

# Geetest



Тест по основам  
программирования  
на Java

## Тест по основам программирования на Java

Быстрая проверка знаний, охватывающая ключевые аспекты языка Java.



Java

В тест включены темы: Основы Java, Объектно-ориентированное программирование (ООП), Обработка исключений, Ввод-вывод (I/O), Коллекции и Generic'и, а также Многопоточность.

## Тема 1. Основы Java.

1. Какое ключевое слово используется для объявления класса в Java?

- 1) `struct`
- 2) `Класс`
- 3)+ `class`
- 4) `object`

① Для объявления класса в Java используется ключевое слово `class`. За ним следует имя класса, которое должно начинаться с заглавной буквы, а затем фигурные скобки, внутри которых находится тело класса, содержащее его поля и методы.

2. Какой тип данных в Java используется для хранения целых чисел без дробной части?

- 1) `float`
- 2) `double`
- 3)+ `int`
- 4) `boolean`

3. Какой оператор используется для присваивания значения переменной в Java?

- 1) `==`
- 2)+ `=`
- 3) `:=`
- 4) `<-`

4. Какое ключевое слово используется для объявления статической переменной или метода?

- 1) `final`
- 2) `abstract`
- 3)+ `static`
- 4) `public`

5. Что выведет на экран следующий код:

```
java
System.out.println(5 + "2");
```

- 1) 7
- 2)+ 52
- 3) 25
- 4) Ошибка компиляции

6. Какой тип данных используется для хранения дробных чисел с одинарной точностью?

- 1) `double`
- 2)+ `float`
- 3) `int`
- 4) `long`

7. Какой оператор используется для проверки равенства двух примитивных типов?

- 1) `=`
- 2)+ `==`
- 3) `===`
- 4) `equals()`

8. Какое ключевое слово используется для выхода из текущего цикла?

- 1) `continue`
- 2) `return`
- 3)+ `break`
- 4) `exit`

9. Какое ключевое слово используется для пропуска текущей итерации цикла и перехода к следующей?

- 1) `break`
- 2) `exit`
- 3)+ `continue`
- 4) `pass`

10. Что выведет на экран следующий код:

```
java
int x = 10;
System.out.println(x++);
```

- 1) 11
- 2)+ 10
- 3) Ошибка компиляции
- 4) 9

11. Какой тип данных используется для хранения символов Unicode?

- 1) `byte`
- 2)+ `char`
- 3) `String`
- 4) `int`

12. Какой оператор используется для логического И?

- 1) `|`
- 2) `||`
- 3) `&`
- 4)+ `&&`

13. Какой оператор используется для логического ИЛИ?

- 1) `&`
- 2) `&&`
- 3) `|`
- 4)+ `||`

14. Что выведет на экран следующий код:

```
java
boolean b = true;
System.out.println(!b);
```

- 1) `true`
- 2)+ `false`
- 3) `1`
- 4) `0`

15. Какое ключевое слово используется для определения блока кода, выполняемого при выполнении условия `if` является ложным?

- 1) `then`
- 2)+ `else`
- 3) `endif`
- 4) `otherwise`

## Тема 2. Объектно-ориентированное программирование (ООП).

1. Что такое объект в объектно-ориентированном программировании?

- 1) Шаблон для создания классов
- 2)+ Экземпляр класса
- 3) Набор методов
- 4) Набор переменных

2. Что такое инкапсуляция в ООП?

- 1) Способность объекта наследовать свойства другого объекта
- 2)+ Скрытие данных и методов внутри класса
- 3) Возможность выполнения одного и того же метода разными способами
- 4) Определение интерфейса для взаимодействия с объектом

3. Какой модификатор доступа делает член класса доступным только внутри самого класса?

- 1) `public`
- 2) `protected`
- 3)+ `private`
- 4) `default (package-private)`

4. Какой ключевое слово используется для наследования класса в Java?

- 1) `implements`
- 2)+ `extends`
- 3) `inherits`
- 4) `super`

5. Какой класс является суперклассом для всех классов в Java по умолчанию?

- 1) `String`
- 2)+ `Object`
- 3) `Class`
- 4) `System`

6. Что такое полиморфизм в ООП?

- 1) Способность объекта наследовать свойства другого объекта
- 2) Скрытие данных и методов внутри класса
- 3)+ Возможность выполнения одного и того же метода разными способами
- 4) Определение интерфейса для взаимодействия с объектом

7. Какой ключевое слово используется для вызова конструктора суперкласса?

