерфейс возвращает boolean?

- A) Predicate
- B) Consumer
- C) Function
- D) BiConsumer

Правильный ответ: A) Predicate

31. Какой класс используется для работы с регулярными выражениями в Java?

- A) Pattern
- B) Matcher
- C) Regex
- D) StringRegex

Правильный ответ: A) Pattern

32. Какой метод проверяет соответствие строки регулярному выражению?

- A) matches()
- B) check()
- C) validate()
- D) compare()

Правильный ответ: A) matches()

33. Что делает \d в регулярных выражениях?

- A) Соответствует одной цифре
- B) Соответствует одной букве
- C) Соответствует символу пробела
- D) Соответствует любому символу

Правильный ответ: A) Соответствует одной цифре

34. Какой метод в Matcher используется для поиска соответствий в строке?

- A) find()
- B) search()
- C) locate()
- D) seek()

Правильный ответ: A) find()

35. Какой флаг передается в Pattern.compile() для игнорирования регистра?

- A) Pattern.CASE_INSENSITIVE
- B) Pattern.IGNORE_CASE
- C) Pattern.NO_CASE
- D) Pattern.LOWER_UPPER

Правильный ответ: A) Pattern.CASE_INSENSITIVE

36. Как записать регулярное выражение для поиска слова "Java" в начале строки?

- A) ^Java
- B) Java\$
- C) \bJava\b
- D) Java+

Правильный ответ: A) ^Java

37. Какое регулярное выражение соответствует email-адресу?

- A) [a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}
- B) .*@.*\..*
- C) [a-zA-Z0-9]+@.+
- D) email:[a-z]+@[a-z]+\.[a-z]+

Правильный ответ: A) [a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}

38. Какой метод Matcher позволяет заменить найденные подстроки?

- A) replaceAll()
- B) set()
- C) change()
- D) substitute()

Правильный ответ: A) replaceAll()

39. Как записать регулярное выражение, соответствующее 3 цифрам подряд?

- A) \d{3}
- B) \d*
- C) \d+
- D) [0-9]{3,}

Правильный ответ: A) \d{3}

40. Как обозначить пробельный символ в регулярных выражениях?

- A) \s
- B) \w
- C) \p
- D) _

Правильный ответ: A) \s

41. Какой интерфейс используется для установления соединения с базой данных в JDBC?

- A) Connection
- B) Statement
- C) Database
- D) DriverManager

Правильный ответ: A) Connection

42. Какой метод используется для выполнения SQL-запроса SELECT?

- A) executeQuery()
- B) executeUpdate()

- C) runQuery()
- D) queryExecute()

Правильный ответ: A) executeQuery()

43. Какой метод Connection закрывает соединение с базой данных?

- A) close()
- B) disconnect()
- C) shutdown()
- D) terminate()

• **Правильный ответ: A) close()**

•

44. Что выведет следующий код, если в таблице users есть запись с id = 1 и name = "John"?

```
Connection conn = DriverManager.getConnection(url, user, pass);
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("SELECT name FROM users WHERE id = 1");
rs.next();
System.out.println(rs.getString("name"));
conn.close();
```

- A) John
- B) Ошибка компиляции
- C) Исключение
- D) null

Правильный ответ: A) John

45. Что делает метод executeUpdate()?

- A) Выполняет SQL-запросы INSERT, UPDATE, DELETE
- B) Выполняет SQL-запросы SELECT
- C) Выполняет SQL-запросы JOIN
- D) Открывает соединение с БД

Правильный ответ: A) Выполняет SQL-запросы INSERT, UPDATE, DELETE

46. Какой JDBC-драйвер используется в современных версиях Java для работы с PostgreSQL?

- A) org.postgresql.Driver
- B) com.mysql.Driver
- C) jdbc.postgres.Driver
- D) postgresql.jdbc.Driver

Правильный ответ: A) org.postgresql.Driver

47. Чем PreparedStatement отличается от Statement?

- A) Позволяет использовать параметры
- B) Работает только с SELECT
- C) Использует JDBC 3.0
- D) Не поддерживает выполнение SQL-запросов

Правильный ответ: A) Позволяет использовать параметры

48. Что выведет следующий код, если таблица users пуста?

```
Connection conn = DriverManager.getConnection(url, user, pass);
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("SELECT * FROM users");
if (rs.next()) {
```

```
        System.out.println("Есть пользователи");  
    } else {  
        System.out.println("Пользователей нет");  
    }  
    conn.close();
```

- A) Есть пользователи
- B) Пользователей нет
- C) Ошибка компиляции
- D) Исключение

Правильный ответ: B) Пользователей нет

49. Что произойдет при попытке второй раз вызвать rs.next() в коде ниже?

```
ResultSet rs = stmt.executeQuery("SELECT name FROM users");  
rs.next();  
rs.next();
```

- A) Переход к следующей строке
- B) Исключение, если данных больше нет
- C) Ошибка компиляции
- D) Вернется null

Правильный ответ: B) Исключение, если данных больше нет

50. Какой метод используется для задания параметра в PreparedStatement?

- A) setString()
- B) addParameter()
- C) setParam()
- D) bindValue()

Правильный ответ: A) setString()

51. Какой API используется в Java для обработки XML?

- A) SAX
- B) DOM
- C) JAXB
- D) Все перечисленные

Правильный ответ: D) Все перечисленные

52. Какой класс используется для парсинга XML с помощью DOM?

- A) DocumentBuilder
- B) XMLParser
- C) XMLProcessor
- D) DOMParser

Правильный ответ: A) DocumentBuilder

53. Какой API является потоковым (обрабатывает XML построчно)?

- A) SAX
- B) DOM
- C) JAXB
- D) XSLT

Правильный ответ: A) SAX

54. Какой из методов **НЕ** относится к обработке XML в Java?

- A) XPath
- B) SAX
- C) Hibernate

D) JAXB

Правильный ответ: C) Hibernate

55. Какой из следующих вариантов является корректным XML-документом?

A)

```
<root>
  <tag>Value</tag>
</root>
```

B)

```
<root>
  <tag>Value</root>
```

C)

```
<root>
  <tag>Value<tag>
</root>
```

D)

```
<root tag="value">
</root>
```

Правильный ответ: A)

56. Какой интерфейс используется в JAXB для конвертации Java-объектов в XML?

