

Dziedziczenie - pytania

Marcin Szupke

Poniższy fragment kodu się nie skompiluje. Dlaczego?

```
class X
{
    //Class X Members
}

class Y
{
    //Class Y Members
}

class Z extends X, Y
{
    //Class Z Members
}
```

Jaki będzie rezultat działania poniższego kodu?

```
class A
{
    int i = 10;
}

class B extends A
{
    int i = 20;
}

public class MainClass
{
    public static void main(String[] args)
    {
        A a = new B();

        System.out.println(a.i);
    }
}
```

Jaki będzie rezultat działania poniższego kodu?

```
class A
{
    {
        System.out.println(1);
    }
}

class B extends A
{
    {
        System.out.println(2);
    }
}

class C extends B
{
    {
        System.out.println(3);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        C c = new C();
    }
}
```

**Czy klasa może rozszerzać
samą siebie?**

Jaki będzie rezultat działania poniższego kodu?

```
class A
{
    String s = "Class A";
}

class B extends A
{
    String s = "Class B";

    {
        System.out.println(super.s);
    }
}

class C extends B
{
    String s = "Class C";

    {
        System.out.println(super.s);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        C c = new C();

        System.out.println(c.s);
    }
}
```

Jaki będzie rezultat działania poniższego kodu?

```
class A
{
    {
        System.out.println("THIRD");
    }
}

class B extends A
{
    {
        System.out.println("SECOND");
    }
}

class C extends B
{
    {
        System.out.println("FIRST");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        C c = new C();
    }
}
```

Jaki będzie rezultat działania poniższego kodu?

```
class A
{
    public A()
    {
        System.out.println("Class A Constructor");
    }
}

class B extends A
{
    public B()
    {
        System.out.println("Class B Constructor");
    }
}

class C extends B
{
    public C()
    {
        System.out.println("Class C Constructor");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        C c = new C();
    }
}
```


Prywatne składowe klasy są dziedziczone przez klasy potomne. Prawda czy fałsz?

Poniższy fragment kodu się nie skompiluje. Dlaczego?

```
class X
{
    public X(int i)
    {
        System.out.println(1);
    }
}

class Y extends X
{
    public Y()
    {
        System.out.println(2);
    }
}
```

Poniższy fragment kodu się nie skompiluje. Dlaczego?

```
public class A
{
    public A()
    {
        System.out.println(1);

        super();

        System.out.println(2);
    }
}
```

**Jaka klasa jest klasą bazową dla
każdej innej klasy napisanej w Javie?**

Poniższy fragment kodu się nie skompiluje. Dlaczego?

```
public class A
{
    public A()
    {
        super();

        this(10);
    }

    public A(int i)
    {
        System.out.println(i);
    }
}
```

Jaki będzie rezultat działania poniższego kodu?

```
class M
{
    int i;

    public M(int i)
    {
        this.i = i--;
    }
}

class N extends M
{
    public N(int i)
    {
        super(++i);

        System.out.println(i);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        N n = new N(26);
    }
}
```

Jaki będzie rezultat działania poniższego kodu?

```
class M
{
    int i = 51;

    public M(int j)
    {
        System.out.println(i);

        this.i = j * 10;
    }
}
```

```
class N extends M
{
    public N(int j)
    {
        super(j);

        System.out.println(i);

        this.i = j * 20;
    }
}
```

```
public class MainClass
{
    public static void main(String[] args)
    {
        N n = new N(26);

        System.out.println(n.i);
    }
}
```

Poniższy fragment kodu się nie skompiluje. Dlaczego?

```
class X
{
    private int m = 48;
}

class Y extends X
{
    void methodOfY()
    {
        System.out.println(m);
    }
}
```


Czy poniższy fragment kodu się skompiluje? Jeśli tak, to jaki będzie rezultat jego działania?

```
class A
{
    void A()
    {
        System.out.println(1);
    }
}

class B extends A
{
    void B()
    {
        A();
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        new B().B();
    }
}
```

**Jak zapobiec dziedziczeniu składowej
lub metody przez klasę potomną?**

Konstruktory są dziedziczone przez klasy potomne. Prawda czy fałsz?

Jaki będzie rezultat działania poniższego kodu?

```
class A
{
    int[] a = new int[5];

    {
        a[0] = 10;
    }
}

public class MainClass extends A
{
    {
        a = new int[5];
    }

    {
        System.out.println(a[0]);
    }

    public static void main(String[] args)
    {
        MainClass main = new MainClass();
    }
}
```

Co się stanie jeżeli klasa bazowa oraz klasa potomna będą miały zadeklarowaną składową o tej samej nazwie?