- 1) `this()`
- 2)+ `super()`
- 3) `parent()`
- 4) `base()`

8. Что такое интерфейс в Java?

- 1) Класс, который не может иметь экземпляров
- 2)+ Набор абстрактных методов
- 3) Класс, который может быть унаследован только один раз
- 4) Класс, содержащий только статические методы

9. Какой ключевое слово используется для реализации интерфейса классом?

- 1) `extends`
- 2)+ `implements`
- 3) `inherits`
- 4) `uses`

10. Что такое абстрактный класс в Java?

- 1) Класс, который не имеет полей
- 2) Класс, который не может иметь подклассов
- 3)+ Класс, который может содержать абстрактные методы
- 4) Класс, все методы которого являются статическими

11. Какой ключевое слово используется для объявления абстрактного метода?

- 1) `static`
- 2) `final`
- 3)+ `abstract`
- 4) `private`

12. Что такое перегрузка метода (method overloading)?

- 1) Определение метода с тем же именем, что и в суперклассе
- 2)+ Определение нескольких методов с одним и тем же именем, но разными параметрами
- 3) Изменение реализации метода в подклассе
- 4) Удаление метода из класса

13. Что такое переопределение метода (method overriding)?

- 1) Определение нескольких методов с одним и тем же именем
- 2)+ Определение метода с тем же именем и сигнатурой, что и в суперклассе, с другой реализацией
- 3) Создание нового метода в суперклассе
- 4) Вызов метода суперкласса

14. Какой ключевое слово используется для обращения к члену текущего объекта?

- 1) `super`
- 2)+ `this`
- 3) `self`
- 4) `current`

15. Может ли абстрактный класс иметь конструкторы?

- 1) Нет
- 2)+ Да
- 3) Только статические конструкторы
- 4) Только конструкторы без параметров

### Тема 3. Обработка исключений.

1. Что такое исключение (exception) в Java?

- 1) Синтаксическая ошибка в коде
- 2)+ Ошибка времени выполнения, которая может быть обработана
- 3) Логическая ошибка в программе
- 4) Предупреждение компилятора

2. Какой блок используется для обработки исключений в Java?

- 1) if-else
- 2) for-loop
- 3)+ try-catch
- 4) switch-case

3. Какой класс является базовым классом для всех проверяемых исключений?

- 1) RuntimeException
- 2) Error
- 3) Throwable
- 4)+ Exception

4. Какой блок кода выполняется всегда, независимо от того, было ли выброшено исключение в блоке try ?

- 1) catch
- 2) try
- 3)+ finally
- 4) throws

5. Какой класс является базовым классом для всех непроверяемых исключений (runtime exceptions)?

- 1) Exception
- 2) Throwable
- 3)+ RuntimeException
- 4) Error

6. Какое ключевое слово используется для явного выброса исключения?

- 1) catch
- 2) try
- 3)+ throw
- 4) throws

7. В каком порядке должны располагаться блоки catch при обработке иерархии исключений?

- 1) От наиболее общего к наиболее специфичному
- 2)+ От наиболее специфичного к наиболее общему
- 3) Порядок не имеет значения
- 4) Только один блок catch

8. Что означает объявление throws в сигнатуре метода?

- 1) Метод обрабатывает все возможные исключения
- 2)+ Метод может выбросить указанное исключение, и вызывающий код должен его обработать
- 3) Метод не может выбросить никаких исключений
- 4) Метод создает новое исключение



9. Что представляет собой блок `finally` ?

- 1) Блок, содержащий код, который может выбросить исключение
- 2) Блок, содержащий код для обработки конкретного типа исключения
- 3)+ Блок, содержащий код, который выполняется всегда, независимо от наличия исключений
- 4) Блок, содержащий код, выполняемый только если исключение не было выброшено

10. Может ли один блок `try` иметь несколько блоков `catch` ?

- 1) Нет
- 2)+ Да
- 3) Только если исключения одного типа
- 4) Только если исключения не связаны наследованием

11. Что произойдет, если в блоке `try` возникнет исключение, для которого нет соответствующего блока `catch` ?

- 1) Программа продолжит выполнение
- 2) Исключение будет перехвачено блоком `finally`
- 3)+ Программа аварийно завершится, и исключение будет выведено в консоль
- 4) Исключение будет проигнорировано

12. Какой класс является суперклассом для всех ошибок (errors) в Java?

