C# dla niewtajemniczonych

CZYLI COŚ SOBIE POPROGRAMUJEMY

C#?

- Obiektowy język programowania autorstwa Microsoftu
- Kompiluje się do CIL (Common Intermediate Language)
- … co oznacza, że potrzebuje maszyny wirtualnej (dla MS: .NET; Mono, DotGNU)
- Wersja 1.0 w grudniu 2002 – obecnie 5.0 (6.0?)
- IDE dla C# to Visual Studio (w różnych edycjach)
- MS dostarcza bogate biblioteki do zrobienia prawie wszystkiego – płatne elementy to raczej toole do pracy nad kodem
- Unity – silnik do pisania gier opiera się na C# ;)

C# - co się w nim da zrobić?

- Zapewnia pełną obiektowość – wsparcie semantyczne dla polimorfizmu, abstrakcji itp.
- Garbage Collector
- Właściwości, iteratory, delegaty, typy uogólnione
- Łatwe programowanie wielowątkowe (od 5.0)

Obiektowość

- Program projektujemy jako zbiór obiektów, które ze sobą współpracują.
- Definiujemy klasy (czyli matryce obiektów), opisujące informacje o nich (dane/pola) oraz to co robią (metody) i jak robią (implementacja metod)
- Obiekt to konkretna instancja danej klasy
- W C# wszystko jest klasą (np. żeby narysować linię na ekranie, tworzymy obiekt „Pen”, który zawiera m. in. obiekt „Color” itp.)

Dygresja – Obsługa Visual Studio

- Okno główne, słowa kluczowe
- Intellisense
- Panel składowych projektu
- Panel składowych klasy
- Prosty przykłady forms
- Nawigacja, skróty klawiszowe
- Breakpointy, podgląd danych.

Obiektowość – 4 terminy

- Dziedziczenie
- Hermetyzacja
- Polimorfizm
- Abstrakcja

Obiektowość - dziedziczenie

- Możliwość rozszerzenia możliwości jednej klasy (bazowej), tworząc inną klasę (dziedziczącą) – można dodać lub nadpisać dane i metody.
- W efekcie powstaje hierarchia (drzewko) kolejnych, coraz bardziej sprecyzowanych klas.
- Główne zastosowanie – uwspólnienie kodu
- Implementuje relację („jest podobne do” – a nie „składa się z”)
- W C# mamy tylko dziedziczenie pojedyncze

Obiektowość - hermetyzacja

- Możliwość ukrycia części danych lub metod.
- W efekcie można stworzyć „czarną skrzynkę”, która czymś się zajmuje – i nikogo nie interesuje JAK coś jest robione w środku.
- Główne zastosowanie – bezpieczeństwo wykonania (np. dostęp do danych tylko metodami), uproszczenie użytkowania klasy.
- C# udostępnia poza bazowymi zakresami dostępu (*public*, *private*) dodatkowe jak *internal*, *external* itp.
- C# dodaje tak zwane właściwości (*properties*)„akcesory” *get;* i *set;*

Obiektowość - polimorfizm

- Możliwość użycia takich samych metod, operatorów, funkcji itp. na różne sposoby (wynika to z kontekstu użycia)
- Rozróżniamy polimorfizm statyczny i dynamiczny
- Statyczny: przeciążanie metod (np. konstruktora) albo operatorów.
- Dynamiczny (gdy aplikacja już działa): np. przydzielanie różnych metod wirtualnych w trakcie tworzenia obiektu

Obiektowość - abstrakcja

- Wsparcie dla koncepcji programistycznej, która polega na wydzielaniu (wyabstrahowaniu?) pewnej wspólnej części wielu problemów (np. algorytmów).
- C# używa słów kluczowych *abstract*, *virtual*, (klasy i metody) oraz *interface* (tylko klasy)
- Typowe wykorzystanie, to stworzenie klasy bazowej (*abstract* lub *interface*) i w efekcie wymuszenie na wszystkich jej użytkownikach określonych implementacji konkretnych metod (zapewne różnych pomiędzy sobą)

Zasady SOLID

Bibliografia

- C# 5.0 Pocket Reference; Albahari Joseph, Albahari Ben
- C#. Rusza głową!; Greene Jennifer, Stellman Adnrew
- http://www.tutorialspoint.com/csharp/index.htm
- http://pankajtiwarii.blogspot.com/p/oops-abstraction-encapsulation.html