

# 1 Exception

Exceptions are mechanisms to signalize i react on unusual situations. Exception is an object derived from class Exceptions and implements Throwable.

```
public class ExceptionExample {  
  
    public static void main(String args[]){  
        int a;  
        int b = 2;  
        int c = 0;  
        a = b / c;  
        System.out.println(a);  
    }  
}
```

## 2 Reacting on exceptions

### 2.1 try...catch block

```
public class ExceptionExample {  
  
    public static void main(String args[]){  
        int a;  
        int b = 2;  
        int c = 0;  
        try {  
            a = b / c;  
            System.out.println(a);  
        } catch(ArithmeticException e){  
            System.out.println("Dzielenie przez zero");  
        }  
    }  
}
```

### 2.2 Marking methods

```
public class ExceptionExample {  
  
    public void metoda() throws ArithmeticException{  
        int a;  
        int b = 2;  
        int c = 0;  
        a = b / c;  
    }  
}
```

```

        System.out.println(a);
    }

    public static void main(String args[]){
        ExceptionExample ee = new ExceptionExample();
        try {
            ee.metoda();
        } catch(ArithmeticException e){
            System.out.println("Dzielenie przez zero");
        }
    }
}

```

### 3 Creating custom exceptions

```

class MyException extends Exception{
}

public class ExceptionExample {

    public static void main(String args[]){
        try {
            throw new MyException();
        } catch (MyException e) {
            e.printStackTrace();
        } catch (Exception e){
            e.printStackTrace();
        }
    }
}

```

### 4 Tasks to complete

1. Create a structure of exceptions that derive one after another (3 levels)
2. Throw exceptions
3. Use try...catch block to react on exceptions
4. How to use exceptions if there is no try...catch in method?
5. Check the difference when changing order of catch block
6. Determine what will happen if the exception is not catch at all
7. What is happening with the result of the method after exception was thrown
8. Determine the difference between Exceptions and RuntimeExceptions

9. Introduce code that will be executed no matter the exception is thrown or not
10. Determine the meaning of information displayed by `printStackTrace()` method
11. Rethrow caught exception