- 1) `Exception`
- 2)+ `Throwable`
- 3) `RuntimeException`
- 4) `Object`

13. В чем основное отличие между `Exception` и `Error` ?

- 1) `Exception` - это непроверяемое исключение, а `Error` - проверяемое
- 2)+ `Exception` обычно указывает на проблемы, которые программа может попытаться обработать, а `Error` - на более серьезные проблемы, которые программа обычно не может восстановить
- 3) `Error` нужно обязательно обрабатывать в блоке `try-catch`
- 4) `Exception` возникает только во время компиляции

14. Может ли метод перехватить исключение и затем снова его выбросить?

- 1) Нет
- 2)+ Да
- 3) Только если это проверяемое исключение
- 4) Только если это непроверяемое исключение

## Тема 4. Ввод-вывод (I/O).

1. Какой класс используется для чтения данных из файла в Java?

- 1) `FileWriter`
- 2) `PrintStream`
- 3)+ `FileInputStream`
- 4) `Scanner`

2. Какой класс используется для записи данных в файл в Java?

- 1) `FileReader`
- 2) `InputStream`
- 3)+ `FileOutputStream`
- 4) `BufferedReader`

3. Что такое сериализация объекта в Java?

- 1) Процесс создания нового экземпляра класса
- 2)+ Процесс преобразования объекта в последовательность байтов для хранения или передачи
- 3) Процесс сравнения двух объектов
- 4) Процесс вызова метода объекта

4. Какой интерфейс необходимо реализовать классу, чтобы его объекты можно было сериализовать?

- 1) `Runnable`
- 2) `Comparable`
- 3)+ `Serializable`
- 4) `Cloneable`

5. Какой класс используется для буферизованного чтения символьных данных из файла?

- 1) `FileReader`
- 2) `InputStreamReader`
- 3)+ `BufferedReader`
- 4) `Scanner`

6. Какой класс используется для буферизованной записи символьных данных в файл?

- 1) `FileWriter`
- 2) `OutputStreamWriter`
- 3)+ `BufferedWriter`
- 4) `PrintStream`

7. Какой интерфейс представляет собой источник входных байтов?

- 1) `Writer`
- 2) `Reader`
- 3)+ `InputStream`
- 4) `OutputStream`

8. Какой интерфейс представляет собой место назначения для выходных байтов?

- 1) `Writer`
- 2) `Reader`
- 3) `InputStream`
- 4)+ `OutputStream`

9. Какой класс используется для работы с файлами и директориями в файловой системе?

- 1) Files
- 2) Path
- 3)+ File
- 4) Directory

10. Какой класс используется для чтения примитивных типов данных из потока ввода?

- 1) InputStream
- 2) Reader
- 3)+ DataInputStream
- 4) BufferedReader

11. Какой класс используется для записи примитивных типов данных в поток вывода?

- 1) OutputStream
- 2) Writer
- 3)+ DataOutputStream
- 4) BufferedWriter

12. Какой класс предоставляет удобные методы для чтения данных различных типов, разделенных пробелами или другими разделителями?

- 1) BufferedReader
- 2) InputStreamReader
- 3)+ Scanner
- 4) FileReader

13. Какой класс используется для записи форматированного вывода в поток?

- 1) OutputStreamWriter
- 2) BufferedWriter
- 3)+ PrintStream
- 4) FileWriter

14. Что такое NIO (New Input/Output) в Java?

- 1) Старый API для ввода-вывода
- 2)+ Неблокирующий API для ввода-вывода, предоставляющий более гибкие возможности
- 3) API для работы с сетью
- 4) API для работы с файлами

## Тема 5. Коллекции (Collections) и Generic'и.

1. Что такое Generic'и в Java?

- 1) Механизм для создания анонимных внутренних классов
- 2)+ Механизм для обеспечения типобезопасности коллекций
- 3) Способ обработки исключений
- 4) Средство для работы с файлами

2. Какой символ используется для объявления типа-параметра в Generic'ах?

- 1) `[]`
- 2) `{}`
- 3)+ `<>`
- 4) `()`

3. Какой интерфейс является корневым интерфейсом для всех коллекций в Java?

- 1) `List`
- 2) `Set`
- 3) `Map`
- 4)+ `Collection`

4. Какой класс реализует интерфейс `List` и обеспечивает упорядоченное хранение элементов с возможностью дублирования?

- 1) `HashSet`
- 2) `TreeSet`
- 3)+ `ArrayList`
- 4) `HashMap`

5. Какой класс реализует интерфейс `Set` и обеспечивает хранение уникальных элементов без определенного порядка?

