

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЛГПУ»)**

Структурное подразделение Институт физико-математического образования, информационных и обслуживающих технологий
Кафедра информационных образовательных технологий и систем

УТВЕРЖДАЮ

Врио директора ИФМОИОТ

 Е.А. Журавлева
« 14 » 01 2026 г.

Приложение к рабочей программе учебной дисциплины

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
для проведения текущего контроля и промежуточной аттестации
обучающихся по дисциплине
«Программирование на языках высокого уровня»

По направлению подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки)

Профиль подготовки Технология. Информатика

Квалификация выпускника – бакалавр

Форма обучения очная, заочная

Курс ОФО – 2-3 курс, ЗФО – 4 курс

Разработчик

Швыров В.В.

канд. физ.-мат. наук, доцент,
доцент кафедры информационных образовательных технологий и систем

Заведующий кафедрой

 Д.А. Капустин

Протокол от «13» сентября 2026 г. № 11

Луганск, 2026

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

1.1. Область применения

Фонд оценочных средств (ФОС) – неотъемлемая часть рабочей программы дисциплины (модуля) Программирование на языках высокого уровня и предназначен для контроля и оценки образовательных достижений студентов, освоивших программу дисциплины (модуля).

1.2. Цели и задачи фонда оценочных средств

Цель ФОС – установить соответствие уровня подготовки обучающегося требованиям ФГОС ВО бакалавриат / специалитет / магистратура по направлению подготовки 44.03.05 Педагогическое (с двумя профилями подготовки), утвержденным приказом Министерства науки и высшего образования Российской Федерации от 22 февраля 2018 г. № 125 (с изменениями и дополнениями от 26.11.2020 г.)

1.3. Перечень компетенций, формируемых в процессе освоения основной образовательной программы

Процесс освоения дисциплины направлен на формирование следующих компетенций и индикаторов их достижения:

Код по ФГОС ВО	Индикатор достижения
Универсальные	
Общепрофессиональные	
ОПК-1. Способен осуществлять профессиональную деятельность в соответствии с нормативными правовыми актами в сфере образования и нормами профессиональной этики	ОПК-1.1. Понимает и объясняет сущность приоритетных направлений развития образовательной системы Российской Федерации, законов и иных нормативно-правовых актов, регламентирующих образовательную деятельность в Российской Федерации, нормативных документов по вопросам обучения и воспитания детей и молодежи, федеральных государственных образовательных стандартов дошкольного, начального общего, основного общего, среднего общего, среднего профессионального образования, профессионального обучения, законодательства о правах ребенка, трудового законодательства. ОПК-1.2. Применяет в своей деятельности основные нормативно-правовые акты в сфере образования и нормы профессиональной этики, обеспечивает конфиденциальность сведений о субъектах образовательных отношений, полученных в процессе профессиональной

	деятельности.
Профессиональные	
ПК-1 Способен осваивать и использовать теоретические знания и практические умения и навыки в предметной области при решении профессиональных задач	ПК-1.1. Знает структуру, состав и дидактические единицы предметной области (преподаваемого предмета). ПК-1.2. Умеет осуществлять отбор учебного содержания для его реализации в различных формах обучения в соответствии с требованиями ФГОС ОО. ПК-1.3. Демонстрирует умение разрабатывать различные формы учебных занятий, применять методы, приемы и технологии обучения, в том числе информационные.

1.4. Этапы формирования компетенций и средства оценивания уровня их сформированности

Этапы формирования компетенций	Компетенции	Контрольно-оценочные средства / способ оценивания
Тема 1. Java Development Kit	ОПК-1; ПК-1	Устный опрос
Тема 2. Синтаксис языка Java	ОПК-1; ПК-1	Устный опрос. Защита лабораторных работ
Тема 3. Типы данных. Преобразование типов	ОПК-1; ПК-1	Устный опрос. Защита лабораторных работ
Тема 4. Условный оператор. Оператор множественного выбора switch	ОПК-1; ПК-1	Устный опрос. Защита лабораторных работ
Тема 5. Циклы с пред- и постусловием. Цикл for. Цикл for-each	ОПК-1; ПК-1	Устный опрос. Защита лабораторных работ
Тема 6. Одномерные массивы. Двумерные массивы	ОПК-1; ПК-1	Устный опрос. Защита лабораторных работ
Тема 7. Строки. StringBuilder vs StringBuffer	ОПК-1; ПК-1	Устный опрос. Защита лабораторных работ
Тема 8. Основные принципы ООП. Класс Object. Методы (объявление, вызов, аргументы). Классы-обёртки	ОПК-1; ПК-1	Устный опрос. Защита лабораторных работ
Тема 9. Конструкторы. Ключевые слова static и final. Интерфейсы	ОПК-1; ПК-1	Устный опрос. Защита лабораторных работ
Тема 10. Работа с файлами в Java	ОПК-1; ПК-1	Выполнение лабораторных работ
Тема 11. Коллекции	ОПК-1; ПК-1	Выполнение лабораторных работ
Тема 12. Stream API	ОПК-1; ПК-1	Выполнение лабораторных работ

