Programowanie Obiektowe

Marcin Kamil Bączyk

Wykład 1

8 października 2020

Informacje o przedmiocie

- Prowadzący wykład : mgr inż. Marcin Bączyk
  M.K.Baczyk@elka.pw.edu.pl,
  pokój 449, Teams
  konsultacje w poniedziałki w godzinach 11-12.
- istnieje możliwość umawiania się na inny termin konsultacji
- strona przedmiotu:
  http://staff.elka.pw.edu.pl/~mbaczyk1/PROE_2020Z/index.html
- strona będzie systematycznie aktualizowana, aż do końca semestru.
- na stronie: http://staff.elka.pw.edu.pl/~mbaczyk1/ można znaleźć również odnośniki do stron poprzednich realizacji przedmiotu PROE.

Zaliczenie

- PROE jest przedmiotem zaliczeniowym - ostatni możliwy termin oddawania projektów, poprawiania sprawdzianów itp. przypada na ostatni dzień semestru
- Przedmiot podzielony jest na dwie części : wykład (15) oraz laboratoria (14)
- W trakcie wykładu odbędą się dwa sprawdziany (???) wykładowe po 25 (???) punktów każdy
- W trakcie zajęć laboratoryjnych odbędzie się cztery (???) zajęcia oceniane w skali 0-5 (???) punków każde
- W trakcie zajęć laboratoryjnych będą realizowane dwa małe projekty oceniane w skali 0-15 (???) punktów
- Prowizoryczny harmonogram laboratorium dostępny na stronie przedmiotu
- Terminy sprawdzianów oraz zajęć laboratoryjnych ustalone zostaną w trakcie kolejnych zajęć.

Zaliczenie

- Aby zaliczyć przedmiot należy spełnić następujące warunki:
  - uzyskać łącznie minimum 50% możliwych do uzyskania w trakcie zajęć wykładowych
  - uzyskać łącznie minimum 50% możliwych do uzyskania w trakcie zajęć laboratoryjnych
  - uzyskać 51 punktów z całego przedmiotu
- Progi ocen
  - $\geq$ 91 - 5
  - $\geq$ 81 - 4.5
  - $\geq$ 71 - 4
  - $\geq$ 61 - 3.5
  - $\geq$ 51 - 3
  - $<$ 51 - 2

Laboratorium

- Terminy
  - poniedziałek 18:15 - 20:00 - grupa 102
  - wtorek 10:15 - 12:00 - grupa 101
  - wtorek 10:15 - 12:00 - grupa 103
- limit miejsc w grupie : 12 osób

Treść wykładu i efekty kształcenia

Treść wykładu

- Zapoznanie się z elementami projektowania obiektowego
  - pojęcie architektury oprogramowania
  - pojęcie wzorca projektowego
- Zapoznanie się z możliwościami nowoczesnego języka C++
- Zapoznanie z możliwościami biblioteki standardowej C++

Efekty kształcenia

1. Zapoznanie z paradygmatem programowania obiektowego.
- Zdobycie umiejętności rozumienia kodu napisanego w C++.
- Zdobycie umiejętności pisania własnych aplikacji C++.
- Zdobycie umiejętności samodzielnego poszukiwania niezbędnych informacji.

Treść wykładu

Projektowanie obiektowe

- Obiektowe podejście w wytwarzaniu oprogramowania
- Zasady projektowania

Założenia paradygmatu

- Abstrakcja
- Hermetyzacja
- Polimorfizm
- Dziedziczenie

Składnia języka C++

- Klasy
- Konstruowanie i niszczenie obiektów
- Szablony
- Obsługa wyjątków
- Zarządzanie pamięcią
- ...

