# **Тест**

1. Какие определения для абстрактных классов в Java являются верными?

* Нельзя создать экземпляр абстрактного класса
* В абстрактных классах нельзя определять поля
* Абстрактный класс не может быть объявлен с модификатором final
* В абстрактном классе нельзя объявлять методы с модификатором default
* Абстрактный класс не может наследовать другие классы

1. Какие из перечисленных типов являются типами с плавающей точкой?

* long
* byte
* int
* float
* char
* double
* String

1. Оператор break

* Прекращает выполнение текущего цикла, независимо от того, выполняется ли условие его окончания.
* Прекращает выполнение текущего метода
* прекращает выполнение текущей итерации цикла
* Немедленно прекращает выполнение метода, в котором он находится.

1. Какой минимальный целочисленный тип в java?

### 

* long
* byte
* int
* short
* float

1. Какие из перечисленных объявлений массива скомпилируются без ошибок

* int a[];
* int[] b;
* int [5]c;
* int d[5];
* int[] k = new int[5];
* int[] f = new int[5]();

1. Может ли конструктор класса быть объявлен как final?

* Да
* Нет

1. Какие из приведенных слов являются ключевыми или зарезервированными в Java.

* final
* unsigned
* default
* operator
* extends

1. Вручную исключение можно сгенерировать при помощи оператора

* throw
* throws
* error
* Exception

1. Корректные вариант(варианты) переопределения методов:

* Метод с другим возвращаемым типом
* final метод
* public метод
* переопределение с расширением уровня доступа (например с protected на public)

1. Чтобы класс применил интерфейс, используется ключевое слово

* interface
* implements
* extends
* abstract

1. Методы default объявляются в

* абстрактных классах
* вложенных классах
* интерфейсах
* статических классах

1. Переменная, определенная внутри цикла имеет видимость

* внутри класса
* внутри метода
* внутри цикла
* внутри пакета

1. Цикл, объявленный след. образом является

|  |
| --- |
| for(; ;) {    } |

### 

* ошибочным
* статическим
* абстрактным
* бесконечным

1. Внутри класса определены два метода с одним именем, но разным числом параметров. Такие методы являются:

|  |
| --- |
| public boolean test(boolean variable) {  return true;  }    public boolean test(String line, boolean variable) {  return false;  } |

### 

* Предопределенными
* Перегруженными
* Ошибочными
* Абстрактными

1. Каким будет результат компиляции и выполнения следующего кода?

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | public class Main {  public static void main(String[] args) {  boolean b = !false;  System.out.println(b ? 1 : 2);  }  } |

### 

* 1
* 2
* Произойдет ошибка компиляции

1. Каким будет результат компиляции и выполнения следующего кода:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | public class Main {  public static void main(String[] args) {  byte i1 = 100;  byte i2 = 28;  byte b = i1 + i2;  System.out.println(b);  }  } |

### 

* -128
* 128
* Произойдет ошибка компиляции
* Произойдет ошибка выполнения

1. Что будет результатом компиляции и выполнения данного кода:

public class Main {

public static void main(String[] args) {

String str = "2";

switch (str) {

case "1":

System.out.println("1");

default:

System.out.println("default");

case "3":

System.out.println("3");

}

}

}

* Ошибка компиляции
* 1

default

3

* default

3

* 3
* Ошибка выполнения

1. Что будет результатом компиляции и выполнения данного кода:

import java.util.ArrayList;

public class Main {

public static void main(String[] args) {

ArrayList<Integer> list = new ArrayList<>(2);

list.add(1);

list.add(2);

list.add(3);

System.out.println(list.size());

}

}

* 2
* 3
* Ошибка компиляции
* Ошибка выполнения

1. Каким будет результат компиляции и выполнения следующего кода?

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | public class Main {  public static void main(String[] args) {  int a = 1;  System.out.println(a + a);  a = "1";  System.out.println(a + a);  }  } |

### 

* 2

2

* 2

11

* 11

11

* Произойдет ошибка компиляции

1. Каким будет результат компиляции и выполнения следующего кода:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | public class Main {  public static void main(String[] args) {  a = 1;  b = 2;  System.out.println(a == --b);  }  } |

### 

* true
* false
* 1
* Произойдет ошибка компиляции

1. Каким будет результат компиляции и выполнения следующего кода?

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | public class Runner {  public static void main(String[] args) {  int sum = 0;  for (int i = 0; i < 3; i++) {  sum += i;  }  System.out.println("Sum of " + i + " numbers is " + sum);  }  } |

### 

* Sum of 3 numbers is 6
* Возникнет ошибка компиляции
* Sum of 2 numbers is 3
* Возникнет ошибка времени выполнения
* Sum of 3 numbers is 3

1. Каким будет результат компиляции и выполнения следующего кода:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | public class Main {  public static void main(String[] args) {  int a = 1;  int b = --a;  if (b) {  System.out.println(1);  } else {  System.out.println(2);  }  }  } |

### 

* 1
* 2
* 3
* Возникнет ошибка компиляции

1. Каким будет результат компиляции и выполнения следующего кода?

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | class Ideone {  public static void main(String[] args) {  System.out.println(test(1) + test(2));  }    static int test(int a) {  System.out.print(a);  return a;  }  } |

### 

* 312
* 321
* 213
* 123

1. Каким будет результат компиляции и выполнения следующего кода:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | public class Main {  public static void main(String[] s) {  int a = 2  int b = 3  System.out.println(a + b \* b)  }  } |

### 

* 11- если расставить точки с запятой
* Произойдет ошибка компиляции
* 15
* 233

1. Каким будет результат компиляции и выполнения следующего кода:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | public class Main {  private static long l;  private static boolean b;  private static String s;  public static void main(String[] args) {  System.out.println(l);  System.out.println(b);  System.out.println(s);  }  } |

### 

* null

null

null

* 0

false

""

* 1

true

""

* 0

false

null

* Произойдет ошибка выполнения

1. Каким будет результат компиляции и выполнения следующего кода?

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | public class Main {  public static void main(String[] args) {  int a = 0;  int b = 1;  try {  System.out.println(b / a);  } catch (ArithmeticException e) {  System.out.println("ArithmeticException");  } catch (Exception e) {  System.out.println("Exception");  }  }  } |

### 

* Infinity
* Код не скомпилируется
* Exception
* ArithmeticException

1. Каким будет результат компиляции и выполнения следующего кода:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | public class Main {  public static void main(String[] args) {  System.out.println(sum(1, 2));  }  static int sum(int a, int b) {  try {  return a + b;  } finally {  return 0;  }  }  } |

### 

* Произойдет ошибка компиляции
* 3
* 0
* Произойдет ошибка выполнения

1. Каким будет результат компиляции и выполнения следующего кода?

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | class MyClass {  public static void main (String[] args) {  String s1="123";  String tmp = new Integer("3").toString();  String s2= "12" + tmp;  System.out.println(s1==s2);  }  } |

### 

* true
* false
* Произойдет ошибка компиляции
* Произойдет ошибка выполнения

1. Каким будет результат компиляции и выполнения следующего кода?

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | class MyClass {  public static void main (String[] args) {  String s1="123";  String tmp = new Integer("3").toString();  String s2= "12" + tmp;  System.out.println(s1.equals(s2));  }  } |

### 

* true
* false
* Произойдет ошибка компиляции
* Произойдет ошибка выполнения

1. Каким будет результат компиляции и выполнения следующего кода?

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | class MyClass {  public static void main (String[] args) {  String s1="123";  String s2="123";  System.out.println(s1==s2);  }  } |

### 

* true
* false
* Произойдет ошибка компиляции
* Произойдет ошибка выполнения