Kurs programowania

Wstęp - wykład 0

Wojciech Macyna

22 lutego 2016

Historia

- Simula 67 – język zaprojektowany do zastosowan symulacyjnych;
- Smalltalk 80 – pierwszy język w pełni obiektowy;
- Dodawanie obiektowości do języków imperatywnych: Pascal – Delphi, C – C++, Ada, PHP, ...;
- Języki w „pełni” obiektowe: Java, C#

Podstawowe cechy języka obiektowego

- Abstrakcyjne typy danych.
- Dziedziczenie.
- Dynamiczne wiązanie wywołań metod z ich właściwymi definicjami.

Podstawowe definicje programowania obiektowego

Klasy, obiekty i podklasy

- Abstrakcyjne typy danych w językach obiektowych zwykle nazywane są klasami.
- Instancje klas są nazywane obiektami.
- Klasa zdefiniowana przez dziedziczenie z innej klasy nazywana jest klasą pochodną lub podklasą.
- Klasa z której stworzono klasę pochodną nazywamy klasą bazową lub nadklasą.
- Podproprogramy które definiują operacje na obiektach klasy nazywamy metodami.
- Przechowywane w klasie dane nazywamy polami.

Podstawowe definicje programowania obiektowego

Sterowanie dostępem

- Sterowanie dostępem pozwala ukryć pewne części klasy przed ich nieuprawnionym użyciem przez inne klasy – deklaracje public i private.
- Deklaracja protected umożliwia publiczny dostęp do elementów klasy przez klasy z tego samego pakietu, a jednocześnie zabrania innym.
- Klasa pochodna dziedziczy wszystkie pola i metody swojej klasy bazowej ale może modyfikować odziedziczone metody.

Dziedziczenie pojedyncze i wielokrotne

- Jeżeli klasa jest podklasą tylko jednej klasy bazowej to mówimy o dziedziczeniu pojedynczym.
- Jeżeli klasa ma więcej niż jedną nadklasę to proces nazywamy dziedziczeniem wielokrotnym.

Podstawowe definicje programowania obiektowego

Metody i klasy abstrakcyjne

- Jeśli w klasie chcemy zdefiniować metodę której implementacja ma sens dopiero w klasie pochodnej to metodę taką nazywamy abstrakcyjną.
- Klasę zawierającą przynajmniej jedną metodę abstrakcyjną nazywamy klasą abstrakcyjną.
- Nie można tworzyć instancji klasy abstrakcyjnej.

Metody i pola obiektu i klasy

- Klasy mogą mieć metody i pola instancyjne oraz metody i pola klasowe (statyczne).
- Pola i metody instancyjne są zawsze przypisane do swojej instancji obiektu (pola określają stan obiektu).
- Pola i metody klasowe należą do całej klasy i dla klasy jest tylko jedna ich kopia (ich użycie nie wymaga istnienia jakiejkolwiek instancji klasy).

Podstawowe definicje programowania obiektowego

Polimorfizm i dynamiczne wiązania

- Zmienne (referencje, wskaźniki) klasy bazowej mogą odwoływać się także do obiektów dowolnej podklasy tej klasy bazowej (relacja przechodnia).
- Nadklasa może definiować metody które są przedefiniowane przez jej podklasy.
- Kiedy wywoływana jest metoda przez zmienną klasy bazowej (lecz wskazującą na klasę pochodną) to wywoływana jest dynamicznie metoda z właściwej podklasy.
- Rodzaj polimorfizmu: dynamiczne wiązanie wywołań z definicjami metod.

Problemy implementacyjne

Czy wszystko jest obiektem?

- W czystym modelu programowania obiektowego wszystkie typy są klasami, nie ma różnicy między klasami predefiniowanymi w języku a własnymi, wszystkie działania realizowane są przez wywoływanie metod – rozwiązanie eleganckie lecz praktycznie niestosowane.
- Dodanie do języka imperatywnego obiektów (C++) – rozwiązanie sprawiające czasami kłopoty.
- Prawie pełna obiektowość przy pozostawieniu prostych typów danych (Java).

Inne problemy

- Sprawdzanie zgodności typów i polimorfizm (dziedziczenie).
- Dziedziczenie pojedyncze czy wielokrotne.
- Alokacja i dealokacja obiektów.

Przegląd języków obiektowych

Smalltalk

- Dziedziczenie tylko pojedyncze, podklasy mają pełny dostęp do klasy bazowej, istnieje systemowa klasa Object będąca przodkiem wszystkich klas.
- Stosowane jest dynamiczne wiązanie wywołań z metodami i dynamiczne wyszukiwanie metody po kolei przez wszystkie nadklasy.
- **Zalety:** prosta regularna składnia, prawie pełna obiektowość.
- **Wady:** niska efektywność – prawie wszystko jest robione dynamicznie, trudne wykrywanie błędów.

Przegląd języków obiektowych

C++

- Język hybrydowy (mieszanka języka imperatywnego i obiektowego).
- Dziedziczenie wielokrotne, brak klasy podstawowej, składniki prywatne dostępne tylko dla klas zaprzyjaźnionych (friends), klasa pochodna może zmienić tryb dostępu do dziedziczonych składników.
- Obiekty zadeklarowane za pomocą zmiennych niewskaźnikowych mają statyczne wiązania metod, zmienne wskaźnikowe dynamiczne, klasy abstrakcyjne nie mają instancji ale ich wskaźniki mogą służyć do polimorficznych wywołań metod z podklas nieabstrakcyjnych.
- **Zalety:** wysoka efektywność kodu (niewiele wiązań jest tak naprawdę dynamiczna).
- **Wady:** złożoność języka spowodowana jego dwoistością.

Przegląd języków obiektowych

Java

- Język prawie całkowicie obiektowy (wyjątek to typy podstawowe).
- Dziedziczenie pojedyńcze ale z wyjątkiem – można dziedziczyć dowolną ilość klas całkowicie abstrakcyjnych (interfejsy), można deklarować metody i klasy abstrakcyjne, istnieje klasa podstawowa Object.
- Wywołania są wiązane z metodami dynamicznie z wyjątkiem metod statycznych i zadeklarowanych jako final.
- **Zalety:** język bardzo jednorodny z prostym sterowaniem dostępem.
- **Wady:** zachowanie typów podstawowych stwarza czasami problemy.

Przegląd języków obiektowych

C#

- Język bardzo podobny do Javy

Główny język wykładu - dlaczego Java?

- Historia Javy (1995-...).
- Nowoczesny język obiektowy (ale z pewnymi ograniczeniami ułatwiającymi użycie).
- Maszyna Wirtualna Javy – przenoszalność oprogramowania.
- Język ma tylko jeden dialekt (nie tak jak np. C czy C++).
- Powszechny darmowy dostęp.
- Duża ilość dostępnych bibliotek – ułatwienie i skrócenie procesu programowania.
- Wpisana w język wielowątkowość.
- Obowiązkowa obsługa błędów (wyjątków).
- Zarządzanie pamięcią – Garbage Collector.

Pierwszy program

HelloWorld.java

```
1  public class HelloWorld {
2    public static void main( String args[] ) {
3      System.out.println( "Hello, World!" );
4    }
5  }
```

Kompilacja - powstanie HelloWorld.class

```
unix> javac HelloWorld.java
```

Uruchamianie

```
unix> java HelloWorld
Hello, World!
unix>
```

Główny język wykładu - dlaczego C++?

- Najpowszechniej stosowany język obiektowy.
- Dużo już stworzonych bibliotek.
- Łatwość pisania przy znajomości C, brak konieczności pisania obiektowego przy możliwości korzystania z bibliotek obiektowych.