A) Marshaller

B) Serializer

C) XMLConverter

D) XMLBinder

Правильный ответ: A) Marshaller

57. Что делает следующий код?

```
DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
DocumentBuilder builder = factory.newDocumentBuilder();
Document doc = builder.parse(new File("data.xml"));
```

A) Загружает XML-файл и строит DOM-модель

B) Записывает данные в XML-файл

C) Преобразует XML в JSON

D) Выполняет XSLT-преобразование

Правильный ответ: A) Загружает XML-файл и строит DOM-модель

58. Как можно получить значение тега <name>John</name> в DOM?

A) element.getTextContent();

B) element.getValue();

C) element.getText();

D) element.getData();

Правильный ответ: A) element.getTextContent();

59. Что делает XPath в Java?

A) Позволяет извлекать данные из XML по заданному пути

B) Компилирует XML-документ

C) Преобразует XML в JSON

D) Создает новую XML-структуру

Правильный ответ: A) Позволяет извлекать данные из XML по заданному пути

60. Как создать новый элемент в XML с помощью DOM?

- A) document.createElement("tag");
- B) document.newElement("tag");
- C) document.add("tag");
- D) document.makeElement("tag");

Правильный ответ: A) document.createElement("tag");

61. Какой из следующих паттернов относится к **порождающим (Creational)** паттернам?

- A) Singleton
- B) Observer
- C) Adapter
- D) Strategy

Правильный ответ: A) Singleton

62. Какой паттерн используется для создания **только одного экземпляра класса**?

- A) Singleton
- B) Factory Method
- C) Builder
- D) Prototype

Правильный ответ: A) Singleton

63. Какой метод обычно используется для получения экземпляра Singleton?

- A) getInstance()
- B) createInstance()
- C) newInstance()
- D) getObject()

Правильный ответ: A) getInstance()

64. Какой паттерн проектирования позволяет создавать объекты без указания их конкретного класса?

- A) Factory Method
- B) Singleton
- C) Decorator
- D) Observer

Правильный ответ: A) Factory Method

65. Какой паттерн проектирования позволяет **изменять поведение объектов во время выполнения**?

- A) Strategy
- B) Builder
- C) Factory Method
- D) Singleton

Правильный ответ: A) Strategy

66. Что выведет следующий код?

```
class Singleton {  
    private static Singleton instance = new Singleton();  
    private Singleton() {}  
    public static Singleton getInstance() {  
        return instance;  
    }  
}
```



```
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Singleton s1 = Singleton.getInstance();  
        Singleton s2 = Singleton.getInstance();  
        System.out.println(s1 == s2);  
    }  
}
```

- A) true
- B) false
- C) Ошибка компиляции
- D) Исключение

Правильный ответ: A) true

67. Какой паттерн используется для **добавления функциональности объекту без изменения его структуры?**

- A) Decorator
- B) Adapter
- C) Factory Method
- D) Singleton

Правильный ответ: A) Decorator

68. Какой паттерн можно использовать для **реализации подписки на события?**

- A) Observer
- B) Factory Method
- C) Singleton
- D) Builder

Правильный ответ: A) Observer

69. Какой паттерн помогает **изолировать высокий уровень логики от низкоуровневых деталей реализации?**

- A) Facade
- B) Strategy
- C) Prototype
- D) Adapter

Правильный ответ: A) Facade

70. Какой паттерн позволяет **объектам с несовместимыми интерфейсами работать вместе?**

- A) Adapter
- B) Factory Method
- C) Prototype
- D) Observer

Правильный ответ: A) Adapter

71. Какой интерфейс необходимо реализовать, чтобы создать поток в Java?

- A) Runnable
- B) Callable
- C) Thread
- D) Executor

Правильный ответ: A) Runnable

72. Какой метод используется для запуска нового потока?

- A) start()
- B) run()
- C) execute()
- D) call()

Правильный ответ: A) start()

73. Что произойдет, если вызвать `run()` вместо `start()` в потоке?

- A) Метод выполнится в текущем потоке
- B) Создастся новый поток
- C) Будет ошибка компиляции
- D) Будет выброшено исключение

Правильный ответ: A) Метод выполнится в текущем потоке

74. Какой оператор используется для предотвращения одновременного доступа к ресурсу несколькими потоками?

- A) synchronized
- B) volatile
- C) atomic
- D) lock

Правильный ответ: A) synchronized

75. Какой из следующих классов используется для управления потоками в `java.util.concurrent`?

- A) ExecutorService
- B) ThreadPool
- C) ThreadManager
- D) RunnablePool

Правильный ответ: A) ExecutorService

76. Что делает ключевое слово `volatile`?

- A) Гарантирует, что значение переменной всегда будет читаться из памяти
- B) Блокирует доступ к переменной другим потокам
- C) Создает копию переменной для каждого потока
- D) Используется для создания потоков

Правильный ответ: A) Гарантирует, что значение переменной всегда будет читаться из памяти

77. Что делает следующий код?

```
synchronized void myMethod() {  
    System.out.println("Executing synchronized method");  
}
```

- A) Обеспечивает, что метод выполняется только одним потоком одновременно
- B) Делает метод асинхронным
- C) Блокирует весь класс для всех потоков
- D) Ничего не делает

Правильный ответ: A) Обеспечивает, что метод выполняется только одним потоком одновременно

78. Какой метод потока заставляет текущий поток ожидать завершения другого потока?

- A) join()
- B) wait()
- C) sleep()

D) notify()

Правильный ответ: A) join()

79. Какой класс можно использовать для блокировки потоков вручную?

A) ReentrantLock

B) AtomicInteger

C) Semaphore

D) Executor

Правильный ответ: A) ReentrantLock

80. Что делает метод `sleep()` в классе `Thread`?

A) Приостанавливает выполнение потока на заданное время

B) Завершает поток

C) Усыпляет поток навсегда

D) Отменяет выполнение потока

Правильный ответ: A) Приостанавливает выполнение потока на заданное время

81. Какой из методов **НЕ** блокирует выполнение потока при работе с `InputStream`?

A) `read()`

B) `available()`

C) `readAllBytes()`

D) `skip()`

Правильный ответ: B) available()

82. Какой из классов **НЕ** является частью `java.io`?

A) `FileChannel`

B) `ObjectInputStream`

C) `PrintWriter`

D) `Path`

Правильный ответ: D) Path

83. Что произойдет, если закрыть `OutputStream`, но затем попытаться записать в него данные?

