

# Platformy Programistyczne .NET i Java

Kierunek

Informatyczne Systemy Automatyki IPS

Termin

czwartek 17<sup>05</sup> – 18<sup>45</sup>

Imię, nazwisko, numer albumu

Igor Frysiak 272548

Link do projektu

[https://github.com/IFrysiak/PP\\_lab1](https://github.com/IFrysiak/PP_lab1)



## Spis treści

1	Wstęp	1
2	Struktura projektu	1
3	Opis klas	1
3.1	Klasa Item	1
3.2	Klasa Problem	2
3.3	Klasa Result	2
3.4	Klasa Program	2
4	Działanie programu	2
5	Testy jednostkowe	2
5.1	Opis testów	2
6	Graficzny interfejs	3
7	Podsumowanie	3

## 1 Wstęp

Zadanie polegało na implementacji rozwiązania problemu plecakowego w .NET C#. Do optymalnego wyboru przedmiotów do plecaka w oparciu o stosunek wartości do wagi wykorzystany został algorytm zachłanny.

## 2 Struktura projektu

Projekt składa się z następujących plików:

- *Item.cs* – klasa reprezentująca pojedynczy przedmiot,
- *Problem.cs* – klasa generująca instancję problemu i implementująca algorytm rozwiązujący,
- *Result.cs* – klasa przechowująca wynik rozwiązania problemu,
- *Program.cs* – plik główny obsługujący wejście użytkownika i wypisanie wyników.

## 3 Opis klas

### 3.1 Klasa Item

Klasa *Item* przechowuje wartość, wagę oraz pierwotny indeks przedmiotu.

### 3.2 Klasa Problem

Klasa *Problem* generuje przedmioty z podanego *seed* oraz implementuje algorytm zachłanny. Przedmioty są sortowane malejąco według stosunku wartości do wagi, a następnie wybierane do momentu osiągnięcia limitu pojemności plecaka.

### 3.3 Klasa Result

Klasa *Result* przechowuje wybrane przedmioty oraz sumaryczną wartość i wagę rozwiązania.

### 3.4 Klasa Program

Klasa *Program* obsługuje interakcję z użytkownikiem. Użytkownik wprowadza liczbę przedmiotów, *seed* generatora oraz pojemność plecaka, po czym program oblicza rozwiązanie i wyświetla wynik.

## 4 Działanie programu

- Użytkownik podaje:
  - liczbę przedmiotów,
  - *seed* generatora.
- Program generuje losowe przedmioty o wartościach i wagach w zakresie od 1 do 10, oraz wylicza stosunek wartości do wagi.
- Użytkownik podaje pojemność plecaka.
- Program rozwiązuje problem i zwraca wynik zawierający:
  - listę wybranych przedmiotów,
  - łączną wartość przedmiotów,
  - łączną wagę przedmiotów.

## 5 Testy jednostkowe

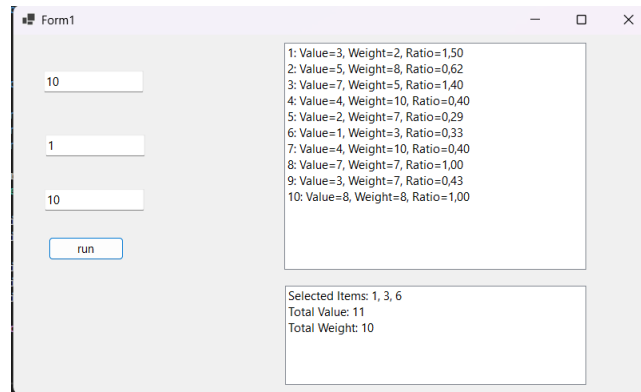
W celu sprawdzenia działania programu zaimplementowano testy jednostkowe. Klasa *UnitTest* zawiera pięć testów sprawdzających poprawność działania algorytmu rozwiązującego problem plecakowy.

### 5.1 Opis testów

- *AtLeastOneItemSelected* – sprawdza, czy jeśli istnieje co najmniej jeden przedmiot spełniający ograniczenia, to zostanie on wybrany.
- *NoItemSelected* – testuje przypadek, w którym pojemność plecaka wynosi zero, więc żaden przedmiot nie może być wybrany.
- *CorrectSolutionForInstance* – weryfikuje, czy rozwiązanie dla konkretnej instancji problemu zwraca poprawne wartości.
- *AllItemsFit* – sprawdza, czy jeśli pojemność plecaka jest wystarczająco duża, to wszystkie przedmioty zostaną wybrane.
- *OnlyOneHighestRatioItemSelected* – testuje przypadek, w którym tylko jeden przedmiot o najwyższym stosunku wartości do wagi może się zmieścić w plecaku.

## 6 Graficzny interfejs

Graficzny interfejs użytkownika został zaimplementowany w Windows Forms App. Po lewej stronie okna znajdują się trzy pola na wpisanie danych, kolejno: ilość przedmiotów, ziarno generatora, pojemność plecaka. Po wpisaniu danych, w celu uruchomienia programu należy nacisnąć przycisk "run", a w polach obok wyświetlą się wygenerowane przedmioty oraz rozwiązanie problemu.



Item	Value	Weight	Ratio
1	3	2	1.50
2	5	8	0.62
3	7	5	1.40
4	4	10	0.40
5	2	7	0.29
6	1	3	0.33
7	4	10	0.40
8	7	7	1.00
9	3	7	0.43
10	8	8	1.00

Selected Items: 1, 3, 6  
Total Value: 11  
Total Weight: 10

Wygląd GUI i przedstawienie wyników.

Gdy do pól wpisane zostaną nieprawidłowe dane, jak na przykład litera, program się nie uruchomi.

## 7 Podsumowanie

Implementacja rozwiązania problemu plecakowego została zaimplementowana oraz działa poprawnie, wykorzystując algorytm zachłanny do wyboru przedmiotów maksymalizujących wartość przy ograniczonej pojemności plecaka. Program przechodzi również wszystkie zaimplementowane testy jednostkowe. Interfejs graficzny, mimo niedokończenia z powodu braku czasu, działa poprawnie.