Тема13. JavaFX в IntelliJ IDEA	ОПК-1; ПК-1	Выполнение лабораторных работ
Текущая аттестация	ОПК-1; ПК-1	Индивидуальная работа
Промежуточная аттестация	ОПК-1; ПК-1	Экзамен (письменный)

1.5. Описание показателей формирования компетенций

Код компетенции	Результаты сформированности
ОПК-1. Способен осуществлять профессиональную деятельность в соответствии с нормативными правовыми актами в сфере образования и нормами профессиональной этики	ОПК-1.1. Понимает и объясняет сущность приоритетных направлений развития образовательной системы Российской Федерации, законов и иных нормативно-правовых актов, регламентирующих образовательную деятельность в Российской Федерации, нормативных документов по вопросам обучения и воспитания детей и молодежи, федеральных государственных образовательных стандартов дошкольного, начального общего, основного общего, среднего общего, среднего профессионального образования, профессионального обучения, законодательства о правах ребенка, трудового законодательства. ОПК-1.2. Применяет в своей деятельности основные нормативно-правовые акты в сфере образования и нормы профессиональной этики, обеспечивает конфиденциальность сведений о субъектах образовательных отношений, полученных в процессе профессиональной деятельности.
ПК-1 Способен осваивать и использовать теоретические знания и практические умения и навыки в предметной области при решении профессиональных задач	ПК-1.1. Знает структуру, состав и дидактические единицы предметной области (преподаваемого предмета). ПК-1.2. Умеет осуществлять отбор учебного содержания для его реализации в различных формах обучения в соответствии с требованиями ФГОС ОО. ПК-1.3. Демонстрирует умение разрабатывать различные формы учебных занятий, применять методы, приемы и технологии обучения, в том числе информационные.

1.6. Критерии оценивания компетенций на разных этапах их формирования

Вид учебной работы	Количество баллов		
4-5 семестр / В – С триместр			
	ОФО	О-ЗФО	ЗФО
Оформление отчетов по лабораторным работам	30 баллов	-	30 баллов
Работа на лабораторных занятиях	30 баллов	-	30 баллов
Выполнение тестовых заданий	-	-	-
Выполнение заданий самостоятельной работы	10 баллов	-	10 баллов
экзамена	30 баллов	-	30 баллов
Итого за семестр:	100 баллов	-	100 баллов

Всего	100 баллов
--------------	-------------------

Накопительная система оценивания по 100-балльной шкале

Четырехбал- льная система оценивания экзамена	100- балльная шкала	Буквенная шкала, соответствующая 100- балльной шкале	Система оценивания зачета
Отлично	90–100	А – отлично – теоретическое содержание курса освоено полностью, без пробелов; необходимые практические навыки работы с освоенным материалом сформированы; все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	Зачтено
Хорошо	83–89	В – очень хорошо – теоретическое содержание курса освоено полностью, без пробелов; необходимые практические навыки работы с освоенным материалом в основном сформированы; все предусмотренные программой обучения учебные задания выполнены, качество выполнения большинства из них оценено числом баллов, близким к максимальному	
Хорошо	75–82	С – хорошо – теоретическое содержание курса освоено полностью; некоторые практические навыки работы с освоенным материалом сформированы недостаточно; все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	
Удовлетво- рительно	63–74	Д – удовлетворительно – теоретическое содержание дисциплины освоено частично, но пробелы не носят существенного характера; необходимые практические навыки работы с освоенным материалом в основном сформированы; большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, содержат ошибки	
Удовлетво- рительно	50–62	Е – посредственно – теоретическое содержание курса освоено частично; некоторые практические навыки работы не сформированы, многие предусмотренные программой обучения учебные задания не выполнены либо качество выполнения некоторых из них оценено числом баллов, близким к минимальному	
Неудовлетво- рительно	21–49	FX – неудовлетворительно – теоретическое содержание курса освоено частично; необходимые практические навыки работы	Не зачтено