Nauka programowania

- uczestnictwo w zajęciach laboratoryjnych i wykładach
- samodzielna realizacja projektów
- polecana literatura:
  - programowanie obiektowe: (ostrożnie)
    - "Czysty kod" oraz "Czysta architektura" - Robert Martin
    - "Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku" - Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
    - "Refaktoryzacja. Ulepszanie struktury istniejącego kodu" - Martin Fowler

Nauka programowania

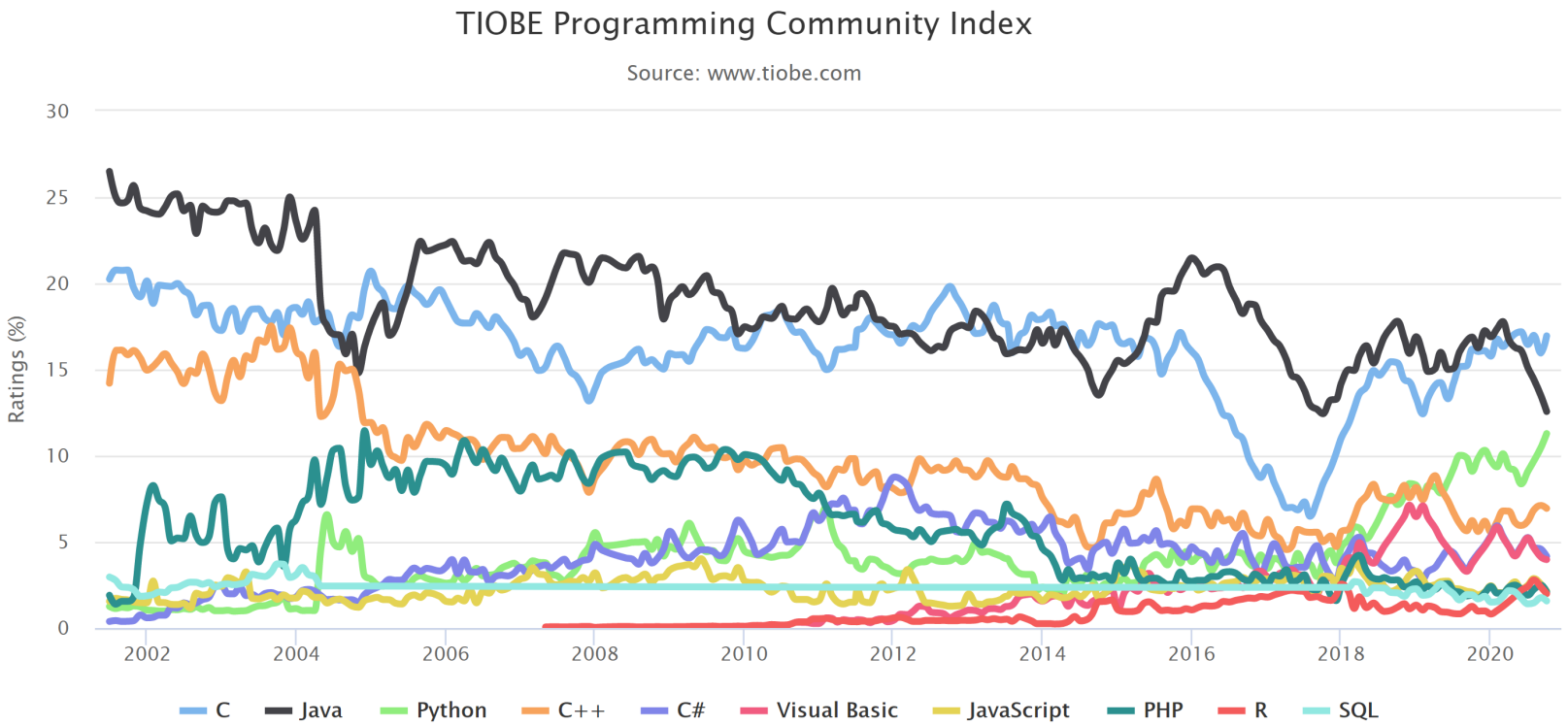
- uczestnictwo w zajęciach laboratoryjnych i wykładach
- samodzielna realizacja projektów
- polecana literatura:
  - programowanie obiektowe: (ostrożnie)
    - "Czysty kod" oraz "Czysta architektura" - Robert Martin
    - "Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku" - Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
    - "Refaktoryzacja. Ulepszanie struktury istniejącego kodu" - Martin Fowler
  - język C++:
    - ...
    -

