



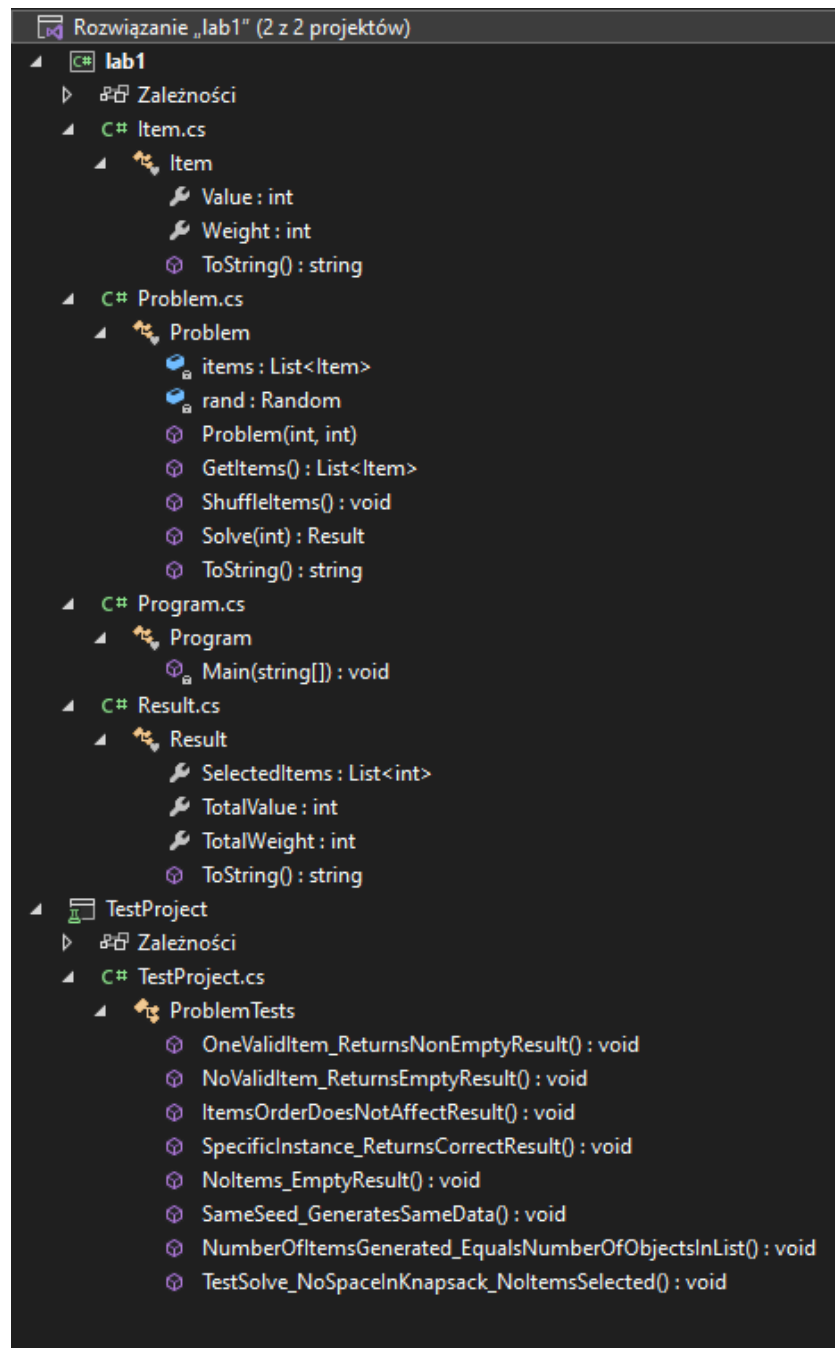
SPRAWOZDANIE PLATFORMY PROGRAMISTYCZNE .NET I JAVA - LAB1



POLITECHNIKA WROCŁAWSKA
WYDZIAŁ INFORMATYKI I TELEKOMUNIKACJI
INFORMATYCZNE SYSTEMY AUTOMATYKI

MICHAŁ WYRZYKOWSKI
INDEKS 264228

1. Kod C#, drzewo projektu aplikacji konsolowej



```

public Result Solve(int capacity)
{
    items = items.OrderByDescending(x => (double)x.Value / x.Weight).ToList();
    List<int> selectedItems = new List<int>();
    int totalValue = 0;
    int totalWeight = 0;

    foreach (var item in items)
    {
        if (totalWeight + item.Weight <= capacity)
        {
            selectedItems.Add(items.IndexOf(item) + 1);