		не сформированы; большинство предусмотренных программой обучения учебных заданий не выполнено либо качество их выполнения оценено числом баллов, близким к минимальному; при дополнительной самостоятельной работе над материалом курса возможно повышение качества выполнения учебных заданий	
Неудовлетворительно	0–20	F – неудовлетворительно – теоретическое содержание курса не освоено; необходимые практические навыки работы не сформированы; все выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над материалом курса не приведет к какому-либо значимому повышению качества выполнения учебных заданий	

2. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА

2.1. Оценочные средства текущего контроля (типовые)

Вопросы для текущего контроля:

1. Какова история языка Java?
2. Что такое JDK и какие основные компоненты в него входят?
3. Чем отличается JDK от JRE и JVM?
4. Какова роль компилятора `javac` и интерпретатора `java`?
5. Что такое `JAVA_HOME` и для чего эта переменная окружения нужна?
6. Какие основные инструменты командной строки поставляются с JDK и для чего они используются (например, `jar`, `javadoc`, `jconsole`)?
7. Каковы основные правила именования классов, методов и переменных в Java?
8. Что такое `main`-метод и каков его обязательный синтаксис?
9. Какие типы комментариев поддерживает Java?
10. Что такое пакеты (`packages`) и для чего они нужны? Как используется ключевое слово `import`?
11. Какие есть примитивные типы данных в Java? Приведите примеры.
12. Чем отличаются примитивные типы данных от ссылочных?
13. Что такое явное (`explicit`) и неявное (`implicit`) преобразование типов? Приведите примеры.
14. Что такое «автоупаковка» (`autoboxing`) и «распаковка» (`unboxing`)?
15. Какие потери данных возможны при преобразовании типов (например, `double` в `int`)?
16. Что такое переполнение (`overflow`) при работе с целочисленными типами?
17. Напишите базовую конструкцию оператора `if-else-if`.
18. Чем отличается оператор `switch` от цепочки `if-else`? В каких случаях его использование предпочтительнее?
19. Какие типы данных можно использовать в качестве выражения для `switch` (начиная с разных версий Java: до Java 7, в Java 7+, в Java 14+)?
20. Что такое «проваливание» (`fall-through`) в операторе `switch` и как его избежать?
21. Что такое `switch-выражение` (`Switch Expressions`), появившееся в Java 14? Чем оно отличается от оператора `switch`?
22. Как объявить, создать (инициализировать) и заполнить одномерный массив?
23. Что такое длина массива и как ее получить? Можно ли ее изменить после создания?
24. Как передается массив в метод: по значению или по ссылке?
25. Как объявить и создать двумерный (нерегулярный, `jagged`) массив?
26. Как скопировать массив? Чем `System.arraycopy()` отличается от `Arrays.copyOf()`?
27. Почему объекты класса `String` являются неизменяемыми (`immutable`)? Какие в этом преимущества и недостатки?
28. Чем отличается инициализация строк через литерал (`String s = "hello"`) от создания через конструктор (`String s = new String("hello")`)?
29. Что такое пул строк (`String Pool`)?
30. В чем ключевая разница между `StringBuilder` и `StringBuffer`? Когда что использовать?
31. Как правильно сравнивать строки: оператором `==` или методом `.equals()`? Почему?
32. Дайте краткое определение четырем основным принципам ООП: инкапсуляция, наследование, полиморфизм, абстракция.

