Programowanie aplikacji w Java

Maciej Gowin

Zjazd 2 - zadania dodatkowe

Linki

Opis

https://maciejgowin.github.io/wsb-java/

Kod źródłowy przykładów oraz zadań

https://github.com/MaciejGowin/wsb-programowanie-aplikacji-java

Zadania dodatkowe

Wszystkie zadanie dodatkowe zostały stworzone z myślą utrwalenia materiału przerobionego na zjeździe. Chociaż mogą one zostać rozwiązane na wiele sposobów, zostały skonstruowane w sposób umożliwiający ukończenie ich jedynie przy użyciu poznanych już technik.

Programowanie: zadanie dodatkowe 06

Zdefiniuj statyczną metodę `void reverse(int[] items)` odwracającą kolejność elementów w tablicy. Metoda ta nie powinna tworzyć tablicy pomocniczej. Wszystkie operacje powinny być wykonane na tablicy przekazanej jako parametr metody.

Przetestuje działanie.

Programowanie: zadanie dodatkowe 07

Zdefiniuj statyczne metody obliczające silnię liczby naturalnej N:

- `int factorial(int n)` przy użyciu podejścia iteracyjnego,
- `int factorialRecursive(int n)` przy użyciu podejścia rekurencyjnego.

Przetestuje działanie.

Programowanie: zadanie dodatkowe 08

Zaimplementuj stos `DoubleStack` przechowujący liczby typu `Double`. Stos powinien realizować metody:

- `void push(Double value)` - dodającą element do stosu,
- `Double pop()` - pobierającą element ze stosu,
- `Double peek()` - sprawdzającą ostatni element na stosie,
- `String toString()` - zwracającą tekstową reprezentację stosu w formacie `[1.0, 2.0, 3.0]`.

Przetestuje działanie.

Programowanie: zadanie dodatkowe 09

Zdefiniuj statyczną metodę `void selectionSort(int[] items)` implementującą sortowanie przez wybieranie (ang. selection sort).

Przetestuje działanie.

Programowanie: zadanie dodatkowe 10

Zdefiniuj statyczną metodę `int[][] reorder(int[][] items)` zmieniającą porządek elementów w tablicach gdzie podtablice tablicy wynikowj będą składać się kolejno z n-tych elementów podtablic tablicy wejściowej o ile takowe istnieją.

Dla przykładu:

`[[1, 2, 3], [1, 2] , [1]]` -> `[[1, 1, 1], [2, 2], [3]]`
`[[4, 5], [8, 7, 1]]` -> `[[4, 8], [5, 7], [1]]`
`[[1, 2, 3]]` -> `[[1], [2], [3]]`

Przetestuje działanie.

Programowanie: zadanie dodatkowe 11 (trudne)

Zaimplementuj binarne drzewo przeszukiwań (BST, ang. binary search tree) przechowujące wartości typu `Integer`. Posłuż się opisem działania pod adresem: https://pl.wikipedia.org/wiki/Binarne_drzewo_poszukiwań.

Implementacja powinna definiować następujące metody:

- `Integer getMinValue()` - zwracającą minimalną wartość w drzewie lub `null` dla braku wartości,
- `boolean contains(Integer searchedValue)` - sprawdzającą, czy dana wartość istnieje w drzewie,
- `void add(final Integer value)` - dodającą wartość do drzewa,
- `void delete(Integer value)` - usuwającą wartość z drzewa.

Przetestuje działanie.

Programowanie: zadanie dodatkowe 12

Zaimplementuj program pobierający słowa zadane przez użytkownika oraz zliczający ilość wystąpień danego słowa.

Do pobierania słów użyj metody `Scanner.next()`.

Przykładowe użycie programu.

```
Podaj słowo (lub q aby przerwać): car
Podaj słowo (lub q aby przerwać): bus
Podaj słowo (lub q aby przerwać): car
Podaj słowo (lub q aby przerwać): q
Wystąpienia słów: [{car: 2}, {bus: 1}]
```