Zajęcie laboratoryjne 2

1. Składnia Javy. Obsługa wyjątków. Operacje IO

Wyjątek – mechanizm kontroli przepływu służący do obsługi zdarzeń wyjątkowych (w szczególności błędów)

Klasa Exception:

```
java.lang.Object
\--java.lang.Throwable
\--java.lang.Exception
```

Obsługa:

– przechwytywanie wyjątków w odwrotnej kolejności do hierarchii klas (od najbardziej szczegółowych do najbardziej ogólnych)

```
try {
//kod generujacy wyjatek
} catch (Exception e) {
//obsluga wyjatkow klasy Exception
} catch (Throwable t) {
//obsluga wyjatkow klasy Throwable
} finally {
//kod zawsze wykonywalny
}
```

✔ sekcja **finally** wykonywana jest bez względu na to, czy wyjątek został obsłużony, czy też nie

Generowanie:

– generowanie nowego wyjątku

```
if (o == null)
throw new NullPointerException();
```

– ponowne wyrzucanie wyjątku

```
try {
...
} catch (Exception e) {
...
throw e; //ponowne wyrzucenie
}
```

Tworzenie:

– rozszerzanie interfejsu klasy bazowej

```
class MyException extends Exception { }
```

Specyfikacja:

– specyfikacja wyrzucanych wyjątków, których obsługą zajmie się kod wywołujący metodę

```
void method() throws IOException { }
```

✔ metody przeciążone – powinny wyrzucać takie same wyjątki

✔ metody implementujące – nie muszą wyrzucać wyjątków specyfikowanych przez metodę abstrakcyjną

✔ konstruktory – mogą dodawać nowe wyjątki

Wyjątek czasu wykonania nie wymaga specyfikacji (jest zawsze wyrzucany)

```
java.lang.Object
\--java.lang.Throwable
|--java.lang.Exception
| \--java.lang.RuntimeException
\--java.lang.Error
```

Klasa Throwable:

.getMessage() - zwraca szczegółowy komunikat

.printStackTrace() - wypisuje komunikat i ślad stosu wywołań (sekwencję wywołań metod)

Zadania:

1. Wykonać aplikację odporną na błędy oraz:
– utworzyć własny wyjątek przechowujący opis oraz numer kodu błędu
– zdefiniować konstruktor oraz metody wyrzucające powyższy wyjątek
– zastosować blok obsługi wyjątków try/catch/finally
2. Napisać klasę:

A) Pokoj zawierającą wymiary pomieszczenia (dlugosc,szerokosc,wysokosc). Klasa ma posiadać konstruktor oraz metody wyświetlające objetość, powierzchnie ścian, powierzchnie podłogi, koszt malowania (przy zadanym koszcie 1m2), koszt podłogi (przy zadanym koszcie 1m2). W programie głównym stworzyć kilka obiektów klasy Pokój (np.salon,kuchnia,sypialnia) i na podstawie przykładowych danych wyświetlić wszystkie informacje.

B) Samochod zawierającą opis samochoda (moc, cena, kolor). Klasa ma posiadać konstruktor oraz metody wyświetlające koszt utrzymania (przy zadanym koszcie na 1 rok), zużycie paliwa (przy zadanym zyżyciu na 100 km). W programie głównym stworzyć kilka obiektów klasy Samochod (np.VW Polo,VW CC, Dodge Durango) i na podstawie przykładowych danych wyświetlić wszystkie informacje.

C) Wakacje zawierającą opis wakacji (kraj, miejscowość, hotel, termin, wyżywienie itp). Klasa ma posiadać konstruktor oraz metody wyświetlające cenę (przy zadanym pokoju i terminie). W programie głównym stworzyć kilka obiektów klasy Wakacje (np.Grecja,Włochy,Hiszpania) i na podstawie przykładowych danych wyświetlić wszystkie informacje.

3. Napisać program wczytujący z klawiatury wartosci dla obiektów z zadania 2 oraz zapisujący je w pliku tekstowym. Obsłużyć wszelkie możliwe wyjątki programu.

2. Tworzenie aplikacji dla Google Application Engine

Ćwiczenia

1. Spróbować wszystkie przykłady z powodu GWT-komponentów na stronie internetowej http://java2s.com/Code/Java/GWT/CatalogGWT.htm
2. Stworzyć GAE- aplikację, co zabezpiecza dokładną rejestrację użytkownika (wprowadzenie informacji o użytkowniku z walidacją oraz zachowywaniem w składowisku Google Datastore) i rozwijać ją na platformie Google Apps Engine (na podstawie projektu na stronie -

https://cloud.google.com/appengine/docs/java/gettingstarted/creating-guestbook