33. От какого класса неявно наследуются все классы в Java? Назовите 5 основных методов этого класса.
34. Что такое перегрузка (overloading) и переопределение (overriding) методов? В чем разница?
35. Что такое сигнатура метода?
36. Для чего нужны классы-обертки (Wrapper Classes)? Назовите их.
37. Что такое конструктор? Чем конструктор по умолчанию отличается от конструктора с параметрами?
38. Что такое «цепочка вызовов конструкторов» и как работает ключевое слово `this()`?
39. Что обозначают и где применяются ключевые слова `static` (для переменных, методов, блоков) и `final` (для переменных, методов, классов)?
40. Чем абстрактный класс отличается от интерфейса (рассмотрите версии Java до 8 и после)?
41. Какие возможности были добавлены в интерфейсы в Java 8 и Java 9?
42. Какие основные классы для работы с файлами и потоками ввода-вывода вы знаете (из пакетов `java.io` и `java.nio`)?
43. В чем разница между классами `FileInputStream/FileOutputStream` и `FileReader/FileWriter`?
44. Что такое буферизация ввода-вывода и какие классы ее предоставляют (например, `BufferedReader`)?
45. Что такое `try-with-resources` и почему его использование предпочтительнее ручного закрытия потоков?
46. Какие основные классы и методы появились в Java NIO (`Path`, `Files`, `Paths`) для более удобной работы с файлами?
47. Нарисуйте или опишите иерархию основных интерфейсов коллекций (`Collection`, `List`, `Set`, `Map`, `Queue`).
48. В чем разница между `ArrayList` и `LinkedList`? Когда что использовать?
49. Чем `HashSet` отличается от `TreeSet`? Как `LinkedHashSet` сочетает их свойства?
50. Как устроена и работает `HashMap`? Что такое хэш-код, коллизия и как она разрешается?
51. В чем разница между `HashMap`, `Hashtable` и `ConcurrentHashMap`? Что такое `fail-fast` и `fail-safe` итераторы?
52. Что такое `Stream API` и для чего оно предназначено? В чем его преимущество перед обычными циклами?
53. Какие два типа операций есть в `Stream API`?
54. В чем разница между `filter()` и `findFirst()/findAny()`?
55. Объясните разницу между промежуточными (intermediate) и терминальными (terminal) операциями. Приведите примеры.
56. Что такое лямбда-выражения и как они связаны с функциональными интерфейсами?
57. Как работает оператор `map`? Чем он отличается от оператора `flatMap`?
58. Что такое метод `reduce` и для каких задач он применяется?
59. Как создать новый JavaFX-проект в IntelliJ IDEA?
60. Как добавить JavaFX в существующий проект?
61. Как настроить VM Options для запуска JavaFX-приложения в IntelliJ IDEA?
62. Как интегрировать Scene Builder в IntelliJ IDEA?
63. Как связать FXML-файл с контроллером в IDEA?
64. Как обрабатывать события нажатие кнопки через FXML?
65. Как обрабатывать события нажатие radio через FXML?
66. Как обрабатывать события нажатие check box через FXML?
67. Какова типичная структура JavaFX-проекта в IDEA?
68. Как правильно загружать FXML и ресурсы (CSS, изображения)?
69. Как отлаживать JavaFX-приложение в IDEA?
70. Какие полезные плагины для JavaFX можно установить?

2.2. Оценочные средства для промежуточной аттестации

Вопросы для проведения аттестации

1. Происхождение языка Java?
2. Почему Java называется платформно-независимым языком?
3. В чем философская разница между абстрактным классом и интерфейсом?.
4. Можно ли переопределить `static` или `private` метод?
5. Что такое ковариантность и контратравариантность? (На примере массивов и дженериков).
6. Что такое двойная проверка блокировки (Double-Checked Locking) и проблема с ней?
7. Что такое паттерн Singleton?
8. Как его правильно реализовать в Java с учетом многопоточности?
9. Что такое JVM, JRE и JDK? В чем разница?
10. Объясните принципы ООП: инкапсуляция?
11. Объясните принципы ООП: наследование?
12. Объясните принципы ООП: полиморфизм?
13. Объясните принципы ООП: абстракция?
14. В чем разница между классом и объектом?
15. Что такое конструктор? Какие типы конструкторов бывают?
16. Объясните модификаторы доступа (`private`, `default`, `protected`, `public`).
17. В чем разница между примитивными типами данных и ссылочными?
18. Что такое `String`, `StringBuilder` и `StringBuffer`? В чем их ключевое отличие (мутабельность/иммутабельность, потокобезопасность)?
19. Что такое перегрузка (`overloading`) и переопределение (`overriding`) методов?
20. Расскажите о ключевых словах `static`, `final`, `this`, `super`.
21. Как работает оператор `==` и метод `.equals()` для объектов?
22. Как передаются аргументы в методы — по значению или по ссылке? (Объясните на примере примитивов и объектов).
23. Для чего нужны ключевые слова `try`, `catch`, `finally`, `throw`, `throws`? 16. Что такое блок `try-with-resources` (появился в Java 7)? 17. Что такое `enum` (перечисление) и для чего он используется?
24. Назовите основные интерфейсы и классы коллекций. Нарисуйте их иерархию.
25. В чем разница между `List`, `Set` и `Map`?
26. Чем `ArrayList` отличается от `LinkedList`? Когда что использовать?
27. Чем `HashSet` отличается от `TreeSet`? Как `HashSet` обеспечивает уникальность элементов?
28. Как работает `HashMap`? Что такое хэш-код, коллизии и как они разрешаются?
29. Чем `HashMap` отличается от `Hashtable`? (Устаревший, но частый вопрос).
30. Что такое `ConcurrentHashMap` и чем она лучше `synchronizedMap`?
31. Что такое `Comparator` и `Comparable`? В чем разница?
32. Что такое `fail-fast` и `fail-safe` итераторы? Приведите примеры.
33. Какие коллекции являются синхронизированными (потокобезопасными) по умолчанию?
34. Что такое поток (`thread`)? Как создать и запустить поток (наследование от `Thread` vs реализация `Runnable`)?
35. В чем разница между процессами и потоками?