Nauka programowania

- uczestnictwo w zajęciach laboratoryjnych i wykładach
- samodzielna realizacja projektów
- polecana literatura:
  - programowanie obiektowe: (ostrożnie)
    - "Czysty kod" oraz "Czysta architektura" - Robert Martin
    - "Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku" - Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
    - "Refaktoryzacja. Ulepszanie struktury istniejącego kodu" - Martin Fowler
  - język C++ (być może)
    - "Język C++. Kompendium wiedzy" - Stroustrup Bjarne
    - "Podstawy języka C++" - Stanley Lippman
    - "Język C++. Standardy kodowania. 101 zasad, wytycznych i zalecanych praktyk" - Herb Sutter, Andrei Alexandrescu
    - https://en.cppreference.com/w/
    - https://stackoverflow.com/

Porównanie "popularności" poszczególnych języków

Ale czy to wszystko ma sens ?!



Elementy C++ umożliwiające/ułatwiające programowanie obiektowe

- Klasy oraz obiekty będące ich instancjami
- Dziedziczenie, dziedziczenie wielobazowe
- Konstruktory oraz destruktory
- Metody wirtualne klas
- Zarządzanie pamięcią
- Kontrola typów
- Obsługa sytuacji wyjątkowych

C++ - klasa a obiekt

Klasa:

Jest to definicja wzorca (opis wszystkich danych i dostępnych metod) według którego będą tworzone obiekty danego typu. Innymi słowy klasa to nowy typ zmiennych definiowany przez użytkownika.

Obiekt:

Obiekty są instancjami konkretnych klas (typów własnych).

Uwaga

Wykorzystywanie definicji klasy nie jest jedynym mechanizmem pozwalającym na tworzenie obiektów. Niektóre języki obiektowe implementują mechanizm prototypów, niedostępny w języku C++.

Pierwsza klasa

Podobnie jak język C pozwala stworzyć pustą strukturę, język C++ pozwala stworzyć pustą klasę.

```
class First;
class First {};

int main(void)
{
    First f1;
    return 0;
}
```

- słowo kluczowe
- deklaracja klasy
- definicja klasy
- obiekt

Czy istnieje jakieś wytłumaczenie dla istnienia pustej klasy? Jaki jest jej rozmiar?

Typy w C++

typ prosty

Typ nazwaZmiennej;

np. float, double, int, ... oraz

np. klasy, struktury, ...

typ wskaźnikowy

Typ * nazwaZmiennejWskaznikowej = &nazwaZmiennej

Typ wskaźnikowy oznacza adres pod jakim zapisana jest dana zmienna

typ referencyjny

Typ & nazwaZmiennejReferencyjnej = nazwaZmiennej

Typ referencyjny należy rozumieć jako odnośnik do zmiennej

Typ referencyjny

L-referencja

Typ & nazwaZmiennejReferencyjnej = nazwaZmiennej

- definicja zmiennej referencyjnej zawsze wymaga podania obiektu (inicjacji)
- związku pomiędzy zmienną referencyjną a obiektem nie można zerwać
- nie można zdefiniować referencji do referencji
- nie można zdefiniować wskazania na referencję (ale referencję do wskazania już tak) - w związku z powyższym nie można używać tablic referencji

R-referencja

Typ && nazwaZmiennejReferencyjnej = zmiennaNieposiadajaNazwy

Typy w C++ - przykłady

```
int i = 1;          // i = 1
int *pi;
pi = &i;            // i = 1
*pi = 2;            // i = 2
// int & ri - Uwaga Blad
int & ri = i;       // i = 2
ri++;               // i = 3

First f;
First *pf = &f;
// First &rf;    //'rf': references must be initialized
First &rf = f;
```

Referencja wskazuje na dany obiekt, również wtedy gdy obiekt przestanie istnieć (np. pamięć zostanie zwolniona). Język C++ pozwala zwrócić referencję do obiektu tymczasowego - to nigdy nie kończy się dobrze.