A) Данные будут записаны успешно

B) Будет выброшено `IOException`

C) Поток автоматически откроется снова

D) Программа зависнет

Правильный ответ: B) Будет выброшено IOException

84. Какой класс предпочтительнее использовать для высокопроизводительного копирования файла?

A) `FileReader` / `FileWriter`

B) `BufferedReader` / `BufferedWriter`

C) `FileChannel`

D) `ObjectInputStream` / `ObjectOutputStream`

Правильный ответ: C) FileChannel

85. В чем разница между `BufferedReader` и `InputStreamReader`?

A) `BufferedReader` работает с байтами, а `InputStreamReader` с символами

B) `BufferedReader` использует внутренний буфер для повышения производительности

- C) `InputStreamReader` быстрее, чем `BufferedReader`
- D) `BufferedReader` можно использовать только с файлами

Правильный ответ: B) `BufferedReader` использует внутренний буфер для повышения производительности

86. Какой метод позволяет записывать объект в файл?

- A) `writeObject()`
- B) `saveObject()`
- C) `storeObject()`
- D) `serialize()`

Правильный ответ: A) `writeObject()`

87. Какой флаг необходимо передать в `FileOutputStream`, чтобы включить режим добавления данных в файл?

- A) `true`
- B) `false`
- C) `"append"`
- D) `"a"`

Правильный ответ: A) `true`

88. Какой из классов является частью `java.nio.file` и удобен для работы с путями?

- A) `Path`
- B) `FileDescriptor`
- C) `PrintStream`
- D) `StreamReader`

Правильный ответ: A) `Path`

89. Почему важно закрывать потоки после использования?

- A) Для предотвращения утечек памяти и освобождения системных ресурсов
- B) Для ускорения выполнения программы
- C) Заккрытие потока повышает безопасность данных
- D) Потоки автоматически закрываются сборщиком мусора

Правильный ответ: A) Для предотвращения утечек памяти и освобождения системных ресурсов

90. Какой способ чтения файла в Java будет самым производительным?

- A) `BufferedReader + FileReader`
- B) `Files.readAllBytes(Path path)`
- C) `RandomAccessFile`
- D) `FileChannel + ByteBuffer`

Правильный ответ: D) `FileChannel + ByteBuffer`

91. Почему `try-catch-finally` может не выполнить блок `finally`?

- A) Если в блоке `catch` есть `System.exit(0);`
- B) Если исключение не поймано
- C) Если `finally` идет после `return`
- D) `finally` выполняется всегда

Правильный ответ: A) Если в блоке `catch` есть `System.exit(0);`

92. Что произойдет, если в блоке `finally` выбросить исключение?

- A) Исключение из `finally` заменит предыдущее исключение
- B) Исключение из `finally` будет проигнорировано

- C) Программа завершится без ошибок
- D) Первоначальное исключение будет подавлено

Правильный ответ: A) Исключение из `finally` заменит предыдущее исключение

93. Можно ли использовать `try` без `catch`?

- A) Да, если есть `finally`
- B) Нет, `catch` обязателен
- C) Только в `static` методах
- D) Только с `throws`

Правильный ответ: A) Да, если есть `finally`

94. Какие исключения являются `unchecked`?

- A) `NullPointerException` и `ArithmeticException`
- B) `IOException` и `SQLException`
- C) `FileNotFoundException` и `InterruptedException`
- D) `Throwable` и `Error`

Правильный ответ: A) `NullPointerException` и `ArithmeticException`

95. Можно ли переопределить метод, который бросает исключение, без указания `throws`?

- A) Да, но тогда метод не сможет выбросить исключение
- B) Нет, `throws` обязателен
- C) Только если исключение является `checked`
- D) Только если метод `private`

Правильный ответ: A) Да, но тогда метод не сможет выбросить исключение

96. Какой метод в `Throwable` позволяет получить полный стек вызовов?

- A) `printStackTrace()`
- B) `getStackTrace()`
- C) `toString()`
- D) `getMessage()`

Правильный ответ: A) `printStackTrace()`

97. Что делает ключевое слово `throws`?

- A) Объявляет, какие исключения метод может выбросить
- B) Перехватывает исключение
- C) Генерирует исключение
- D) Подавляет исключение

Правильный ответ: A) Объявляет, какие исключения метод может выбросить

98. Что произойдет, если в блоке `catch` выбросить другое исключение?

- A) Первоначальное исключение будет потеряно
- B) Оба исключения будут обработаны
- C) Программа аварийно завершится
- D) Первоначальное исключение будет подавлено

Правильный ответ: D) Первоначальное исключение будет подавлено

99. Что произойдет, если в `try` и `finally` есть `return`?

- A) Выполнится `return` из `finally`
- B) Выполнится `return` из `try`

С) Будет ошибка компиляции
D) Исключение в `try` прервет выполнение `finally`
Правильный ответ: A) Выполнится return из finally

100. Можно ли перехватить `Error`?
A) Да, через `catch (Error e)`
B) Нет, `Error` нельзя перехватывать
C) Только если он `checked`
D) Только в `static` методах

Правильный ответ: A) Да, через catch (Error e)

101. Какой функциональный интерфейс используется для представления **лямбда-выражений**, которые принимают два аргумента и возвращают результат?

A) `Function<T, R>`
B) `Predicate<T>`
C) `BiFunction<T, U, R>`
D) `Consumer<T>`

Правильный ответ: C) BiFunction<T, U, R

102. Как можно **параллельно** обрабатывать элементы списка с помощью лямбда-выражений?

A) Использовать `stream().parallel()`
B) Использовать `forEach()`
C) Применять `Thread` для каждого элемента
D) Использовать `forEachOrdered()`

Правильный ответ: A) Использовать stream().parallel()

103. Что произойдет при вызове следующего кода?

```
List<String> list = Arrays.asList("apple", "banana", "cherry");  
list.forEach(e -> System.out.print(e + " "));
```

A) Выведет все элементы списка через пробел
B) Выведет элементы списка на отдельных строках
C) Вызовет ошибку компиляции
D) Ничего не выведет

Правильный ответ: A) Выведет все элементы списка через пробел

104. Что произойдет, если в лямбда-выражении отсутствует возвращаемое значение, но используется `return`?

