## 1. Какие из следующих выражения являются допустимыми идентификаторами Java? (Выберите все подходящие варианты)

**A.** A$B

**B.** \_helloWorld

**C.** true

**D.** java.lang

**E.** Public

**F.** 1980\_s

## 2. Каков вывод следующей программы?

1: public class WaterBottle {

2: private String brand;

3: private boolean empty;

4: public static void main(String[] args) {

5: WaterBottle wb = new WaterBottle();

6: System.out.print("Empty = " + wb.empty);

7: System.out.print(", Brand = " + wb.brand);

8: }

9: }

**A.** Строка 6 генерирует ошибку на этапе компиляции.

**B.** Строка 7 генерирует ошибку на этапе компиляции.

**C.** Нет вывода.

**D.** Empty = false, Brand = null

**E.** Empty = false, Brand =

**F.** Empty = null, Brand = null

## 3. Что из перечисленного верно? (Выберите все подходящие варианты)

4: short numPets = 5;

5: int numGrains = 5.6;

6: String name = "Scruffy";

7: numPets.length();

8: numGrains.length();

9: name.length();

**A**. Строка 4 генерирует ошибку на этапе компиляции.

**B**. Строка 5 генерирует ошибку на этапе компиляции.

**C**. Строка 6 генерирует ошибку на этапе компиляции.

**D**. Строка 7 генерирует ошибку на этапе компиляции.

**E**. Строка 8 генерирует ошибку на этапе компиляции.

**F**. Строка 9 генерирует ошибку на этапе компиляции r.

**G**. Код успешно скомпилируется.

## 4. Какое из нижеследующих утвержденний истинно относительно класса Snake? (Выберите все верные варианты)

1: public class Snake {

2:

3: public void shed(boolean time) {

4:

5: if (time) {

6:

7: }

8: System.out.println(result);

9:

10: }

11: }

**A**. Если втсавить строку String result = "done"; вместо строки 2, то код скомпилируется.

**B**. Если вставить строку String result = "done"; вместо строки 4, то код скомпилируется.

**C.** Если вставить строку String result = "done"; вместо строки 6, то код скомпилируется.

**D**. Если вставить строку String result = "done"; вместро строки 9, то код скомпилируется.

**E**. Ни одно из вышеперечисленных изменений не позволит скомпилировать код.

## 5. Для заданных классов, какое выражение можно подтсавить вместо строки INSERT IMPORTS HERE для того, что бы этот код мог быть скомпилирован? (Выберите все возможные варианты)

package aquarium;

public class Tank { }

package aquarium.jellies;

public class Jelly { }

package visitor;

INSERT IMPORTS HERE

public class AquariumVisitor {

public void admire(Jelly jelly) { } }

**A**. import aquarium.\*;

**B**. import aquarium.\*.Jelly;

**C**. import aquarium.jellies.Jelly;

**D**. import aquarium.jellies.\*;

**E**. import aquarium.jellies.Jelly.\*;

**F**. Ни одно из вышеперечисленных изменений не позволит скомпилировать код.

## 6. С учетом следующих классов, какое максимальное количество импортов можно безопасно удалить и при этом код все еще будет компилироваться?

package aquarium; public class Water { }

package aquarium;

import java.lang.\*;

import java.lang.System;

import aquarium.Water;

import aquarium.\*;

public class Tank {

public void print(Water water) {

System.out.println(water);

}

}

**A**. 0

**B**. 1

**C**. 2

**D**. 3

**E**. 4

**F**. Не компилируется.

## 7. С учетом представленных классов, какие из следующих фрагментов можно вставить вместо INSERT IMPORTS HERE и скомпилировать код? (Выберите все возможные варианты)

package aquarium;

public class Water {

boolean salty = false;

}

package aquarium.jellies;

public class Water {

boolean salty = true;

}

package employee;

INSERT IMPORTS HERE

public class WaterFiller {

Water water;

}

**A**. import aquarium.\*;

**B**. import aquarium.Water;

import aquarium.jellies.\*;

**C**. import aquarium.\*;

import aquarium.jellies.Water;

**D**. import aquarium.\*;

import aquarium.jellies.\*;

**E**. import aquarium.Water;

import aquarium.jellies.Water;

