

## Тест 5

5.5/6 points (graded)

1. Чи буде зкомпільовано наступний клас?

```
class Test
{
    int test(int i, int d)
    {
        return 0;
    }

    static int test(int i, double d)
    {
        return 0;
    }

    static double test(double i, double d)
    {
        return 0;
    }
}
```

☒ Так

☐ Ні



2. Скільки конструкторів матиме наступний клас?

Виберіть найбільш точну відповідь

```
public class A
{
}
```

☒ Один

☐ Жодного

☐ Такий клас викличе помилку компіляції



3. Чи буде зкомпільовано наступний інтерфейс?

```
interface A
{
    private int i;
}
```

☐ Так

☒ Ні



4. Вкажіть причину помилки компіляції для наступного коду. Виберіть найбільш точну відповідь

```
interface A
{
    void test();
}

class B implements A
{
    void test()
    {
        System.out.println("Ok");
    }
}
```

☐ метод test() в класі B повинен бути static

☐ метод test() в класі B повинен бути final

☒ метод test() в класі B повинен бути public

☐ тут не буде помилки компіляції



5. Клас було оголошено як абстрактний без жодного абстрактного методу. Чи буде він зкомпільований? Виберіть найбільш точну відповідь

```
abstract class Demo  
{  
}
```

☒ так

☐ так, за умови що буде додано абстрактний конструктор

☐ ні

☐ тільки розробник може вирішувати чи буде щось зкомпільовано



6. Наступний код викликає помилку при компіляції. Вкажіть причину.

```
abstract class A  
{  
    abstract int demo();  
}  
  
class B extends A  
{  
}
```

☐ Клас B не містить конструктора

☐ Клас А не містить абстрактного конструктора

☒ Клас В не реалізував всі абстрактні методи батьківського класу



7. Чи буде зкомпільовано наступний код? Виберіть найбільш точну відповідь

```
interface A
{
    int variable = 1;
}

class B implements A
{
    void methodB()
    {
        variable = 2;
    }
}
```

☐ так

☐ ні, оскільки variable вже оголошено

☒ ні, оскільки variable вже оголошено і це константа



8. Чи буде зкомпільовано наступний код?

```
class A
{
}

interface Demo extends A
{
}
```



так



ні, оскільки A не реалізує інший інтерфейс



ні, оскільки інтерфейс не може розширювати клас



9. Що буде виведено на екран в результаті виконання наступного коду?

```
abstract class Calculator
{
    abstract int sum(int a, int b);
}

public class Demo
{
    public static void main(String[] args)
    {
        int result = new Calculator()
        {
            int sum(int a, int b)
            {
                return a+b;
            }
        }.sum(2, 3);
        System.out.println(result);
    }
}
```



Нічого

☐ Помилка часу виконання

☒ 5

☐ 6



10. Що буде виведено на екран в результаті виконання наступного коду?

```
interface Iface1
{
    String a = "A";
    String methodA();
}

interface Iface2 extends Iface1
{
    String b = "B";
    String methodB();
}

class Demo implements Iface1, Iface2
{
    public String methodA()
    {
        return a+b;
    }

    public String methodB()
    {
        return b+a;
    }
}

public class Main
{
    public static void main(String[] args)
    {
        Demo demo = new Demo();
        System.out.println(demo.methodA());
        System.out.println(demo.methodB());
    }
}
```

☒ AB☐ BA☐ BA☐ AB

☐ Помилка часу виконання

☐ Помилка компіляції



11. Що буде виведено на екран в результаті виконання наступного коду?

```
class Demo
{
    void test(String s){
        System.out.println("String");
    }

    void test(Integer i){
        System.out.println("Integer");
    }
}

public class Main
{
    public static void main(String[] args)
    {
        Demo demo = new Demo();
        demo.test(null);
    }
}
```

☒ Integer

☐ String

☐ null

☐ нічого, помилка компіляції





12. Чи дозволено викликати `this()` та `super()` в методах?

☐ так

☒ ні



Надіслати

You have used 2 of 2 attempts

---

\* Partially correct (5.5/6 points)