A) Компилятор выдаст ошибку
B) Лямбда-выражение будет правильно выполнено
C) Лямбда-выражение завершится без ошибок
D) Будет выброшено исключение

Правильный ответ: A) Компилятор выдаст ошибку

105. Какой из следующих интерфейсов **не является функциональным**?

A) `Runnable`
B) `Comparator`
C) `Callable`
D) `Thread`

Правильный ответ: D) Thread

106. Какой из методов в `Stream` **не является промежуточным**?

- A) `map()`
- B) `filter()`
- C) `reduce()`
- D) `forEach()`

Правильный ответ: D) `forEach()`

107. Какая ошибка возникнет, если при использовании `Stream` не указать функцию `collect()` в конце цепочки?

- A) `NullPointerException`
- B) `UnsupportedOperationException`
- C) Ошибка компиляции
- D) Стрим не будет работать, так как не указан терминальный метод

Правильный ответ: D) Стрим не будет работать, так как не указан терминальный метод

108. Что возвращает метод `Optional.map()`?

- A) Новый объект типа `Optional`
- B) Исходное значение
- C) Преобразованное значение типа `T`
- D) `null`, если значение отсутствует

Правильный ответ: A) Новый объект типа `Optional`

109. Какой метод используется для обработки пустого значения в `Optional`?

- A) `orElse()`
- B) `get()`
- C) `map()`
- D) `filter()`

Правильный ответ: A) `orElse()`

110. Как лямбда-выражение может быть использовано для вызова метода с несколькими аргументами?

- A) Передавая параметры через запятую
- B) Через `Varargs`
- C) Используя метод с несколькими `Function`
- D) В лямбда-выражении можно передавать только один аргумент

Правильный ответ: A) Передавая параметры через запятую

111. Какой метод используется для поиска первого совпадения с регулярным выражением?

- A) `find()`
- B) `matches()`
- C) `search()`
- D) `pattern()`

Правильный ответ: B) `matches()`

112. Как создать регулярное выражение для проверки электронной почты?

- A) `^[A-Za-z0-9+_.-]+@[A-Za-z0-9.-]+$`
- B) `^[a-zA-Z0-9_]+@[a-zA-Z]+\.[a-z]+$`
- C) `^.+ $`
- D) `^\w+@\w+\.\w{2,3}$`

Правильный ответ: A) `^[A-Za-z0-9+_.-]+@[A-Za-z0-9.-]+$`

113. Что произойдет, если регулярное выражение **не совпадает** с текстом в методе `Pattern.matcher()`?
- A) Исключение будет выброшено
 - B) Метод вернет `false`
 - C) Метод вернет пустой список
 - D) Метод вернет пустую строку
- Правильный ответ: B) Метод вернет false**
114. Какой из следующих методов используется для **замены текста**, соответствующего регулярному выражению?
- A) `replaceAll()`
 - B) `replaceFirst()`
 - C) `substitute()`
 - D) `transform()`
- Правильный ответ: A) replaceAll()**
115. Какой метасимвол в регулярных выражениях используется для **повторения предыдущего символа**?
- A) `*`
 - B) `+`
 - C) `?`
 - D) `{n}`
- Правильный ответ: A)**
116. Какой метод в `Matcher` используется для **нахождения всех совпадений** с регулярным выражением?
- A) `find()`
 - B) `matches()`
 - C) `group()`
 - D) `replaceAll()`
- Правильный ответ: A) find()**
117. Какой модификатор используется для игнорирования регистра в регулярном выражении?
- A) `(?i)`
 - B) `(?a)`
 - C) `(?I)`
 - D) `(?r)`
- Правильный ответ: A) (?i)**
118. Какой метод используется для **проверки, соответствует ли строка регулярному выражению**?
- A) `isMatch()`
 - B) `matches()`
 - C) `check()`
 - D) `validate()`
- Правильный ответ: B) matches()**
119. Как создать регулярное выражение для **даты в формате yyyy-mm-dd**?
- A) `\d{4}-\d{2}-\d{2}`
 - B) `\d{2}/\d{2}/\d{4}`
 - C) `\d{2}-\d{2}-\d{4}`

D) \d{4}\/\d{2}\/\d{2}

Правильный ответ: A) \d{4}-\d{2}-\d{2}

120. Какой метод используется для **подключения к базе данных** с использованием JDBC?

A) DriverManager.getConnection()

B) Database.connect()

C) Connection.create()

D) Database.connect()

Правильный ответ: A) DriverManager.getConnection()

121. Что необходимо сделать для **избежания утечек памяти** при работе с JDBC?

A) Закрыть все ресурсы, такие как Statement и ResultSet

B) Использовать try-with-resources

C) Оба варианта верны

D) Ничего не нужно делать, все автоматически закрывается

Правильный ответ: C) Оба варианта верны

122. Как правильно **обновить запись** в базе данных через JDBC?

A) Использовать метод executeUpdate()

B) Использовать executeQuery()

C) Использовать execute()

D) Использовать executeInsert()

Правильный ответ: A) Использовать метод executeUpdate()

123. Какая ошибка может быть вызвана при работе с JDBC в случае **неправильной строки подключения**?

A) SQLException

B) SQLSyntaxErrorException

C) NullPointerException

D) IOException

Правильный ответ: A) SQLException

124. Какая из следующих **параметрических форм** используется для **предотвращения SQL инъекций** в JDBC?

A) PreparedStatement

B) Statement

C) CallableStatement

D) Both A and C

Правильный ответ: A) PreparedStatement

125. Какой метод **не используется** для выполнения SQL-запросов в JDBC?

A) executeQuery()

B) executeUpdate()

C) execute()

D) executeDelete()

Правильный ответ: D) executeDelete()

126. Какой метод **предназначен** для **извлечения данных** из таблицы в базе данных с помощью JDBC?

A) executeQuery()

B) executeUpdate()

- C) `execute()`
- D) `getResultSet()`

Правильный ответ: A) `executeQuery()`

127. Какой объект используется для извлечения результата SQL-запроса?

- A) `ResultSet`
- B) `ResultSetMetaData`
- C) `Statement`
- D) `PreparedStatement`

Правильный ответ: A) `ResultSet`

128. Какой метод используется для вставки данных в таблицу базы данных через JDBC?

