Programowanie wizualne

opracował: Paweł Wojciechowski 2023/24

Lab 4

Zagadnienia do opanowania:

- delegaty
- metody anonimowe
- wyrażenia lambda
- zdarzenia
- metody rozszerzania

Legenda:
Zadania oznaczone (-) można pominąć – prezentują one często elementarne cechy danego zagadnienia.
Zadania oznaczone (*) też można pominąć ;) – to są zadania trudniejsze.
Temat: Delegaty i metody anonimowe

1. Utworzyć nowy projekt i skopiować od niego klasy Pracownik i Firma z pliku z przykładami DelegateCallback.cs. Zmodyfikować metodę ToString dla klasy Pracownik żeby wyświetlały się również zarobki pracownika.
2. Zdefiniować delegata, który pozwoli nam wyciągnąć z obiektu Firma najlepszego pracownika, przy czym nie definiujemy na razie w czym dany pracownik ma być najlepszy. Jakiego typu będą parametry wejściowe i wyjściowy? Dla ułatwienia, zakładamy, że zawsze jest 1 najlepszy pracownik w danej kategorii. Co należałoby zmienić, jeśli dopuścilibyśmy możliwość wyboru więcej niż jednego najlepszego pracownika?
   Definiujemy publiczny obiekt delegata w klasie Firma o nazwie Selector. Implementujemy metodę publiczną wg. Następującego wzoru:
   ```
   public Pracownik GetTheBest()
   ```
   W tej metodzie wywołujemy Selectora – proszę pamiętać o sprawdzeniu, czy obiekt nie jest null-em.
3. Proszę zdefiniować następujące metody wyciągające najlepszych pracowników według następujących kryteriów:
   a. Najlepiej zarabiający
   b. Najmłodszy
   c. Mający najkrótsze imię.

   (*) Można spróbować wykonać te same akcje dla metod rozszerzania zdefiniowanych dla interfejsu generycznego IEnumerable<Pracownik>
4. (-)Zdefiniować klasę `TestDelegate` a w niej delegata
   ```
   public delegate string PrintIt(string s);
   ```
5. (-)Przetestować działanie delegata z przykładu 1 dla metody obiektu i metody statycznej.
6. (-)Sprawdzić która wartość zostanie zwrócona po wywołaniu delegata przy podpięciu do niego wielu metod.
7. (-)Dodać kolejnego delegata jako metodę anonimową.
8. (-)W klasie `TestDelegate` zdefiniować prywatne pole. Czy metoda anonimowa ma do tego pola dostęp?
9. (-)Zdefiniować nowego delegata
   ```
   public delegate int Sum(params int[] args);
   ```
   Napisać metodę anonimową jako metodę obsługującą tego delegata.

   Pytania kontrolne:
   - Jak należy podać typ metody anonimowej?
   - W jakich przypadkach można pominąć parametry delegata w metodzie anonimowej?
   - Co się dzieje z modyfikatorem `params` w metodzie anonimowej?
   - Czy metoda anonimowa ma dostęp do prywatnych pól klasy?

Temat: Wyrażenia lambda

10. Utworzyć tablice `int[]` z kilkunastoma elementami. Następnie używając wyrażenia lambda (`Where`) wyświetlić elementy nieparzyste.

11. (*)Powtórzyć ćwiczenie dla delegata i własnej metody `bool Cmp(int).`

Temat: Zdarzenia

12. Zdefiniować zdarzenie `ParityCheck` wywoływane w momencie sprawdzania parzystości z zadania 7/8. Użyć standardowego `EventHandler` jako delegata.
13. Umożliwić przesłanie parametru (liczby) jako EventArgs. W tym celu należy zdefiniować klasę dziedziczącą po EventArgs i zmienić typ zdarzenia ParityCheck na EventHandler<>.

    Pytania kontrolne:
    - Zadanie 10 wykonać na dwa sposoby: nowy delegat i EventHandler<>.

Temat: Metody rozszerzania (extension methods)

15. Utworzyć metodę rozszerzania `SumOfDigits` dla typu `int`, która doda zwróci sumę cyfr w liczbie. Czy da się zrobić taką metodę generyczną?

    Definicja metody rozszerzania powinna znajdować się w klasie statycznej i powinna być statyczna o następującym nagłówku:

    ```
    public static int SumOfDigits(this int i)
    ```