36. Объясните состояния жизненного цикла потока (new, runnable, waiting, timed waiting, blocked, terminated).
37. Что такое синхронизация? Ключевое слово `synchronized`.
38. Что такое монитор и взаимная блокировка (deadlock)? Как ее избежать?
39. Что такое `volatile` переменная? Чем отличается от `synchronized`?
40. Что такое `wait()`, `notify()` и `notifyAll()`? Где они определены?
41. Что такое `ExecutorService` и пулы потоков (Thread Pools)? Зачем они нужны?
42. Что такое `Callable` и `Future`? Чем `Callable` отличается от `Runnable`?
43. Что такое атомарные типы (`AtomicInteger` и др.) и зачем они нужны?
44. Как устроена память JVM (стэк, куча, `Metaspace/PermGen`, области памяти внутри кучи)?
45. Что такое `Garbage Collection` (сборка мусора)? Как она работает?
46. Какие алгоритмы сборки мусора вы знаете (`Mark-and-Sweep`, `Generational`)?
47. Что такое поколения объектов в куче (`Young`, `Old`, `Permanent`)? Зачем такое разделение?
48. Что такое `finalize()` метод? Почему его использование не рекомендуется?
49. Что такое утечки памяти в Java? Как их можно обнаружить?
50. Что такое функциональный интерфейс? Приведите примеры (`Predicate`, `Function`, `Consumer`, `Supplier`).
51. Что такое лямбда-выражение?
52. Что такое ссылки на методы (method references)?
53. Что такое `Stream API`? Объясните промежуточные (`filter`, `map`, `sorted`) и терминальные (`collect`, `forEach`, `reduce`) операции.
54. В чем разница между `Collection` и `Stream`?
55. Что такое `var` для локальных переменных (Java 10)?
56. Что такое `record` (Java 14) и `sealed` классы (Java 15/17)?
57. Как создать новый JavaFX-проект в IntelliJ IDEA?
58. Как добавить JavaFX в существующий проект?
59. Чем JavaFX отличается от `Swing/AWT`?
60. Основные компоненты архитектуры `Stage` (окно)
61. Основные компоненты архитектуры `Scene` (сцена/контейнер)
62. Основные компоненты архитектуры `Node` (элементы: `Button`, `Label`, etc.)
63. Основные компоненты архитектуры `Pane` (Layout-контейнеры)
64. Что такое `FXML` и его преимущества?
65. Основные типы контейнеров: `HBox`, `VBox` (горизонтальное/вертикальное расположение)
66. Основные типы контейнеров: `BorderPane` (5 областей: `top`, `bottom`, `left`, `right`, `center`)
67. Основные типы контейнеров: `GridPane` (табличное расположение)
68. Основные типы контейнеров: `FlowPane`, `TilePane` (потокное расположение)
69. Основные типы контейнеров: `StackPane` (наложение элементов)
70. Что такое `Observable` и `Property`?
71. Система событий JavaFX: `Event` - базовый класс события
72. Система событий JavaFX: `EventHandler` - обработчик
73. Система событий JavaFX: `EventDispatcher` - диспетчер событий
74. Типы событий: `ActionEvent` (кнопки, меню)
75. Типы событий: `MouseEvent`, `KeyEvent`
76. Типы событий: `WindowEvent`, `DragEvent`
77. Каковы основные правила именования классов, методов и переменных в Java?
78. Что такое `main`-метод и каков его обязательный синтаксис?
79. Какие типы комментариев поддерживает Java?
80. Что такое пакеты (`packages`) и для чего они нужны? Как используется ключевое слово `import`?
81. Какие есть примитивные типы данных в Java? Приведите примеры.
82. Чем отличаются примитивные типы данных от ссылочных?