- A) `executeInsert()`
- B) `executeUpdate()`
- C) `addBatch()`
- D) `insertBatch()`

Правильный ответ: B) `executeUpdate()`

129. Какой интерфейс используется для парсинга XML-документов с использованием стандартных библиотек Java?

- A) `DocumentBuilderFactory`
- B) `XMLStreamReader`
- C) `SAXParser`
- D) `DOMParser`

Правильный ответ: A) `DocumentBuilderFactory`

130. Какой метод используется для получения элементов из XML-документа в библиотеке `DocumentBuilder`?

- A) `getElementById()`
- B) `getElementsByTagName()`
- C) `getNodesByName()`
- D) `getElementByTag()`

Правильный ответ: B) `getElementsByTagName()`

131. Какой класс представляет элемент XML-документа в модели DOM?

- A) `Element`
- B) `Node`
- C) `TextNode`
- D) `XMLNode`

Правильный ответ: A) `Element`

132. Какое исключение может быть выброшено при парсинге некорректного XML-файла с использованием `DocumentBuilder`?

- A) `SAXException`
- B) `IOException`
- C) `ParserConfigurationException`
- D) Все перечисленные

Правильный ответ: D) Все перечисленные

133. Как создать объект `Document` из строки XML с помощью DOM API?

- A) Использовать `new Document()`
- B) Использовать `DocumentBuilder.parse()`

- C) Использовать `Transformer`
- D) Использовать `DocumentBuilderFactory.newInstance()`

Правильный ответ: B) Использовать `DocumentBuilder.parse()`

134. Какой метод используется для **сохранения XML-документа** в файл с помощью Java?

- A) `save()`
- B) `write()`
- C) `transform()`
- D) `store()`

Правильный ответ: C) `transform()`

135. Как в библиотеке **JAXB** происходит процесс **сериализации объектов в XML**?

- A) С помощью аннотации `@XmlElement`
- B) С помощью метода `marshal()`
- C) С помощью метода `unmarshal()`
- D) С помощью интерфейса `XMLSerializer`

Правильный ответ: B) С помощью метода `marshal()`

136. Какое исключение будет выброшено, если попытаться получить **некорректный элемент** из XML с помощью `getElementsByName()` ?

- A) `NullPointerException`
- B) `IndexOutOfBoundsException`
- C) `DOMException`
- D) `NoSuchElementException`

Правильный ответ: A) `NullPointerException`

137. Как в XML с использованием **JAXP (Java API for XML Processing)** можно обрабатывать события при разборе XML?

- A) С использованием `SAXParser`
- B) С использованием `DOMParser`
- C) С использованием `XMLStreamReader`
- D) С использованием `XMLDecoder`

Правильный ответ: A) С использованием `SAXParser`

138. Какой паттерн проектирования применяется для создания объектов, где конкретный класс может быть выбран на основе параметров, переданных в конструктор?

- A) `Singleton`
- B) `Factory Method`
- C) `Builder`
- D) `Prototype`

Правильный ответ: B) `Factory Method`

139. В паттерне **Observer** какой объект **оповещает** все подписанные объекты о произошедших изменениях?

- A) `Subject`
- B) `Observer`
- C) `Client`
- D) `Handler`

Правильный ответ: A) `Subject`

140. Какой паттерн проектирования позволяет **сделать один экземпляр класса доступным для всех клиентов** и обеспечивает глобальный доступ?
- A) Factory
 - B) Singleton
 - C) Adapter
 - D) Proxy
- Правильный ответ: B) Singleton**
141. В каком случае применим паттерн **Strategy**?
- A) Когда необходимо определить алгоритм, но клиент не должен знать его внутреннюю реализацию
 - B) Когда нужно использовать одну и ту же логику в различных классах
 - C) Когда есть несколько возможных реализаций одного интерфейса
 - D) Когда нужно скрыть сложность системы от пользователя
- Правильный ответ: A) Когда необходимо определить алгоритм, но клиент не должен знать его внутреннюю реализацию**
142. В паттерне **Decorator** как можно добавить новое поведение в объект?
- A) Создавая новый класс, который расширяет базовый класс
 - B) Добавляя новое поведение к существующему классу
 - C) Используя фабричный метод для добавления поведения
 - D) Изменяя параметры конструктора класса
- Правильный ответ: B) Добавляя новое поведение к существующему классу**
143. Какой паттерн проектирования используется, чтобы обеспечить **единый интерфейс** для группы объектов?
- A) Command
 - B) Composite
 - C) Observer
 - D) Chain of Responsibility
- Правильный ответ: B) Composite**
144. Какой паттерн проектирования используется для реализации **отложенной инициализации** объектов в момент их первого использования?
- A) Prototype
 - B) Singleton
 - C) Lazy Initialization
 - D) Abstract Factory
- Правильный ответ: C) Lazy Initialization**
145. Паттерн **Builder** чаще всего используется, когда необходимо создать объект с **множеством параметров**, некоторые из которых могут быть опциональными. Что из следующего верно?
- A) Каждый объект может быть создан с помощью конструктора с множеством аргументов
 - B) Все параметры обязательны и должны быть переданы
 - C) Параметры могут быть указаны последовательно через цепочку методов
 - D) Строитель создает один конкретный объект для всех случаев
- Правильный ответ: C) Параметры могут быть указаны последовательно через цепочку методов**

146. В паттерне **Facade** используется следующий принцип:
A) Скрыть сложность системы, предоставив простой интерфейс
B) Разделить систему на несколько частей
C) Преобразовать один интерфейс в другой
D) Позволить объектам взаимодействовать без прямого контакта
Правильный ответ: A) Скрыть сложность системы, предоставив простой интерфейс
147. Какой метод используется для **создания нового потока** в Java?
A) `Thread.start()`
B) `Thread.run()`
C) `Thread.create()`
D) `Thread.execute()`
Правильный ответ: A) Thread.start()
148. Что произойдет, если вызвать метод `run()` в потоке напрямую, а не `start()`?
A) Поток выполнится в том же потоке, в котором был вызван метод
B) Поток не будет запущен
C) Поток будет запущен синхронно
D) Поток будет завершен сразу
Правильный ответ: A) Поток выполнится в том же потоке, в котором был вызван метод
149. Какой интерфейс является основным для создания многопоточных задач в Java?
A) `Runnable`
B) `Callable`
C) `Thread`
D) `Executor`
Правильный ответ: A) Runnable
150. Как можно синхронизировать доступ к разделяемому ресурсу между несколькими потоками?
A) Использовать `synchronized` блоки или методы
B) Использовать `volatile` переменные
C) Использовать `Thread.sleep()`
D) Использовать `Thread.yield()`
Правильный ответ: A) Использовать synchronized блоки или методы
151. Какой метод используется для **ожидания завершения потока**?
A) `Thread.join()`
B) `Thread.wait()`
C) `Thread.sleep()`
D) `Thread.stop()`
Правильный ответ: A) Thread.join()
152. Какой метод блокирует выполнение потока до тех пор, пока другой поток не завершит свою работу?
A) `wait()`
B) `join()`
C) `sleep()`
D) `stop()`

