

# Zaawansowane Techniki Programowania

## Zad 4 Plate

### Klasa AppClient

Główna klasa aplikacji

#### Metody:

- main

#### Opis :

Funkcja podczas uruchamiania przyjmuje parametry datasource oraz table. Inicjuje moduły EJB, następnie wywołuje funkcję odpowiedzialną za pobranie danych z bazy, uruchamia główny algorytm po czym wypisuje wynik.

#### Dane wejściowe:

- String[] args - dane wejściowe podawane podczas uruchamiania

#### Dane wyjściowe:

- void

- loadData

#### Opis :

Funkcja nawiązuje połączenie z bazą danych, następnie pobiera parametry x i y z tabeli po czym zapisuje je w listach modułu Plate.

#### Dane wejściowe:

- DataSource ds - źródło bazy danych
- String table - nazwa tabeli

#### Dane wyjściowe:

- void

# Klasa Plate

Klasa implementująca interfejs IPlateRemote

## Pola:

- ArrayList<Double> x - lista kosztów cięcia materiału wzdłuż danego wiersza
- ArrayList<Double> y - lista kosztów cięcia materiału wzdłuż danej kolumny

## Metody:

- addX

Opis :

Dodanie parametru do listy x

Dane wejściowe:

- double x

Dane wyjściowe:

- void

- addY

Opis :

Dodanie parametru do listy y

Dane wejściowe:

- double y

Dane wyjściowe:

- void

- getXList - pobieranie listy x
- getYList - pobranie listy y
- Plate - konstruktor

# Klasa Cost

Klasa implementująca interfejs ICostRemote

## Metody:

- calculateMinimalCutCost

Opis :

Funkcja realizuje algorytm odnalezienia najmniejszego kosztu pocięcia

Dane wejściowe:

- ArrayList<Double> x - lista kosztów cięcia materiału wzdłuż danego wiersza
- ArrayList<Double> y - lista kosztów cięcia materiału wzdłuż danej kolumny

Dane wyjściowe:

- double - minimalny koszt pocięcia materiału na pojedyncze kawałki

# Interface IPlateRemote

Interfejs modułu Plate

## Metody:

- addX
- addY
- getXList
- getYList

# Interface ICostRemote

Interfejs modułu Cost

## Metody:

- calculateMinimalCutCost

# Opis algorytmu

Pierwszym krokiem algorytmu jest posortowanie obu listy x oraz y malejąco od najwyższych wartości do najmniejszych.

Ilość cięć wzdłuż wierszy cutX, oraz kolumn cutY zostaje ustawione na 1. Parametr i oraz j odpowiadające za numer elementów z listy x oraz y zostają ustawione na 0.

Następnie w pętli następuje przeszukanie obu list jednocześnie. W każdej iteracji najdroższego cięcia z listy x pomnożonego przez ilość nacięć wzdłuż kolumn (cutY), a cięcia z listy y pomnożonego wzdłuż wierszy (cutX). Wyrażenie o większej wartości zostaje dodane do ostatecznego wyniku, następuje inkrementacja parametru i oraz cutX jeśli cięcie wzdłuż wierszy było kosztowniejsze bądź j oraz cutY jeśli wzdłuż kolumn. Pętla trwa tak długo, aż zostaną wyczerpane wszystkie wartości z listy x bądź y.

Następnie w pętli zostaną wykonane wszystkie pozostałe cięcia z listy która zawiera jeszcze elementy i dodane do ostatecznego wyniku. Jeśli listy x oraz y miały ten sam rozmiar, krok ten jest pominięty.