83. Что такое явное (explicit) и неявное (implicit) преобразование типов? Приведите примеры.
84. Что такое «автоупаковка» (autoboxing) и «распаковка» (unboxing)?
85. Какие потери данных возможны при преобразовании типов (например, double в int)?
86. Что такое переполнение (overflow) при работе с целочисленными типами?
87. Напишите базовую конструкцию оператора if-else-if.
88. Чем отличается оператор switch от цепочки if-else? В каких случаях его использование предпочтительнее?
89. Какие типы данных можно использовать в качестве выражения для switch (начиная с разных версий Java: до Java 7, в Java 7+, в Java 14+)?
90. Какие есть циклы?
91. Чем break отличается от continue?
92. Что такое тернарный оператор?
93. Что такое «проваливание» (fall-through) в операторе switch и как его избежать?
94. Что такое switch-выражение (Switch Expressions), появившееся в Java 14? Чем оно отличается от оператора switch?
95. Как объявить, создать (инициализировать) и заполнить одномерный массив?
96. Что такое длина массива и как ее получить? Можно ли ее изменить после создания?
97. Как передается массив в метод: по значению или по ссылке?
98. Как объявить и создать двумерный (нерегулярный, jagged) массив?
99. Как скопировать массив? Чем System.arraycopy() отличается от Arrays.copyOf()?
100. Почему объекты класса String являются неизменяемыми (immutable)? Какие в этом преимущества и недостатки?
101. Чем отличается инициализация строк через литерал (String s = "hello") от создания через конструктор (String s = new String("hello"))?
102. Что такое пул строк (String Pool)?
103. В чем ключевая разница между StringBuilder и StringBuffer? Когда что использовать?
104. Как правильно сравнивать строки: оператором == или методом .equals()? Почему?
105. Дайте краткое определение четырем основным принципам ООП: инкапсуляция, наследование, полиморфизм, абстракция.
106. От какого класса неявно наследуются все классы в Java? Назовите 5 основных методов этого класса.
107. Что такое перегрузка (overloading) и переопределение (overriding) методов? В чем разница?
108. В чем разница между == и .equals()?
109. Что такое класс и как его объявить?
110. Что такое метод main и зачем он нужен?
111. Какие примитивные типы данных вы знаете?
112. Как объявить переменную?
113. Чем отличается float от double?
114. Что такое final переменная?
115. Что такое перегрузка методов (overloading)?
116. Что такое сигнатура метода?
117. Для чего нужны классы-обертки (Wrapper Classes)? Назовите их.
118. Что такое конструктор? Чем конструктор по умолчанию отличается от конструктора с параметрами?
119. Что такое «цепочка вызовов конструкторов» и как работает ключевое слово this()?

120. Что обозначают и где применяются ключевые слова `static` (для переменных, методов, блоков) и `final` (для переменных, методов, классов)?
121. Чем абстрактный класс отличается от интерфейса (рассмотрите версии Java до 8 и после)?
122. Какие возможности были добавлены в интерфейсы в Java 8 и Java 9?
123. Какие основные классы для работы с файлами и потоками ввода-вывода вы знаете (из пакетов `java.io` и `java.nio`)?
124. Что такое `super` и `this`?
125. Что такое переопределение метода (`overriding`)?
126. В чем разница между классами `FileInputStream/FileOutputStream` и `FileReader/FileWriter`?
127. Что такое буферизация ввода-вывода и какие классы ее предоставляют (например, `BufferedReader`)?
128. Что такое `try-with-resources` и почему его использование предпочтительнее ручного закрытия потоков?
129. Какие основные классы и методы появились в Java NIO (`Path`, `Files`, `Paths`) для более удобной работы с файлами?
130. Нарисуйте или опишите иерархию основных интерфейсов коллекций (`Collection`, `List`, `Set`, `Map`, `Queue`).
131. В чем разница между `ArrayList` и `LinkedList`? Когда что использовать?
132. Чем `HashSet` отличается от `TreeSet`? Как `LinkedHashSet` сочетает их свойства?
133. Как устроена и работает `HashMap`? Что такое хэш-код, коллизия и как она разрешается?
134. В чем разница между `HashMap`, `Hashtable` и `ConcurrentHashMap`? Что такое `fail-fast` и `fail-safe` итераторы?
135. Что такое `Stream API` и для чего оно предназначено? В чем его преимущество перед обычными циклами?
136. Какие два типа операций есть в `Stream API`?
137. В чем разница между `filter()` и `findFirst()/findAny()`?
138. Объясните разницу между промежуточными (`intermediate`) и терминальными (`terminal`) операциями. Приведите примеры.
139. Что такое лямбда-выражения и как они связаны с функциональными интерфейсами?
140. Как работает оператор `map`? Чем он отличается от оператора `flatMap`?
141. Как создать таблицу с данными?
142. Что такое `CellValueFactory` и `CellFactory`?

Задания для проверки

1. Выведите на экран сообщение: «My name is Java!».
2. Выведите на экран сообщения(соблюдая перевод строк):

```
I like programming!  
This is my homework  
5 + 1  
100
```

3. Выведите на экран сообщения(соблюдая перевод строк):
Math operations

```
900 - 10000
```

```
5 + 100
```

100 + 100

4. Напиши программу, которая выведет в консоль следующее сообщение (каждую строку отдельно):

Изучать Java — это интересно!
Не бойся делать ошибки.
Каждый день ты становишься лучше!