Правильный ответ: B) join()

153. Как можно избежать **deadlock** при синхронизации потоков?
A) Использовать только один блок `synchronized`
B) Убедиться, что потоки захватывают блокировки в одном и том же порядке
C) Использовать `Thread.sleep()`
D) Все перечисленное верно
Правильный ответ: B) Убедиться, что потоки захватывают блокировки в одном и том же порядке
154. Какой механизм **предотвращает гонки** между потоками в Java?
A) `synchronized` блоки
B) `volatile` переменные
C) Использование атомарных операций
D) Все перечисленное верно
Правильный ответ: D) Все перечисленное верно
155. Какой класс используется для **реализации пула потоков** в Java?
A) `ExecutorService`
B) `ThreadPoolExecutor`
C) `Thread`
D) Все перечисленное верно
Правильный ответ: A) ExecutorService

Краткое описание и регламент выполнения

К тестам допускаются все обучающиеся.

По результатам итогового теста обучающийся может набрать максимально 40 баллов.

7.2.2. Пример практической работы

Практическая работа №3 «Лямбда-выражения»

Цель работы: сформировать навыки применения лямбда выражений при разработке приложений для повышения читабельность и уменьшения количество кода.

Задача 1. Лямбда выражение для интерфейса `Printable`.

Написать лямбда выражение для интерфейса `Printable`, который содержит один метод `void print()`.

Задача 2. Проверка пустой строки.

Создать лямбда выражение, которое возвращает значение `true`, если строка не `null`, используя функциональный интерфейс `Predicate`.

Создать лямбда выражение, которое проверяет, что строка не пуста, используя функциональный интерфейс `Predicate`.

Написать программу проверяющую, что строка не `null` и не пуста, используя метод `and()` функционального интерфейса `Predicate`.

Задача 3. Проверка строки.

Написать программу, которая проверяет, что строка начинается буквой “J”или “N” и заканчивается “A”. Используем функциональный интерфейс `Predicate`.

Задача 4. Лямбда выражение для HeavyBox.

Написать лямбда выражение, которое принимает на вход объект типа HeavyBox и выводит на консоль сообщение “Отгрузили ящик с весом n”. “Отправляем ящик с весом n” Используем функциональный интерфейс Consumer и метод по умолчанию andThen.

Задача 5. Лямбда для Function.

Написать лямбда выражение, которое принимает на вход число и возвращает значение “Положительное число”, “Отрицательное число” или “Ноль”. Используем функциональный интерфейс Function.

Задача 6. Лямбда для Supplier.

Написать лямбда выражение, которое возвращает случайное число от 0 до 10. Используем функциональный интерфейс Supplier.

Задача 7. Ссылка на статический метод.

Переделать класс, использующий Printable используя ссылку на статический метод.

Предоставьте отчет, который содержит текст программного кода выполненных заданий и скрин результатов работы. Приложите к отчету скомпилированные программы по выполнению указанных заданий. Все программы следует оформлять в соответствии с Code Convention, при оформлении кода использование комментариев обязательно.

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 межстрочный 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 - 2	6 баллов – задание выполнено в полном объёме 5 баллов – задание выполнено в объёме 70% 3-4 балла - задание выполнено в объёме менее 50%

		1-2 балл – задание выполнено в объеме менее 50% 0 баллов – задание не выполнено.
Отчеты по практическим работам 3 - 8		8 баллов – задание выполнено в полном объеме 7 баллов – задание выполнено в объеме 70% 4-6 балла - задание выполнено в объеме менее 50% 1-3 балл – задание выполнено в объеме менее 50% 0 баллов – задание не выполнено.

7.2.3. Комплект заданий для оценки сформированности компетенций (примеры)

(наименование оценочного средства)

ПК-2 Способен разрабатывать прототипы ИС в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС

код и наименование компетенции

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");
        }
    }
}
```

- Запись текста в файл из определенного каталога
- Чтение текста из файла внутри каталога
- Получение всех подкаталогов и файлов в определенном каталоге
- 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);
}
```

- Создание файла
- Определение объекта для каталога
- Переименование файла
- 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: a.

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: a, 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
- b. Ошибка компиляции в строке 2
- c. 10
- d. Ошибка выполнения в строке 2

Правильный ответ: c.

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.

7.3. Оценочные средства для промежуточной аттестации по итогам освоения дисциплины

7.3.1. Вопросы к промежуточной аттестации

Семестр 5

п/п	Вопросы к экзамену
1.	Какие возможности предоставляют пакеты java.io и java.nio? В чем их основные отличия? Приведите примеры использования каждого из них.
2.	Что такое поток в Java? Как различаются байтовые и символьные потоки? В каких случаях используются одни или другие?
3.	Какие бывают входные и выходные потоки в Java? Приведите примеры классов, относящихся к этим категориям.
4.	Как работают буферизированные потоки? В чем их преимущества перед обычными потоками? Опишите общий алгоритм работы с потоками.
5.	Как устроена иерархия символьных потоков в Java? Какие классы относятся к этой категории? В чем их основные особенности?
6.	Как выглядит иерархия байтовых потоков в Java? Чем они отличаются от символьных потоков?
7.	Какие классы используются для создания потоков в Java? В чем разница между ними?
8.	Какие классы позволяют управлять потоками в Java? Как они используются на практике?
9.	Что такое цепочки потоков в Java? Как они позволяют упрощать работу с вводом-выводом? Приведите примеры.
10.	Какие классы позволяют преобразовывать байтовые данные в примитивные типы в Java? Для чего это нужно?
11.	Как осуществляется вывод на строчные устройства в Java? Какие классы для этого используются?
12.	Какую роль играют потоки класса System в Java? В чем их особенности?
13.	Что представляет собой класс StreamTokenizer? В каких случаях он может быть полезен?
14.	Как работает класс SequenceInputStream? В каких ситуациях он применяется?
15.	Для чего используется класс LineNumberReader? В чем его преимущества?
16.	Как работает класс RandomAccessFile? Как он позволяет работать с файлами в произвольном порядке?
17.	Какие возможности предоставляет класс Console? В чем его особенности?
18.	Как управление жизненным циклом потока осуществляется с помощью try-with-resources? Почему это считается хорошей практикой?
19.	Как работает класс FileSystems? Какие функции он выполняет?
20.	Что представляет собой интерфейс Path и класс Paths? В чем их различие и предназначение?
21.	Как работает класс Files? Какие удобные методы он предоставляет?
22.	Какие особенности работы с файлами в режиме Random Access? В каких ситуациях это полезно?
23.	Что такое исключение в Java? Как работает механизм перехвата и обработки исключений? Что представляет собой объект Exception?