**F**. Ничего из вышеприведенного не позволит скомпилировать.

## 8. Что из перечисленного верно? (Выберите все возможные варианты)

**A.** Переменная экземпляра класса типа double по умолчанию null.

**B.** Переменная экземпляра класса типа int по умолчаниюn null.

**C.** Переменная экземпляра класса типа String по умолчаниюn null.

**D.** Переменная экземпляра класса типа double по умолчанию 0.0.

**E.** Переменная экземпляра класса типа int по умолчанию 0.0.

**F.** Переменная экземпляра класса типа String по умолчанию 0.0.

**G.** Ничего из вышеперечисленного не верно.

## 9. Что из перечисленного верно? (Выберите все возможные варианты)

**A**. Локальная переменная типа boolean по умолчанию null.

**B**. Локальная переменная типа float по умолчанию 0.

**C**. Локальная переменная типа Object по умолчанию null.

**D**. Локальная переменная типа boolean по умолчанию false.

**E**. Локальная переменная типа boolean по умолчанию true.

**F**. Локальная переменная типа float по умолчанию 0.0.

**G**. Ничего из вышеперечисленного не верно.

## 10. Что из перечисленного верно? (Выберите все возможные варианты)

**A**. Переменная экземпляра класса типа boolean по умолчанию false.

**B**. Переменная экземпляра класса типа boolean по умолчанию true.

**C**. Переменная экземпляра класса типа boolean по умолчанию null.

**D**. Переменная экземпляра класса типа int по умолчанию 0.

**E**. Переменная экземпляра класса типа int по умолчанию 0.0.

**F**. Переменная экземпляра класса типа int по умолчанию null.

**G**. Ничего из вышеперечисленного не верно.

## 11. Дан класс в файле /my/directory/named/A/Bird.java:

INSERT CODE HERE

public class Bird { }

## Чем может быть заменена строка INSERT CODE HERE если компиляция выполняется в директории /my/directory? (Выберите все возможные варианты)

**A**. package my.directory.named.a;

**B**. package my.directory.named.A;

**C**. package named.a;

**D**. package named.A;

**E.** package a;

**F.** package A;

**G**. Не комплилируется.

## 12. Какие из следующих строк пройдут компиляцию? (Выберите все возможные варианты)

**A**. int i1 = 1\_234;

**B**. double d1 = 1\_234\_.0;

**C**. double d2 = 1\_234.\_0;

**D**. double d3 = 1\_234.0\_;

**E**. double d4 = 1\_234.0;

**F**. Ничего из выше перечисленного.

## 13. Дан класс Price, что может быть подставлено на место строки INSERT CODE HERE для успешной компиляции класса? (Выберите все возможные варианты)

public class Price {

public void admission() {

INSERT CODE HERE

System.out.println(amount);

}

}

**A.** int amount = 9L;

**B**. int amount = 0b101;

**C**. int amount = 0xE;

**D**. double amount = 0xE;

**E**. double amount = 1\_2\_.0\_0;

**F**. int amount = 1\_2\_;

**G**. None of the above.

## 14. Какие из следующих утверждений правильные? (Выберите все возможные варианты)

public class Bunny {

public static void main(String[] args) {

Bunny bun = new Bunny();

}

}

**A**. Bunny – это класс.

**B**. bun – это класс.

**C**. main – это класс.

**D**. Bunny – это ссылка на объект.

**E**. bun – это ссылка на объект.

**F**. main – это ссылка на объект.

**G**. Ничего из выше перечисленного.

## 15. В какой последовательности необходимо разместить строки кода для успешной компиляции класса? (Выберите все возможные варианты)

A: class Rabbit {}

B: import java.util.\*;

C: package animals;

**A.** A, B, C

**B.** B, C, A

**C.** C, B, A

**D.** B, A

**E.** C, A

**F.** A, C

**G.** A, B

## 16. Есть класс Rabbit. Какие из следующих утверждений верны? (Выберите все возможные варианты)

1: public class Rabbit {

2: public static void main(String[] args) {

3: Rabbit one = new Rabbit();

4: Rabbit two = new Rabbit();

5: Rabbit three = one;

6: one = null;

7: Rabbit four = one;

8: three = null;

9: two = null;

10: two = new Rabbit();

11: System.gc();

