

ZPR PWr – Zintegrowany Program Rozwoju Politechniki Wrocławskiej

PWr

Spotkanie 5

Aplikacje webowe na platformie .NET

Laboratorium – **Lista 5**

Wstęp.

Jak każdy język obiektowy również C# posiada wiele mechanizmów związanych z programowaniem obiektowych:

- Implementując konstruktor obiektu można wywołać na początku inny konstruktor używając słowa kluczowego **this** z listą parametrów dopasowanych do tegoż innego konstruktora.
- Tworzenie pól, metod w ramach obiektu oraz decydowaniu o ich dostępności dla użytkownika implementowanej klasy.
- Zamiast tzw. getterów i setterów, które zakładają po prostu pewien standard nazewnictwa metod do odczytu i zapisu (najczęściej prywatnych) pól, w języku C# istnieje mechanizm właściwości, czyli funkcji odczytu i zapisu, które w zapisie wyglądają jak zwykłe pola. Pozwala to na uproszczenie zapisu do bardziej naturalnego np.
`obiektX.WlasciwoscY++;`
zamiast (z pomocą getterów i setterów)
`obiektX.setPoleY(obiektX.getPoleY()+1);`
- Tworzenie pól, metod i właściwości statycznych, czyli wspólnych dla wszystkich obiektów danej klasy. Można również wytworzyć konstruktor dla tego typu elementów.
- Dla klas, które nie można rozszerzyć poprzez mechanizm dziedziczenia istnieje mechanizm ułatwiający rozszerzenie możliwości takiej klasy poprzez funkcje rozszerzające. Formalnie nie jest to rzeczywiste dodanie nowej metody do klasy, jednak kompilator potrafi zmienić zapis z operatorem kropki wyglądający jak wywołanie danej metody na rzecz referencji na obiekt na wywołanie takiej metody. Wymogiem jest , aby metoda ta była statyczna, w klasie statycznej, a pierwszy parametr posiadał słowo kluczowo **this** przed nazwą typu, a tym ten to musi być oczywiście nazwa klasy, która ma być rozszerzona.
- W języku część wybranych operatorów typu '+' itd. Może zostać przeładowana. Nie jest to tak szeroki wachlarz możliwości jaki jest np. w języku C++, jednak dzięki temu działanie np. instrukcji z przeładowanym operatorem dodawania: `a=a+b;` będzie zawsze równoważne zachowaniu instrukcji `a+=b;` (w C++ mogą to być całkiem różne zachowania).

List zadań

Przygotować program demonstrujący użycie zaimplementowanych rozwiązań dla różnych przypadków użycia.

1. Napisać klasę `MixedNumber`, do pamiętania liczby mieszanej, która w trzech właściwościach będzie przechowywała część całkowitą, licznik i mianownik (licznik i mianownik zawsze dodatni). Dla części całkowitej należy wykorzystać właściwość automatyczną, dla licznika i mianownika właściwości, które w przypadku podania wartości ujemnej, zapisze wartość bezwzględną. Wartość ułamka powinna być **zawsze** w postaci uproszczonej do najmniejszego możliwego mianownika, którą zawsze jest większy niż licznik (również po zmianie mianownika lub licznika). Zaproponować minimum 3 konstruktory (część powinna wywoływać inny konstruktor). Przeładować operator '+'. Klasa ma posiadać właściwość do pobierania ułamka jako liczby typu `double`. Zaimplementować własną metodę `ToString()` dla tej klasy. Klasa ma również liczyć ile razy (globalnie) należało zmodyfikować reprezentację obiektu, aby zmodyfikować go do postaci zgodnej z założeniem. Ma też, oczywiście, udostępniać możliwość pobrania wartości tego licznika.
2. Napisać funkcję rozszerzającą klasę `string` o możliwość zamiany ciągu znaków w ten sposób, że pozycje parzyste ciągu będą zamienione na wielkie litery, a nieparzyste na małe. Pozycje nie będące literami mają zostać zamienione na kropkę '.'.

Data I: Spotkanie 6 (max 100 punktów)

Data II: Spotkanie 7 (max 80 punktów)

Data III: Spotkanie 8 (max 50 punktów)