п/п	Вопросы к экзамену
24.	Что такое правило Catch or Specify Requirement? Как оно влияет на обработку исключений в Java?
25.	Как выглядит классификация и иерархия исключений в Java? Чем проверяемые (checked) исключения отличаются от непроверяемых (unchecked)?
26.	Какую роль играет класс Throwable? Почему он является базовым для всех исключений?
27.	Как можно обрабатывать исключения с использованием try-with-resources? Почему это улучшает код?
28.	Как происходит выбрасывание исключений в Java? Что такое цепочки исключений и как их можно использовать?
29.	Какие возможности предоставляет Logging API? Как он помогает в обработке ошибок и ведении логов?
30.	Как взаимодействуют потоки и процессы в Java? Что такое стандарт Posix и как он связан с областями памяти JVM?
31.	Какие существуют способы создания потоков в Java? Чем отличается использование класса Thread от интерфейса Runnable?
32.	Как можно приостанавливать и прерывать выполнение потока? Какие методы для этого используются?
33.	Как работают методы wait() и notify()? Каков общий алгоритм работы с монитором в Java?
34.	Чем потоки-пользователи (user threads) отличаются от потоков-демонов (daemon threads)? В каких ситуациях используются демоны?
35.	Что такое правило happens-before? Как оно влияет на порядок выполнения многопоточного кода?
36.	Как реализуется синхронизация в Java? Что такое критические секции?
37.	Что представляет собой Atomic переменные? Как они помогают в многопоточных программах?
38.	Как работает ключевое слово volatile? Когда его стоит использовать?
39.	Какие проблемы могут возникнуть в многопоточных программах на Java? Как их можно избежать?
40.	Как реализуются блокировки (Locks) в Java? Чем они отличаются от synchronized?
41.	Как работает Executors? В чем их преимущества при управлении потоками?
42.	Что такое Thread Pools и зачем они нужны? Как они помогают в оптимизации работы потоков?
43.	Как работает Fork/Join Framework? В чем его преимущества для обработки больших объемов данных?
44.	Какие коллекции в Java являются потокобезопасными? В чем их особенности?
45.	Что такое XML? Какова структура XML-документа? Какие правила нужно соблюдать при его создании?
46.	Какие элементы могут присутствовать в XML-документе? Как они взаимодействуют друг с другом?
47.	Что такое схема XML-документа? Как работает DTD?
48.	Как работает XSD? Чем он отличается от DTD?
49.	Какие API предоставляет Java для работы с XML? Как их можно использовать?
50.	Какие виды парсеров XML существуют? В чем их различия?
51.	Как работает DOM-парсер? Какие узлы он поддерживает?

п/п	Вопросы к экзамену
52.	Какие методы предоставляет DOM-парсер? Как с их помощью можно обработать XML?
53.	Как работает SAX-парсер? Как он обрабатывает XML-документы?
54.	Как можно построить приложение с использованием SAX-парсера? Какие преимущества у данного метода?
55.	Что такое JAXP? Какие технологии входят в этот API?
56.	Что такое JAXB? Как он используется для работы с XML?
57.	Что такое JDBC? Какова его структура и основные компоненты?
58.	Как JDBC используется в двухуровневых и многоуровневых приложениях?
59.	Какие существуют категории JDBC-драйверов? В чем их различия?
60.	Как работает java.sql.DriverManager? Как он управляет подключениями?
61.	Как использовать java.sql.Connection для работы с базой данных?
62.	Какие возможности предоставляет java.sql.Statement?
63.	Как работает java.sql.ResultSet? Какие методы он поддерживает?
64.	Каков общий алгоритм работы с JDBC? Как выполняются запросы?
65.	Как работают PreparedStatement и CallableStatement? Чем они отличаются от обычных Statement?
66.	Как обрабатывать SQLException в Java? Как правильно вести логирование ошибок?
67.	Какие существуют порождающие паттерны проектирования? Для каких задач они применяются?
68.	Какие структурные паттерны используются в проектировании? Как они помогают улучшить архитектуру кода?
69.	Какие поведенческие паттерны существуют? Как они применяются в практике программирования?
70.	Как выбрать подходящий паттерн проектирования для создания объектов? Какие критерии следует учитывать?
71.	Какие существуют порождающие паттерны проектирования? Для каких задач они применяются?
72.	Какие структурные паттерны используются в проектировании? Как они помогают улучшить архитектуру кода?
73.	Что такое паттерны проектирования и зачем они используются?
74.	Приведите пример реализации паттерна Singleton в Java.
75.	В чём преимущества и недостатки паттерна Singleton?
76.	В чём отличие паттерна Factory Method от паттерна Abstract Factory?
77.	Приведите пример использования паттерна Strategy в Java.
78.	Как работает паттерн Observer и где его применение наиболее уместно?
79.	Чем отличается паттерн Decorator от Adapter?
80.	Опишите работу паттерна Facade. Для чего он используется?
81.	Какие паттерны относятся к порождающим, структурным, поведенческим?
82.	Что такое лямбда-выражение в Java и в чём его основное назначение?
83.	Каковы требования к интерфейсу, чтобы его можно было использовать в лямбда-выражении?

