Programowanie w języku Java - Adam Krechowicz

1 Wyjątek

Wyjątki są mechanizmami służącymi sygnalizowaniu i reagowaniu na sytuacje wyjątkowe. Wyjątek jest obiektem, którego klasa wywodzi się z klas Exception i 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 Reagowanie na wyjątki

2.1 Blok try...catch

```
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 Oznaczanie metod wyrzucających wyjątki

```
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 Tworzenie własnych wyjątków

```
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 Zadania do wykonania

1. Stworzyć strukturę kilku wyjątków dziedziczących po sobie
2. Wyrzucić wyjątki
3. Zabezpieczyć program przed wyjątkami za pomocą bloków try...catch
4. Jak wykorzystać wyjątki w metodzie, która nie zawiera bloku try...catch?
5. Sprawdzić różnice podczas zmiany kolejności bloków catch
6. Ustalić co dzieje się z nieprzechwyconymi wyjątkami wyrzuconymi w metodach
7. Co się dzieje z rezultatem metody podczas wyrzucenia wyjątku?

8. Ustalić różnicę pomiędzy zwykłymi wyjątkami a wyjątkami *RuntimeException*
9. Wprowadzić do programu fragment kodu, który zostanie wykonany bez względu na fakt wyrzucenia wyjątku
10. Ustalić znaczenie komunikatu wyświetlanego przez metodę *printStackTrace()*
11. Powtórnie wyrzucić złapany wyjątek