

1. Каков результат следующего кода?

```
int number = 90;
switch (number) {
    case 50:
        System.out.println("50");
    case 50:
        System.out.println("50");
    case 140:
        System.out.println("140");
    default:
        System.out.println("default");
}
```

2. Какой интерфейс предоставляет возможность хранить объекты в виде пар ключ-значение?

3. Должен ли существовать объект внешнего класса для того, чтобы создать объект обычного внутреннего класса (regular inner)?

4. Корректно ли переопределить private метод и сделать его protected?

5. Каков результат?

```
enum Vegetable {
    POTATO, TOMATO, CUCUMBER;
}
...
Vegetable p = Vegetable.POTATO;
switch (p) {
    case TOMATO:
        System.out.print("tomato ");
    case POTATO:
        System.out.print("potato ");
    case CUCUMBER:
        System.out.print("cucumber ");
    default:
        System.out.println("any");
}
```

6.

```
class Alien {  
    String invade(short ships) { return "a few"; }  
    String invade(short... ships) { return "many"; }  
}  
class Defender {  
    public static void main(String [] args) {  
        System.out.println(new Alien().invade(7));  
    }  
}
```

Каков результат?

7. Какой класс коллекции позволяет увеличивать или уменьшать его размер и обеспечивает индексированный доступ к его элементам?

8.

```
public class Bottom2 extends Top {  
    public Bottom2(String s) {  
        System.out.print("D");  
    }  
  
    public static void main(String[] args) {  
        new Bottom2("C");  
        System.out.println(" ");  
    }  
}  
  
class Top {  
    public Top(String s) {  
        System.out.print("B");  
    }  
}
```

Каков результат?

9.

Каков результат выполнения кода?

```
class SomeClass {  
    public static void main(String[] args) {  
        boolean flag = true;  
        int value;  
        if (flag) {  
            value = flag ? 1 : 2;  
        } else {  
            value = flag ? 3 : 4;  
        }  
        System.out.println(value);  
    }  
}
```

10.

Каков результат?

```
StringBuilder numberRange = new StringBuilder("123");  
numberRange.append("456")  
            .reverse()  
            .insert(5, "---");  
System.out.println(numberRange);
```

11.

Что является допустимым объявлением в интерфейсе?

a `public static int START = 1;` ☐

b `protected int START = 1;` ☐

c `final void go(int start);` ☐

d `transient int START = 1;` ☐

12.

» Что будет выводом следующей программы?

```
class Example {  
    public static void main(String[] args) {  
        int y = 0;  
        label:  
        if (y < 2) {  
            System.out.print("y is " + y);  
            y++;  
            continue label;  
        }  
    }  
}
```

13.

Может ли идентификатор начинаться с цифры?

14.

Дано:

```
3. class Chicks {
4.     synchronized void yack(long id) {
5.         for(int x = 1; x < 3; x++) {
6.             System.out.print(id + " ");
7.             Thread.yield();
8.         }
9.     }
10. }
11. public class ChicksYack implements Runnable {
12.     Chicks c;
13.     public static void main(String[] args) {
14.         new ChicksYack().go();
15.     }
16.     void go() {
17.         c = new Chicks();
18.         new Thread(new ChicksYack()).start();
19.         new Thread(new ChicksYack()).start();
20.     }
21.     public void run() {
22.         c.yack(Thread.currentThread().getId());
23.     }
24. }
```

15.

Каков результат выполнения следующего кода?

```
public class MyArrays {  
    public static void main(String[] args) {  
        byte array[][] = {{1, 2, 3, 4}, {1, 2, 3}};  
        System.out.println(array[1].length + " " + array.length);  
    }  
}
```

16.

Каким будет вывод программы?

```
class MyProgram {  
    public static void main(String[] args) {  
        String a = "ABCD";  
        String b = a.toLowerCase();  
        b.replace('a', 'd');  
        b.replace('b', 'c');  
        System.out.println(b);  
    }  
}
```

17.

Опишите основные принципы ООП

18.

Опишите контраст между equals и hashCode