

1. Obsługa wyjątków.

Dla obsługi błędów najlepszym momentem jest ich wykrycie na etapie kompilacji, jeszcze przed uruchomieniem programu. Nie ma jednak możliwości wykrycia wszystkich błędów podczas kompilacji. Błędy które nie są obsługiwane na tym etapie muszą być obsługane przez specjalny mechanizm pozwalający na ich identyfikację i zwrócenie odpowiedniej informacji do odbiorcy.

Wyjątek jest więc mechanizmem kontroli przepływu służący do obsługi zdarzeń wyjątkowych w szczególności błędów.

Klasa Exception:

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

Składnia Wyjątków

Wyjątki należy przechwytywać w odwrotnej kolejności do hierarchii klas (od najbardziej szczegółowych do najbardziej ogólnych). Natomiast sekcja **finally** wykonywana jest bez względu na to, czy wyjątek został obsłużony czy też nie. Z użyciem **finally** wiąże się jeden mankament. Kod zawarty w tym bloku w żadnym wypadku nie powinien zgłaszać wyjątku. Jeżeli wyjątek zgłaszany w bloku **try** nie został obsłużony, a w **finally** również zgłaszany jest wyjątek, to wyjątek z bloku **try** zaginie.

```
try {
    //kod generujący wyjątek
} catch (Exception e) {
    //obsługa wyjątków klasy Exception
} catch (Throwable t) {
    //obsługa wyjątków klasy Throwable
} finally {
    //kod zawsze wykonywalny
}
```

Generowanie Wyjątków:

- generowanie nowego wyjątku

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

- ponowne wyrzucenie wyjątku

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

Tworzenie:

- rozszerzanie interfejsu klasy
`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)

Zadanie:

Wkonać aplikację odporną na błędy oraz:

- utworzyć własny wyjątek przechowujący opis oraz kod błędu
- zdefiniować konstruktor oraz metody wyrzucające powyższy wyjątek
- zastosować blok obsługi wyjątków try/catch/finally