- 1) `ArrayList`
- 2) `LinkedList`
- 3)+ `HashSet`
- 4) `TreeMap`

6. Какой интерфейс представляет собой структуру данных "ключ-значение"?

- 1) `List`
- 2) `Set`
- 3)+ `Map`
- 4) `Collection`

7. Какой класс реализует интерфейс `Map` и обеспечивает хранение элементов в виде пар "ключ-значение" без определенного порядка ключей?

- 1) `TreeMap`
- 2) `LinkedHashMap`
- 3)+ `HashMap`
- 4) `Hashtable`

8. Какой цикл используется для итерации по элементам коллекции в Java (начиная с Java 5)?

- 1) `for (int i = 0; ...)`
- 2) `while (iterator.hasNext())`
- 3)+ `for-each loop`
- 4) `do-while`

9. Какой класс реализует интерфейс `List` и обеспечивает быстрое добавление и удаление элементов в середине списка?

- 1) `ArrayList`
- 2) `Vector`
- 3)+ `LinkedList`
- 4) `Stack`

10. Какой класс реализует интерфейс `Set` и обеспечивает хранение элементов в отсортированном порядке?

- 1) `HashSet`
- 2) `LinkedHashSet`
- 3)+ `TreeSet`
- 4) `PriorityQueue`

11. Какой класс реализует интерфейс `Map` и обеспечивает хранение элементов в порядке их вставки?

- 1) `HashMap`
- 2) `TreeMap`
- 3)+ `LinkedHashMap`
- 4) `Hashtable`

12. Какой интерфейс представляет собой упорядоченную коллекцию, поддерживающую дубликаты?

- 1) `Set`
- 2) `Map`
- 3)+ `List`
- 4) `Queue`

13. Какой интерфейс представляет собой коллекцию, предназначенную для хранения элементов перед их обработкой (обычно FIFO)?

- 1) `Set`
- 2) `List`
- 3)+ `Queue`
- 4) `Deque`

14. Какой интерфейс расширяет интерфейс `Queue` и представляет собой двустороннюю очередь (можно добавлять и удалять элементы с обоих концов)?

- 1) `List`
- 2) `Set`
- 3)+ `Deque`
- 4) `Stack`

15. Какой класс реализует интерфейс `Deque` и обеспечивает быстрое добавление и удаление элементов с обоих концов?

- 1) `ArrayList`
- 2)+ `LinkedList`
- 3) `PriorityQueue`
- 4) `TreeSet`

## Тема 6. Многопоточность (Multithreading).

1. Что такое многопоточность (multithreading)?

- 1) Выполнение нескольких программ одновременно
- 2)+ Выполнение нескольких частей одной программы одновременно
- 3) Разделение программы на несколько классов
- 4) Использование нескольких процессоров для выполнения программы

2. Какой метод используется для запуска потока ( `thread` ) в Java?

- 1) `run()`
- 2)+ `start()`
- 3) `execute()`
- 4) `begin()`

3. Какой метод используется для приостановки выполнения текущего потока на определенное время?

- 1) `yield()`
- 2) `interrupt()`
- 3)+ `sleep()`
- 4) `wait()`

4. Какой интерфейс необходимо реализовать классу, чтобы его экземпляры могли выполняться в отдельном потоке?

- 1) `Callable`
- 2) `Thread`
- 3)+ `Runnable`
- 4) `Executor`

5. Какой метод используется для ожидания завершения другого потока?

- 1) `start()`
- 2) `resume()`
- 3)+ `join()`
- 4) `stop()`

6. Что такое состояние гонки (race condition)?

- 1)+ Ситуация, когда несколько потоков пытаются получить доступ к одному и тому же ресурсу одновременно, и результат зависит от порядка их выполнения
- 2) Ситуация, когда поток находится в состоянии ожидания

generated at [geetest.ru](https://geetest.ru)

## Table of Contents

|                                                          |    |
|----------------------------------------------------------|----|
| Тема 1. Основы Java.                                     | 2  |
| Тема 2. Объектно-ориентированное программирование (ООП). | 5  |
| Тема 3. Обработка исключений.                            | 7  |
| Тема 4. Ввод-вывод (I/O).                                | 9  |
| Тема 5. Коллекции (Collections) и Generic'и.             | 11 |
| Тема 6. Многопоточность (Multithreading).                | 13 |