

Programowanie Java  
Anna Gogolińska  
Zestaw 5

Zad 1. Napisać klasę *LapanieWyjatkov* zawierającą metodę *obliczOdwrotnosc(-)*, która jako parametr pobiera tablicę liczb. Następnie, każdy element modyfikujemy obliczając jego odwrotność, czyli dzieląc 1 przez niego samego. Należy dodać w tej metodzie obsługę wyjątków: *Exception*, *NullPointerException*, *ArithmeticException*. Zastanów się i uzasadnij w jakiej kolejności powinny zostać obsłużone wymienione wyżej wyjątki.

Zad 2. Zmodyfikuj klasę utworzoną w zadaniu pierwszym tak, aby przekazywała do kontekstu wyżej wyjątek *NullPointerException*. Należy tak poprawić program, aby nadal obsługiwał wszystkie wyjątki wypisując na ekran nazwę wyjątku oraz stan stosu wywołań metod programu.

Zad 3. Utworzyć nową klasę wyjątku o nazwie *MojWyjatek*. Następnie napisać w osobnej klasie *TestMojegoWyjatku* metodę *losuj()*, która generuje z prawdopodobieństwem 0.25 (losowanie wartości od 0 do 1) ten wyjątek do kontekstu wyżej. W metodzie *main()* głównej klasy programu należy przechwycić ten wyjątek oraz niezależnie od jego wystąpienia zapewnić wykonanie części kodu wypisującego na ekran wylosowaną w metodzie *losuj()* wartość (wykorzystać sekcję *finally*).

Zad 4. Napisać program realizujący wyszukiwanie pierwiastka metodą równego podziału (połowienia). W tym celu należy utworzyć klasę *SzukajPierwiastka* zawierającą metody:

- *f(-)* obliczająca wartość określonej funkcji (np.  $x^3 - 2x^2 - x + 2$ ).
- *testujPrzedzial(-,-)* metoda sprawdzająca czy przedział spełnia założenia istnienia pierwiastka. W tym celu należy utworzyć wyjątek *ZlyPrzedzial* generowany w przypadku, gdy znak funkcji na końcach przedziału jest taki sam.
- Metodę *szukaj(-,-)* pobierającą lewy i prawy zakres przedziału. Metoda ta ma wyznaczyć pierwiastek ograniczając przedział poszukiwania go, z dokładnością do epsilon (zadana stała). Dla zadanego przedziału nowe granice przedziału mają być wyznaczane za pomocą przechwytywania wyjątku generowanego z metody *testujPrzedzial(-,-)*. W głównej pętli metody *szukaj(-,-)* zamiast instrukcji warunkowej *if* wykorzystujemy wyjątek określający która część przedziału jest prawidłowa.

Zadania samodzielne

Zad 5.

Zaimplementuj własną klasę *ListaObiektow*, która będzie zawierała następujące metody:

- *pobierzObiekt(-)*, która zwraca obiekt znajdujący się na pozycji o podanym

- jako argument numerze,
- *pierwszyObiekt()* zwracającą pierwszy obiekt znajdujący się na liście,
  - *indeksObiektu(-)* szukającą i zwracającą pozycję na której znajduje się podany jako parametr obiekt,
  - *dodajObiekt(-)*, która dodaje podany jako argument obiekt do listy,
  - *usunObiekt(-)*, która usuwa obiekt z listy, który znajduje się na podanej jako parametr pozycji.

Dodatkowo należy zaimplementować własne wyjątki: *ZlyIndeksException*, *BrakObiektuException*, *PustaListaException*, które należy generować w przypadku, gdy odpowiednio:

- podana pozycja pobieranego lub usuwanego obiektu jest mniejsza niż 0 lub większa od liczby obiektów na liście,
- próbujemy odczytać lub usunąć obiekt z pustej listy,
- szukamy indeksu obiektu, który nie znajduje się aktualnie na liście.

Zastanów się na hierarchią tych trzech wyjątków np. dlaczego klasa *PustaListaException* może dziedziczyć z *BrakObiektuException*. Następnie spróbować tak zmodyfikować klasę, aby możliwe było wykorzystanie jej jako jednokierunkową kolejkę.