12: } }

A. Сборщик мусора сразу освободит память на строке 6 выделенную для объекта Rabbit созданного на строке 3.

B. Сборщик мусора сразу освободит память на строке 8 выделенную для объекта Rabbit созданного на строке 3.

C. Сборщик мусора сразу освободит память на строке 12 выделенную для объекта Rabbit созданного на строке 3.

D. Сборщик мусора сразу освободит память на строке 9 выделенную для объекта Rabbit созданного на строке 4.

E. Сборщик мусора сразу освободит память на строке 11 выделенную для объекта Rabbit созданного на строке 4.

F. Сборщик мусора сразу освободит память на строке 12 выделенную для объекта Rabbit созданного на строке 4.

## 17. Какие из следующих утверждений верны в отношении нижеследующего кода? (Выберите все возможные варианты)

public class Bear {

protected void finalize() {

System.out.println("Roar!");

}

public static void main(String[] args) {

Bear bear = new Bear();

bear = null;

System.gc();

}

}

**A**. Метод finalize() точно будет вызван.

**B**. Метод finalize() может выть вызван, а может быть и не вызван.

**C**. Метод finalize() точно не будет вызван.

**D**. Сборщик мусора точно будет запущен.

**E**. Сборщик мусора может быть запущен, а может быть и не запущен.

**F**. Сборщик мусора точно не запустится.

**G**. Код не скомпилируется.

## 18. Каков вывод следующего кода?

1: public class Salmon {

2: int count;

3: public void Salmon() {

4: count = 4;

5: }

6: public static void main(String[] args) {

7: Salmon s = new Salmon();

8: System.out.println(s.count);

9: } }

**A**. 0

**B**. 4

**C**. Ошибка компиляции на строке 3.

**D**. Ошибка компиляции на строке 4.

**E**. Ошибка компиляции на строке 7.

**F**. Ошибка компиляции на строке 8.

## 19. Какие из следующих утвержедний верны? (Выберите все возможные варианты)

A. В Java есть возможность перерузки операторов.

B. Код Java скомпилированный в Windows может быть запущен в Linux.

C. Java имеет указатели для прямого обращения к памяти.

D. Java – это процедурный язык программировани.

E. Java – это объектно ориентированный язык программирования.

F. Java – это функциональный язык программирования.

## 20. Какие из следующих утвержедний верны? (Выберите все возможные варианты)

A. javac компилирует файл .class в файл .java.

B. javac компилирует файл .java в файл .bytecode.

C. javac компилирует файл .java в файл .class.

D. Java принимает имя класса как параметр.

E. Java принимает имя файла .bytecode как параметр.

F. Java принимает имя класса .class как параметр.

## 21. Какие из следующих операторов Java можно использовать с булевыми переменными? (Выберите все возможные варианты)

**A**. ==

**B**. +

**C**. --

**D**. !

**E**. %

**F**. <=

## 22. Какой тип данных (или типы) необходимо подставить на место прочерка для успешной компиляции следующего фрагмента кода? (Выберите все возможные варианты)

byte x = 5;

byte y = 10;

\_\_\_\_\_ z = x + y;

**A**. int

**B**. long

**C**. boolean

**D**. double

**E**. short

**F**. byte

## 23. Каков результат работы следующего приложения?

1: public class CompareValues {

2: public static void main(String[] args) {

3: int x = 0;

4: while(x++ < 10) {}

5: String message = x > 10 ? "Greater than" : false;

6: System.out.println(message+","+x);

7: }

8: }

**A**. Greater than,10

**B**. false,10

**C**. Greater than,11

**D**. false,11

**E**. Этот код не скомпилируется из-за ошибки на строке 4.

**F**. Этот код не скомпилируется из-за ошибки на строке 5.

## 24. Что необходимо изменить? (Выберите все возможные варианты)

3: long x = 10;

4: int y = 2 \* x;

**A**. Ничего не нужно изменять; код скомпилируется.

**B**. Привести значение переменной x на строке 4 к int.

**C**. Изменить тип данных переменной x на строке 3 на short.

**D**. Привести 2 \* x на строке 4 к int.

**E**. Изменить тип данных переменной y на строке 4 к short.

**F** Изменить тип данных переменной y на строке 4 к long.