п/п	Вопросы к экзамену
84.	Приведите пример лямбда-выражения, реализующего интерфейс Predicate<String>, проверяющий, что строка не пуста.
85.	Чем лямбда-выражение отличается от анонимного класса?
86.	Как связаны лямбда-выражения с понятием " эффективно final "?
87.	Как можно использовать лямбда-выражения совместно с методами map(), filter() и forEach() в Stream API?
88.	Приведите пример использования лямбда-выражения с интерфейсом Function<Integer, String>.
89.	Как работает метод andThen() у интерфейса Consumer и как его можно использовать?
90.	Какие встроенные функциональные интерфейсы содержит пакет java.util.function?
91.	В каких случаях лямбда-выражения повышают читаемость кода, а в каких — могут затруднить понимание?
92.	Что такое регулярное выражение и где в Java они применяются?
93.	Приведите пример регулярного выражения для проверки email-адреса.
94.	Объясните разницу между символами . и \. в регулярных выражениях.
95.	Как с помощью Java проверить, соответствует ли строка заданному шаблону?
96.	Что делает выражение \d{3}-\d{2}-\d{4}?
97.	Какой класс используется для компиляции регулярного выражения и какой — для сопоставления с текстом?
98.	Приведите пример использования метода replaceAll() с регулярным выражением.
99.	Чем полезны флаги Pattern.CASE_INSENSITIVE и Pattern.MULTILINE?
100.	Объясните принцип работы метода find() в классе Matcher.
101.	В чём различие между методами matches() и find()?

7.3.2. Критерии и нормы оценки

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

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	Курбатова, И. В.	Основы программирования на языке Java : учебное пособие для вузов / И. В. Курбатова, А. В. Печуров. — 2-е изд., стер. — Санкт-Петербург : Лань, 2026. — 348 с. — ISBN 978-5-507-51392-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/511511	учебное пособие	2026	ЭБС «Лань»
2	Шнейдеров, Е. Н.	Разработка приложений на языке Java. Лабораторный практикум : учебное пособие / Е. Н. Шнейдеров, А. Ю. Писарчик, В. О. Казюциц. — Минск : БГУИР, 2023. — 92 с. — ISBN 978-985-543-561-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/479516	лабораторный практикум	2023	ЭБС «Лань»
3	Вязовик, Н. А.	Вязовик, Н. А. Программирование на Java : учебное пособие / Н. А. Вязовик. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. — 601 с. — ISBN 978-5-4497-0852-6. — Текст : электронный // Цифровой образовательный ресурс IPR	учебное пособие	2025	ЭБС «IPRbooks»

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно- методическое пособие, практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
		SMART : [сайт]. — URL: https://www.iprbookshop.ru/146383.html			
4	Блох, Дж.	Java. Эффективное программирование / Дж. Блох ; перевод В. Стрельцов ; под редакцией Р. Усманова. — 3-е изд. — Саратов : Профобразование, 2024. — 310 с. — ISBN 978-5-4488-0127-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/145890.html	учебник	2024	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	Васюткина И.А.	Технология разработки объектно- ориентированных программ на JAVA [Электронный ресурс]: учебно- методическое пособие/ И.А. Васюткина. — Электрон. текстовые данные. — Новосибирск: Новосибирский государственный технический университет, 2012. — 152 с.— 978-5- 7782-1973-1.	Учебно- методическое пособие	2012	ЭБС «IPRbooks»

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно-методическое пособие, практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
2	Свистунов А.Н.	Построение распределенных систем на Java [Электронный ресурс] / А.Н. Свистунов. — 2-е изд. — Электрон. текстовые данные.— М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 317 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/73707.html	учебное пособие	2016	ЭБС «IPRbooks»
3	Монажв В.В.	Язык программирования Java и среда NetBeans [Электронный ресурс] / В.В. Монажв. — 2-е изд. — Электрон. текстовые данные.— М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 450 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/73739.html	учебное пособие	2016	ЭБС «IPRbooks»
4	Николаев Е.И.	Объектно-ориентированное программирование. Часть 1 [Электронный ресурс] : лабораторный практикум / Е.И. Николаев.— Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2015. — 183 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/62966.html	лабораторный практикум	2015	ЭБС «IPRbooks»

8.3. Перечень профессиональных баз данных и информационных справочных систем

1. Oracle Java Documentation : официальная документация платформы Java SE и JDK. – Режим доступа: <https://docs.oracle.com/javase/>
2. Java SE API Documentation : справочная система по пакетам, классам и методам Java SE. – Режим доступа: <https://docs.oracle.com/en/java/javase/>
3. Dev.java : официальный образовательный ресурс для изучения Java и инструментов разработки. – Режим доступа: <https://dev.java/>
4. Java Community Process (JCP) : ресурс, содержащий сведения о стандартах и спецификациях платформы Java. – Режим доступа: <https://www.jcp.org/>

8.4. Перечень программного обеспечения

№ п/п	Наименование ПО	Реквизиты (дата, номер, срок действия)
1	Visual Paradigm Community Edition	Бессрочно https://www.visual-paradigm.com/download/community.jsp
2	Java SE Development Kit (JDK)	Бессрочно https://www.oracle.com/java/technologies/downloads/
3	Apache NetBeans IDE	Бессрочно https://netbeans.apache.org/download/index.html
4	Eclipse IDE for Java Developers	Бессрочно https://www.eclipse.org/downloads/
5	IntelliJ IDEA Community Edition	Бессрочно https://www.jetbrains.com/idea/download/

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-401).	Компьютер (монитор 19", системный блок Pentium (R) Dual-Core E5500 2,8 GHz / 4 Gb / 500 Gb), столы ученические , столы компьютерные , стол преподавательский, стулья, доска аудиторная(меловая)
2	Компьютерный класс. Учебная аудитория для проведения занятий лекционного типа.	Системные блоки (HP ProDesk), мониторы (Samsung), коммутатор (D-

№ 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
	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-402).	Link), столы ученические, столы компьютерные, стулья, доска аудиторная.
3	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.
4	Учебная аудитория для проведения занятий лекционного типа. Учебная аудитория для проведения занятий семинарского типа. 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-418).	Стол�ы ученические двухместные (моноблок), доска аудиторная 3-х секционная (меловая), стол преподавательский , стулья, проектор Acer
5	Помещение для самостоятельной работы студентов (Г-401)	Стол�ы, стулья, компьютеры
6	Помещение для самостоятельной работы обучающихся (С-401).	Шкафы для документации, доски магнитные, столы письменные, столы компьютерные