            totalValue += item.Value;
            totalWeight += item.Weight;
        }
        else
        {
            break;
        }
    }

    return new Result
    {
        SelectedItems = selectedItems,
        TotalValue = totalValue,
        TotalWeight = totalWeight
    };
}

```

Konsola debugowania programu Microsoft Visual Studio

```

Przedmioty:
Item 1: Value: 10, Weight: 10
Item 2: Value: 8, Weight: 9
Item 3: Value: 8, Weight: 1
Item 4: Value: 1, Weight: 2
Item 5: Value: 2, Weight: 7

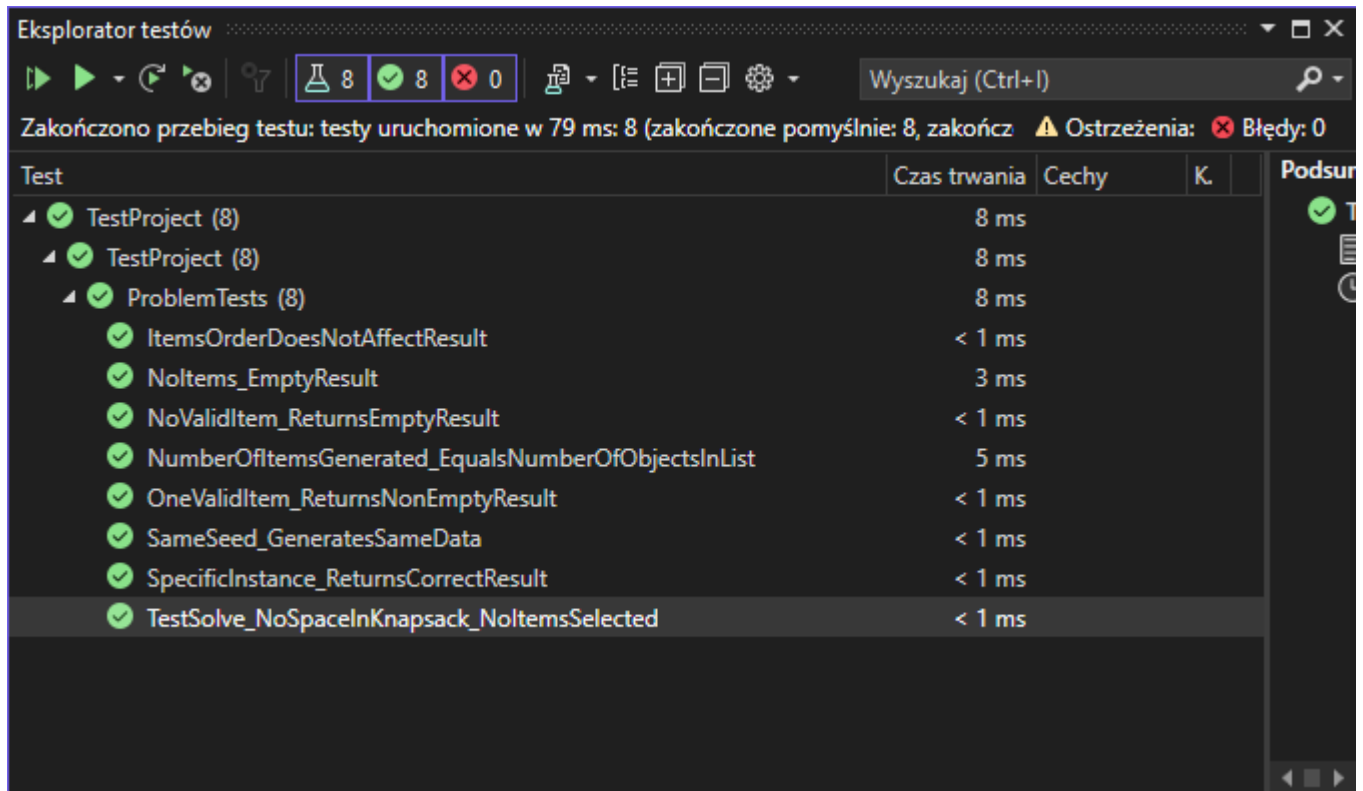
Rozwiązanie dla plecaka o pojemności 20:
Selected items: 1, 2, 3, Total value: 26, Total weight: 20

