Object Oriented Programming - Handout 7

Konrad Wojda, 9307820244

Code

```java
class MyException_A extends Exception {
    public String toString() {
        return "I am a MyException_A object";
    }
}

class MyException_B extends MyException_A {
    public String toString() {
        return "I am a MyException_B object";
    }
}

class MyException_C extends MyException_A {
    public String toString() {
        return "I am a MyException_C object";
    }
}

public class JavaProgram {
    public static void fun() throws MyException_A {     // purple code
        try {
            int i;
            System.out.println("FA");
            if (true)
                throw new MyException_B();      // green code
            if (true)
                throw new MyException_B();
            System.out.println("BA");
        } catch (MyException_B e) {
            System.out.println("P1 " + e.toString());
        }
        catch (MyException_A e) {
            System.out.println("P2 " + e.toString());
            throw e;
        }
        /* red code (insert finally clause here) */
        System.out.println("BC");
    }

    public static void main(String[] args) {
        try {
            System.out.println("A");
            fun();
            if (true)
                throw new MyException_B();
            System.out.println("B");
        } catch (MyException_B e) {
            System.out.println("P3 " + e.toString());
        } catch (MyException_A e) {
            System.out.println("P4 " + e.toString());
        } catch (Exception e) {
            System.out.println("P5 " + e.toString());
        }
        System.out.println("F");
    }
}
```

Exercise 1a

Output:

```
A
FA
P1 I am a MyException_B object
BC
P3 I am a MyException_B object
F
```

Exercise 1b

1

Output:

```
A
FA
P2 I am a MyException_A object
P4 I am a MyException_A object
F
```

2

Output:

```
A
FA
P2 I am a MyException_C object
P4 I am a MyException_C object
```

```
F
```

3

Output:

```
A
FA
P5 java.lang.Exception
F
```

4

Output:

```
A
FA
P5 java.lang.ArithmeticException: / by zero
F
```

Exercise 2a

0 (1a)

Output:

```
A
FA
P1 I am a MyException_B object
In finally
BC
P3 I am a MyException_B object
F
```

1 (1b)

Output:

```
A
FA
P2 I am a MyException_A object
In finally
P4 I am a MyException_A object
F
```

2 (1b)

Output:

```
A
FA
P2 I am a MyException_C object
In finally
P4 I am a MyException_C object
F
```

3 (1b)

Output:

```
A
FA
In finally
P5 java.lang.Exception
F
```

4 (1b)

Output:

```
A
FA
In finally
P5 java.lang.ArithmeticException: / by zero
F
```

Exercise 2b

0 (1a)

Output:

```
A
FA
P1 I am a MyException_B object
In finally
P5 java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 2
F
```

1 (1b)

Output:

```
A
FA
P2 I am a MyException_A object
In finally
P5 java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 2
```

```
F
```

2 (1b)

Output:

```
A
FA
P2 I am a MyException_C object
In finally
P5 java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 2
F
```

3 (1b)

Output:

```
A
FA
In finally
P5 java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 2
F
```

4 (1b)

Output:

```
A
FA
In finally
P5 java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 2
F
```

Answer to the question

In the `finally` block from exercise 2b there is a problem - this code will throw an `ArrayIndexOutOfBoundsException` every time. In this case it can override the exception from the try or catch block at runtime (in older Java versions). While modern Java (7+) preserves the original exception by marking it as suppressed, many developers and tools ignore suppressed exceptions, which can lead to loss of important error context. To avoid this, risky code in `finally` block should have its own try-catch block.