

Programowanie obiektowe

Lista 4.

Poniższe zadania mają rozszerzyć możliwości klas zaprogramowanych na poprzedniej liście z wykorzystaniem poznanych na dzisiejszym wykładzie cech języka C^\sharp .

Za tę listę można otrzymać 8 punktów.

Zadanie 1

Uzupełnij zaprogramowane na poprzedniej liście klasy o następujące elementy:

- metodę `string ToString()` zwracającą czytelnie sformatowany napis reprezentujący obiekt;
- metodę `bool Equals(Object obj)` porównującą zaprogramowane wyrażenia. Na przykład w zad. 1 takie wyrażenia są równe:

$$x + x \text{ i } x + x$$

a takie wyrażenia

$$x + y \text{ i } x + x$$

już nie. Trudniejsza sprawa jest z wyrażeniami, które „wyglądają podobnie”, ale mają pozmienniane nazwy zmiennych, np.

$$x * x + 2 * x * x * x \text{ i } z * z + 2 * z * z * z$$

Można zaprogramować `Equals()` tak, by zwracała *true* dla takich wyrażeń, ale jest to trudniejsze i zostawiam to jako zadanie dla ambitnych;

- operatory (przynajmniej dwa) umożliwiające budowanie w naturalny sposób rozbudowane wyrażenia, np. zamiast pisać

```
Expression expr = new Add(new Const(4), new Variable("x"));
```

można napisać

```
Expression expr = new Const(4) + new Variable("x");
```

- metody umożliwiające przeglądanie kolejnych elementów drzewa wyrażeń za pomocą instrukcji **foreach**. Porządek przeglądania (*preorder* czy *inorder*) jest do wyboru;
- metody statyczne umożliwiające kontrolowanie liczby tworzonych obiektów wybranych klas; na przykład w zadaniu o formułach logicznych nie ma potrzeby tworzenia wielu obiektów reprezentujących stałą `false`, a w zadaniu o wyrażeniach można przyjąć, że obiekt reprezentujący stałą `2` jest tylko jeden. Dla zadania 2. (tego o programach) trudno jest wskazać naturalną klasę, dla której będziemy ograniczać liczbę instancji, więc może nieco sztucznie przyjmijmy, że będziemy ograniczać liczbę obiektów reprezentujących przypisanie ¹.

Marcin Młotkowski

¹Można tu skorzystać z omówionego na wykładzie schematu zarządzania wektorami jednostkowymi i wzorca *singleton*.