



SPRAWOZDANIE PLATFORMY PROGRAMISTYCZNE .NET I JAVA – LAB5



POLITECHNIKA WROCŁAWSKA
WYDZIAŁ INFORMATYKI I TELEKOMUNIKACJI
INFORMATYCZNE SYSTEMY AUTOMATYKI

MICHAŁ WYRZYKOWSKI
INDEKS 264228

1. Opis ogólny kodu

Program przedstawia klasyczne rozwiązanie problemu plecakowego (knapsack problem), w którym celem jest maksymalizacja wartości przedmiotów umieszczonych w plecaku o ograniczonej pojemności. Program składa się z kilku klas: Problem, Result, Item i Main.

2. Klasy i ich funkcjonalność

Klasa Item

Klasa Item reprezentuje pojedynczy przedmiot z dwoma atrybutami:

- value - wartość przedmiotu,
- weight - waga przedmiotu.

Klasa Problem

Klasa Problem generuje listę przedmiotów (Item) oraz definiuje pojemność plecaka. Zawiera metody:

- Konstruktor Problem(int n, int seed, int lowerBound, int upperBound, int capacity) - tworzy listę przedmiotów z losowo przypisanymi wagami i wartościami w zadanym zakresie, korzystając z ziarna losowości (seed) dla powtarzalności wyników.
- List<Item> getItems() - zwraca listę przedmiotów.
- int getCapacity() - zwraca pojemność plecaka.
- String toString() - zwraca tekstową reprezentację wszystkich przedmiotów.
- Result solve() - rozwiązuje problem plecakowy, wykorzystując heurystykę opartą na stosunku wartości do wagi przedmiotu.

Klasa Result

Klasa Result przechowuje wyniki rozwiązania problemu plecakowego:

- items - lista wybranych przedmiotów,
- totalValue - łączna wartość wybranych przedmiotów,
- totalWeight - łączna waga wybranych przedmiotów.
- Zawiera metodę toString(), która zwraca tekstową reprezentację wyniku.

Klasa Main

Klasa Main jest punktem startowym programu, odpowiedzialnym za interakcję z użytkownikiem.

Program:

- Pobiera od użytkownika liczbę przedmiotów, ziarno losowości oraz pojemność plecaka.
- Tworzy instancję klasy Problem, generując listę przedmiotów i określając pojemność plecaka.
- Wyświetla wygenerowane przedmioty.
- Rozwiązuje problem plecakowy, wyświetlając wybrane przedmioty wraz z ich łączną wartością i wagą.

3. Podsumowanie testów jednostkowych i sposób ich działania

Opis ogólny

Testy jednostkowe w klasie ProblemTest służą do weryfikacji poprawności działania klasy Problem i jej metody solve(). Testy zostały napisane przy użyciu frameworka JUnit 5. Każdy test sprawdza określony scenariusz, aby upewnić się, że implementacja działa zgodnie z oczekiwaniami.

Testy jednostkowe

- testGenerateItems()

Ten test sprawdza, czy konstruktor klasy Problem poprawnie generuje listę przedmiotów:

- Tworzy obiekt Problem z 10 przedmiotami, używając ziarna losowości 1234 i przedziału wartości oraz wag od 1 do 10.
- Sprawdza, czy liczba wygenerowanych przedmiotów wynosi 10.
- Dla każdego przedmiotu sprawdza, czy jego waga i wartość mieszczą się w zadanym przedziale.

- testSolveWithItemsThatFit()

Ten test weryfikuje poprawność działania metody solve() w sytuacji, gdy istnieje możliwość umieszczenia wszystkich przedmiotów w plecaku:

- Tworzy obiekt Problem z 10 przedmiotami, z pojemnością plecaka wynoszącą 50.
- Rozwiązuje problem plecakowy.
- Sprawdza, czy łączna waga wybranych przedmiotów nie przekracza pojemności plecaka.
- Sprawdza, czy łączna wartość wybranych przedmiotów jest większa od zera.

