

WYJĄTKI

Ćwiczenia 8

ZADANIE 1.

Proszę stworzyć metodę która wygeneruje wyjątek *NullPointerException*.

Metodę należy wywołać w oddzielnej klasie z w metodzie *main()*, oraz przechwycić za pomocą bloku *try/catch*.

W klauzuli *catch* proszę wypisać jakiś komunikat dla użytkownika oraz wywołać na obiekcie reprezentującym wyjątek metody *printStackTrace()* oraz *toString()*.

ZADANIE 2.

Proszę stworzyć metodę która w zależności od tego jaką wartość otrzyma jako argument wygeneruje *ArithmeticException* lub *ArrayIndexOutOfBoundsException* lub nie wygeneruje żadnego wyjątku. Każdy z powyższych wyjątków ma być obsługiwany oddzielną klauzulą *catch* , dodatkowo umieszczamy w kodzie klauzulę *finally* która tylko nas informuje że się wykonała.

Całość zostaje uruchomiona w oddzielnej klasie z metodą *main()* .

ZADANIE 3.

Proszę stworzyć klasę *Account* która będzie miała pola: *owner*, *balance* (typu *int*) oraz *accountNumber*.

Klasa ta implementuje również konstruktor składający się ze wszystkich parametrów, gettery i settery oraz metodę symulującą wykonanie przelewu z możliwością wyrzucenia wyjątku jeśli przelew będzie większy niż wartość środków na koncie (zdefiniowana podczas tworzenia obiektu).

Wyjątek to typ który stworzą Państwo sami np. o nazwie *NotEnoughMoneyException*.

W klasie z metodą *main()* stworzą Państwo obiekt klasy *Account* oraz zasymulujecie wyrzucenie wyjątku przez wykonanie metody symulującej przelew z wartością większą niż wartość dostępnych środków. Dodatkowo proszę o dodanie *catcha* który wyłapie wszystkie inne wyjątki oraz klauzuli *finally* która zwróci nam aktualny stan konta.

ZADANIE 4.

Proszę stworzyć trzy nowe typy wyjątków. Napisać klasę z metodą która w czasie swoje wykonania może zgłosić każdy z nich (pojedynczo).

W metodzie *main()* proszę wywołać tę metodę w bloku *try*, z jedną klauzulą *catch* która będzie mogła przechwycić każdy z utworzonych przez Państwa wyjątków zależności od parametru przekazanego do metody.

ZADANIE 5.

W kolejnym przykładzie proszę o skorzystanie z tego że wyjątki można rejestrować (logować) przy użyciu mechanizmu z biblioteki *java.util.logging* (konieczne jest dodanie importu). Statyczna metoda *Logger.getLogger()* zwraca obiekt rejestratora (*Logger*) skojarzony z argumentami typu String (zwykle reprezentuję on nazwę pakietu i klasy, który dotyczy rejestrowanie) wypisującym informacje na standardowym wyjściu diagnostycznym (*System.err*). Najprostszym sposobem pisania do logu jest wywołania odpowiadającej poziomowi rejestrowania komunikatu – użyjemy do tego metody *severe()*. W komunikacie proszę ująć stos wywołań który doprowadził do wyjątku. Problem polega na tym, że *printStackTrace()* nie zwraca obiektu String. Aby otrzymać taki obiekt, musimy użyć przeciążonej wersji *printStackTrace()* przyjmującej argument w postaci obiektu klasa *java.io.PrintWriter* (będzie na następny wykładzie). Jeśli konstruktor *PrintWriter* wypełnimy argumentem *java.io.StringWriter*, otrzymamy wartość, z której metody będzie można wyciągnąć obiekt String.

ciąg dalszy na następnym slajdzie...

ZADANIE 5.

Przechodząc do polecenia, proszę o utworzenie dwóch klas wyjątków, korzystając z powyższego opisu, moja pierwsza klasa wygląda tak:

```
6 class MyException1 extends Exception {  
7     private static Logger logger = Logger.getLogger("LoggingException");  
8     public MyException1() {  
9         StringWriter trace = new StringWriter();  
10        printStackTrace(new PrintWriter(trace));  
11        logger.severe(trace.toString());  
12    }  
13 }
```

W oddzielnej klasie proszę o utworzenie dwie statyczne metody, pierwsza deklaruje że może wyrzucić pierwszy i drugi nasz wyjątek ale wyrzuca tylko pierwszy, a druga deklaruje i wyrzuca tylko drugi.

W metodzie *main()* deklarujemy dwie klauzule *try* dla każdej z metod oraz jednego *catcha* który wyłapuje wszystkie wyjątki.

DZIĘKUJĘ

Więcej na:

www.vistula.edu.pl



Akademia Finansów i Biznesu Vistula
ul. Stokłosy 3
02-787 Warszawa
(obok stacji metro Stokłosy)