5. Напиши программу, которая выведет в консоль список покупок, по одному пункту на строку.

Список покупок:

1. Хлеб
2. Молоко
3. Яблоки
4. Шоколад

6. Объявите две переменные целочисленного типа $a = 10$ и $b = 91$, а затем сложите их вместе и выведите результат в консоль.
7. Объявите две переменные вещественного типа $a = 10.69$ и $b = 91.34$, а затем выведите результат вычитания a из b в консоль.
8. Реализуйте выражение $(a + b) - 100 * (15 - 1)$ до этого заранее проинициализировав переменные a и b значениями 0 и 1 , соответственно. Значение выведите на экран.
9. Напишите программу, которая выведет в консоль остаток от деления x на y . До команды вывода необходимо инициализировать x и y значениями 10 и 3 , соответственно.
10. Напишите программу, которая выведет в консоль частное от деления x на y . До команды вывода необходимо инициализировать x и y значениями 100 и 10 , соответственно.
11. Напишите программу, которая выведет в консоль остаток от деления x на y . До команды вывода необходимо инициализировать x и y значениями 64 и 8 , соответственно.
12. Напишите программу для нахождения периметра прямоугольника $p = 2 * (a + b)$, до этого заранее проинициализировав переменные a и b значениями 60 и 43 , соответственно.
13. Реализуйте выражение для нахождения периметра квадрата $p = 4 * a$, до этого заранее проинициализировав переменную a значением 100 .
14. Создайте массив из трёх целочисленных значений и проинициализируйте их таким образом, чтобы сумма всех элементов была равна 101 . Вычислите сумму всех элементов массива и присвойте результат переменной `result`.
15. Создай целочисленный массив длиной 5 элементов. Проинициализируй элемент только под третьим индексом значением -7 . Вычисли сумму всех элементов и присвойте результат переменной `result`.
16. Создайте массив типа `String` с длиной в 3 элемента. После создания проинициализируйте каждый индекс массива частью ФИО, где под первым индексом Иванов, под вторым индексом Иван, а под третьим Иванович. Сложите все элементы массива в одну строку, расставляя пробелы.
17. Дан массив строк, содержащий названия трёх городов. Нужно вывести их в консоль **в строго заданном формате**.

Город 1: Москва

Город 2: Лондон

Город 3: Париж

18. Дана переменная `int age = 20`. Если возраст больше 18, вывести в консоль "Взрослый".
19. Пользователь вводит число в переменную `input`. Если число положительное, вывести "Положительное", если отрицательное — "Отрицательное", если ноль — "Это ноль".
20. Пользователь вводит два числа в переменные `inputOne` и `inputTwo`.
Вывести:
"Первое больше", если первое число больше второго,
"Второе больше", если второе больше первого,
"Числа равны", если они одинаковы.
21. На вход дается целое число, которое записывается в переменную `input`. Если значение `input` больше 0, то выводим в консоль сообщение Больше нуля!, если меньше 0, то выводим сообщение Меньше нуля!.
22. Пользователь вводит номер месяца (1–12). Программа определяет время года и выводит:
"Зима" (12, 1, 2),
"Весна" (3–5),
"Лето" (6–8),
"Осень" (9–11).
23. На вход дается целое число от 1 до 7, которое записывается в переменную `input`. Необходимо сопоставить его с днем недели и вывести название в консоль.
Например: 1 -> Понедельник, 2 -> Вторник, ..., 7 -> Воскресенье
24. На вход дается строка, которое записывается в переменную `input`. Если длина строки больше 5 символов, выводим в консоль сообщение Длинная строка!, если меньше 5, выводим в консоль Короткая строка!, если длина равна 5, то выводим Строка средней длины!.
25. Дан массив `array` типа `int`. Необходимо его перебрать через цикл и вывести в консоль только те элементы у которых индексы четные.
Например: в случае массива [9, -9, 0, 0, 88, -65, -7, 43] под четными индексами находятся значения 9, 0, 88, -7.
26. Напиши программу, которая выводит числа от 5 до 1 в обратном порядке, используя цикл `while`.
27. Напиши программу, которая выводит все чётные числа от 2 до 10, используя цикл `while`.
28. Дан массив чисел {5, 3, 8, 2}. Напиши программу, которая вычисляет их сумму, используя цикл `for-each`. Результат присвой переменной `sum`.
29. На вход подается строка, которая записывается в переменную `input`.
Необходимо проверить содержит ли она подстроку `Java`. Результат необходимо присвоить переменной `result`.