```

Kod implementuje algorytm rozwiązujący problem plecakowy dla zestawu przedmiotów o określonych wartościach i wagach, przy założeniu, że można wybrać każdy przedmiot tylko raz.

- Sortuje elementy w kolejności malejącej według wartości do wagi stosunku dla każdego elementu.
- Tworzy pustą listę wybranych przedmiotów oraz inicjuje zmienne przechowujące łączną wartość i wagę wybranych przedmiotów.
- Przechodzi przez posortowane elementy:
- Jeśli dodanie aktualnego przedmiotu nie przekroczy pojemności plecaka, to dodaje go do listy wybranych przedmiotów, dodaje jego wartość do łącznej wartości oraz dodaje jego wagę do łącznej wagi.
- Jeśli dodanie przedmiotu spowoduje przekroczenie pojemności plecaka, to przerywa iterację.
- Na koniec zwraca wynik zawierający listę indeksów wybranych przedmiotów, łączną wartość i łączną wagę wybranych przedmiotów.

2. Testy jednostkowe



The screenshot shows the 'Eksplorator testów' (Test Explorer) window in Visual Studio. At the top, a status bar indicates 'Zakończono przebieg testu: testy uruchomione w 79 ms: 8 (zakończone pomyślnie: 8, zakończ Ostrzeżenia: 0 Błędy: 0'. Below this, a table lists the test results.

Test	Czas trwania	Cechy	K.	Podsumowanie
TestProject (8)	8 ms			8/8
TestProject (8)	8 ms			8/8
ProblemTests (8)	8 ms			8/8
ItemsOrderDoesNotAffectResult	< 1 ms			✓
NoItems_EmptyResult	3 ms			✓
NoValidItem_ReturnsEmptyResult	< 1 ms			✓
NumberOfItemsGenerated_EqualsNumberOfObjectsInList	5 ms			✓
OneValidItem_ReturnsNonEmptyResult	< 1 ms			✓
SameSeed_GeneratesSameData	< 1 ms			✓
SpecificInstance_ReturnsCorrectResult	< 1 ms			✓
TestSolve_NoSpaceInKnapsack_NoItemsSelected	< 1 ms			✓

- OneValidItem_ReturnsNonEmptyResult: Sprawdza, czy rozwiązanie zawiera co najmniej jeden poprawny przedmiot.
- NoValidItem_ReturnsEmptyResult: Sprawdza, czy dla braku dostępnych przedmiotów zwracane jest puste rozwiązanie.
- ItemsOrderDoesNotAffectResult: Sprawdza, czy kolejność przedmiotów nie wpływa na ostateczne rozwiązanie.
- SpecificInstance_ReturnsCorrectResult: Sprawdza, czy dla określonego zestawu przedmiotów i pojemności plecaka zwracane jest poprawne rozwiązanie.
- NoItems_EmptyResult: Sprawdza, czy dla braku przedmiotów zwracane jest puste rozwiązanie.
- SameSeed_GeneratesSameData: Sprawdza, czy dla tego samego ziarna generującego dane, generowane są takie same zestawy przedmiotów.
- NumberOfItemsGenerated_EqualsNumberOfObjectsInList: Sprawdza, czy liczba wygenerowanych przedmiotów jest równa liczbie obiektów na liście.
- NoSpaceInKnapsack_NoItemsSelected: Sprawdza, czy dla braku miejsca w plecaku nie zostają wybrane żadne przedmioty.