Praca domowa 7 – plate

Przemysław Kleszcz

Zadanie

Cel pracy domowej: Obliczyć minimalny koszt cięcia przedstawionej w zadaniu tafli materiału na pojedyncze elementy.

Organizacja strukturalna

Projekt zawiera pięć plików: **AppClient.java, Plate.java, IplateRemote.java, Cost.java, ICostRemote.java**

AppClient.java:
public class AppClient
- public static void main(String[] args)
- private static void getData(Connection conn, String table)
- private static void prepareConnections()

Cost.java:
public class Cost
- public double calculateMinCost()
- private void deleteGreaterElement(IPlateRemote plate)
- private void prepareConnections()

Plate.java:
public class Plate
- public void reInitialize()
- public void addCost(double x, double y)
- public double searchForMaxX()
- public double searchForMaxY()
- public void removeX(double min)
- public void removeY(double min)
- public boolean isEmptyX()
- public boolean isEmptyY()
- private void remove(List<Double> list, double min)
- private double searchForMax(List<Double> list)

ICostRemote oraz IPlateRemote to interfejsy modelujące odpowiedzialność klas Cost i Plate.

void main(String[] args) - Przyjmuje jako parametr tablicę argumentów linii komend. Punkt wejścia programu.

getData(Connection conn, String table) - Przyjmuje jako parametry połączenie do źródła danych oraz nazwę tabeli. Pozyskuje dane z tabeli i przekazuje do dalszego komponentu.

prepareConnections() - Inicjalizacja zmiennych będących referencjami do komponentów Cost i Plate.

calculateMinCost() - Zwraca minimalny koszt cięcia.

deleteGreaterElement() - Wyszukuje maksymalne elementy z list kosztów X oraz Y oraz usuwa z listy element o wyższej wartości.

prepareConnections() - Inicjalizacja zmiennej będącej referencją do komponentu Plate.

reInitialize() - Inicjalizuje na nowo pola klasy, zeruje stan.

addCost(double x, double y) - Przyjmuje jako parametry koszty x i y. Dodaje nowe elementy do listy kosztów.

searchForMaxX() - Zwraca maksymalny element z listy X.

searchForMaxY() - Zwraca maksymalny element z listy Y.

removeX(**double min**) - Przyjmuje jako parametr wartość z listy. Usuwa wartość z listy X.

removeY(double min) - Przyjmuje jako parametr wartość listy. Usuwa wartość z listy Y.

isEmptyX() - Zwraca true jeśli lista X jest pusta, w przeciwnym wypadku false.

isEmptyY() - Zwraca true jeśli lista Y jest pusta, w przeciwnym wypadku false.

remove(List<Double> list, double min) - Przyjmuje jako parametry listę oraz wartość listy. Usuwa wartość z listy przekazanej parametrem.

searchForMax(List<Double> list) - Przyjmuję listę jako parametr. Wyszukuje maksymalny element z listy przekazanej parametrem.

Opis mechanizmu

Program jest zrealizowany w postaci aplikacji klienckiej + EJB. Do programu przekazywane są dwa parametry: <datasource> oraz <table>. <datasource> to nazwa JNDI na podstawie, której pobierane jest źródło danych (poprzez zastosowanie lookup). W dalszej kolejności, korzystając z źródła danych oraz tabeli o nazwie przekazanej parametrem, pozyskiwane są dane poszczególnych kosztów. Dane są umieszczane w strukturze obliczeniowej poprzez przekazanie ich do komponentu Plate, który pobierany jest za pomocą stosownej nazwy JNDI. W następnej kolejności uruchamiana jest

funkcjonalność obliczania minimalnego kosztu komponentu Cost. Wynik działania metody jest wynikiem działania programu i jest przekazywany na standardowe wyjście.

Proces podziału tafli materiału na poszczególne pojedyncze elementy przy uzyskaniu minimalnego kosztu jest realizowany w następujący sposób. W strukturach agregujących koszty wzdłuż linii poziomej i pionowej należy wyszukać maksymalne wartości. Jeśli jedna bądź druga lista nie posiadają już elementów, to za wartość maksymalną liczby należy podstawić -1. W następnym etapie należy wybrać element większy z dwóch wybranych elementów maksymalnych. Następnie należy usunąć większy element z zawierającej go listy i powiększyć sumę kosztów cięć o koszt cięcia wzdłuż bieżącej linii razy liczba dokonanych cięć wzdłuż drugiej osi. Powyższy proces należy powtarzać aż do momentu kiedy obie listy cięć nie będą zawierać elementów. Uzyskana suma kosztów jest wynikiem działania programu. Złożoność obliczeniowa: n2.