Пример: Я изучаю Java -> true, Я изучаю языки программирования -> false
30. На вход подается строка, которая записывается в переменную `input`. Необходимо вычислить ее длину и присвоить результат переменной `result`.

Типовые задания на экзамен

Вариант 1: Библиотечная система Создайте систему для учета книг в библиотеке.

- **Классы:** `Book` (автор, название, год и т.д.), `Library` (коллекция книг).

- **Задача:** Реализовать добавление/удаление книг, поиск по разным критериям, загрузку/сохранение каталога в файл.

Вариант 2: Геометрические фигуры Создайте иерархию геометрических фигур.

- **Классы:** Shape (абстрактный, с методами area(), perimeter()), Circle, Rectangle, Triangle.
- **Задача:** Реализовать вычисление площади и периметра для каждой фигуры. Создать вектор указателей на Shape и продемонстрировать полиморфизм.

Вариант 3: Банковский счет Создайте систему для управления банковскими счетами.

- **Классы:** Account (владелец, баланс, номер счета), Transaction (сумма, тип, дата).
- **Задача:** Реализовать методы пополнения/снятия средств, просмотр истории транзакций. Перегрузить оператор += для пополнения.

Вариант 4: Учебное заведение Создайте систему для учета студентов и преподавателей.

- **Классы:** Person (имя, возраст), Student (наследует Person, + группа, оценки), Professor (наследует Person, + кафедра, предмет).
- **Задача:** Реализовать добавление студентов в группу, выставление оценок, вывод информации о человеке в зависимости от его типа.

Вариант 5: Автопарк Создайте систему учета транспортных средств.

- **Классы:** Vehicle (абстрактный, марка, год, пробег), Car, Truck, Motorcycle.
- **Задача:** Реализовать метод getServiceCost() для каждого типа ТС. Создать класс Fleet, который может подсчитывать общие затраты на обслуживание автопарка.

Вариант 6: Система заказов в кафе Создайте систему для формирования заказов.

- **Классы:** MenuItem (название, цена), Order (список позиций, статус, сумма).
- **Задача:** Реализовать добавление/удаление позиций в заказ, расчет общей суммы, смену статуса заказа (например, "принят", "готовится", "выполнен").

Вариант 7: Торговая система на аукционе Создайте систему для проведения аукциона.

- **Классы:** Item (лот, стартовая цена), Bid (ставка, участник), Auction (лот, список ставок).
- **Задача:** Реализовать метод размещения новой ставки (которая должна быть выше предыдущей), определение победителя аукциона, хранение истории всех ставок в файле.

Вариант 8: Музыкальный плейлист Создайте систему для управления музыкальными треками.

- **Классы:** Song (исполнитель, название, длительность), Playlist (название, список песен).
- **Задача:** Реализовать добавление/удаление песен, перемешивание плейлиста, вычисление общей длительности. Перегрузить оператор + для объединения двух плейлистов.

Вариант 9: Фильмотека Создайте каталог фильмов.

- **Классы:** Movie (название, режиссер, год, жанр, рейтинг), MovieCollection.
- **Задача:** Реализовать поиск по жанру/режиссеру, сортировку по году/рейтингу, рекомендацию случайного фильма. Использовать std::sort с пользовательскими компараторами.

Вариант 10: Социальная сеть (упрощенная) Создайте модель социальной сети.

- **Классы:** User (имя, список друзей, список постов), Post (текст, дата).
- **Задача:** Реализовать добавление в друзья, ленту новостей (посты друзей), поиск общих друзей между двумя пользователями.

Вариант 11: База данных сотрудников (с файлами) Создайте систему для учета сотрудников компании.

- **Классы:** Employee (id, имя, должность, зарплата), Department (название, коллекция сотрудников).
- **Задача:** Реализовать расчет средней зарплаты по отделу, поиск сотрудника с максимальной зарплатой, перевод сотрудника между отделами.

Вариант 12: Умный дом Создайте систему управления устройствами в умном доме.

- **Классы:** Device (абстрактный, с методами turnOn(), turnOff()), Lamp, Thermostat, SecurityCamera.
- **Задача:** Создать класс SmartHome, который содержит коллекцию устройств. Реализовать сценарий "Все дома" (включить свет, выключить камеры) и "Никого нет" (наоборот).

Вариант 13: Матрицы и операции над ними Создайте класс для работы с матрицами.

- **Класс:** Matrix (данные, строки, столбцы).
- **Задача:** Реализовать конструктор копирования, оператор присваивания (правило трех). Перегрузить операторы +, -, *, <<. Реализовать метод для нахождения определителя.