- testSolveWithNoItemsThatFit()

Ten test sprawdza działanie metody solve() w sytuacji, gdy żaden z przedmiotów nie mieści się w plecaku:

- Tworzy obiekt Problem z 10 przedmiotami, z wartościami i wagami w przedziale od 20 do 30 oraz z pojemnością plecaka wynoszącą 5.
- Rozwiązuje problem plecakowy.
- Sprawdza, czy łączna waga wybranych przedmiotów wynosi 0.
- Sprawdza, czy łączna wartość wybranych przedmiotów wynosi 0.

- testSolveSpecificInstance()

Ten test sprawdza działanie metody solve() dla konkretnego zestawu przedmiotów:

- Tworzy obiekt Problem z 3 przedmiotami, z pojemnością plecaka wynoszącą 5.
- Usuwa wygenerowane losowo przedmioty i dodaje własne przedmioty:
- Przedmiot o wartości 10 i wadze 1,
- Przedmiot o wartości 20 i wadze 2,
- Przedmiot o wartości 30 i wadze 3.
- Rozwiązuje problem plecakowy.
- Sprawdza, czy łączna waga wybranych przedmiotów wynosi 5.
- Sprawdza, czy łączna wartość wybranych przedmiotów wynosi 50.

4. Podsumowanie kodu GUI i sposób działania

Opis ogólny

Program KnapsackGUI to graficzny interfejs użytkownika (GUI) dla rozwiązania problemu plecakowego, który umożliwia użytkownikowi wprowadzenie liczby przedmiotów, ziarna losowości oraz pojemności plecaka, a następnie generuje przedmioty i rozwiązuje problem plecakowy, wyświetlając wyniki. Interfejs został stworzony przy użyciu biblioteki Swing.

Klasa KnapsackGUI

Atrybuty klasy:

- JPanel panel1 - główny panel zawierający wszystkie komponenty GUI.
- JTextField textFieldNumberOfItems - pole tekstowe do wprowadzenia liczby przedmiotów.
- JTextField textFieldSeed - pole tekstowe do wprowadzenia ziarna losowości.
- JTextField textFieldCapacity - pole tekstowe do wprowadzenia pojemności plecaka.
- JButton solveButton - przycisk do uruchomienia algorytmu rozwiązującego problem plecakowy.
- JTextArea textAreaGeneratedItems - pole tekstowe do wyświetlania wygenerowanych przedmiotów.
- JTextArea textAreaKnapsackSolution - pole tekstowe do wyświetlania wyniku rozwiązania problemu plecakowego.

Konstruktor KnapsackGUI() tworzy komponenty GUI i definiuje akcję do wykonania po naciśnięciu przycisku solveButton:

- Pobiera dane z pól tekstowych: liczbę przedmiotów, ziarno losowości oraz pojemność plecaka.
- Tworzy obiekt klasy Problem z wprowadzonymi danymi.
- Wyświetla wygenerowane przedmioty w textAreaGeneratedItems.
- Rozwiązuje problem plecakowy i wyświetla wynik w textAreaKnapsackSolution.
- W przypadku błędnych danych wejściowych wyświetla komunikat błędu.

Metoda createUIComponents() tworzy i konfiguruje komponenty GUI oraz dodaje je do głównego panelu panel1:

- Tworzy pola tekstowe, przycisk oraz pola tekstowe do wyświetlania wyników.
- Ustawia układ komponentów na GridBagLayout i definiuje pozycje oraz wypełnienie dla poszczególnych komponentów.
- Dodaje komponenty do głównego panelu panel1.

Metoda main() tworzy okno aplikacji, ustawia jego zawartość na główny panel panel1, ustawia domyślną operację zamknięcia okna, dopasowuje rozmiar okna do zawartości oraz wyświetla okno.

Działanie programu

- Użytkownik uruchamia program i wprowadza liczbę przedmiotów, ziarno losowości oraz pojemność plecaka.
- Po naciśnięciu przycisku "Solve", program tworzy obiekt klasy Problem i generuje losowe przedmioty.
- Wygenerowane przedmioty są wyświetlane w polu textAreaGeneratedItems.
- Program rozwiązuje problem plecakowy za pomocą metody solve() klasy Problem.
- Wynik (wybrane przedmioty, ich łączna waga i wartość) jest wyświetlany w polu textAreaKnapsackSolution.
- Jeśli użytkownik wprowadzi błędne dane, wyświetlany jest komunikat błędu.