Programowanie - część I

Anna Gogolińska

Programowanie

- Języki programowania oparte są o pewne wzorce, można je podzielić na kilka rodzajów (paradygmaty).
- Języki oparte na tym samym paradygmacie są do siebie bardzo podobne - występują w nich analogiczne elementy.
- Obecnie najbardziej popularne paradygmanty: programowanie obiektowe oraz programowanie proceduralne.

Języki obiektowe (proceduralne)

- Języki:
  - C++
  - Java
  - JavaScript
  - PHP
  - Python
- Dobre zrozumienie idei - podstawowych konstukcji w jednym z nich zapewni zrozumienie we wszystkich pozostałych.

Pisanie programów

- Program pisze się używając składni języka (w danym paradygmacie te składnie są podobne).
  - Taki program to plik tekstowy - można go wyświetlić i człowiek może go przeczytać, ale nie będzie zrozumiały dla komputer.
- Aby program był zrozumiały dla komputera należy użyć (w zależności od języka):
  - Kompilatora, który na podstawie pliku z kodem tworzy program wykonywalny i zrozumiały dla komputera.
  - Interpretera, który uruchamia program i wykonuje zawarte w nim instrukcje.

Podstawowe idee

- Program składa się z **instrukcji**.
- W większości języków na końcu instrukcji stawie się średnik **;**
- W niektórych miejscach programu powinno podać się jedną instukcję (według składni języka), ale potrzebujemy podać kilka instrukcji. Wtedy instrukcje łączy się w **blok** i ten blok wpisuje się w miejsce instrukcji. Blok tworzy się zazwyczaj przy użyciu klamer **{ }**.

Podstawowe idee

- Zmienne:
  - Nazwy, które reprezentują jakąś wartość.
  - W programie operuje się nimi jako abstrakcyjnymi wartościami - zakłada się, że to symbol, który reprezentuje wartość.
  - Podczas wykonania programu w miejscu zmiennych są konkretne wartości.
  - Te wartości są podane np. przez użytkownika, obliczone, pobrane z bazy danych.
  - W dobrze napisanym programie trzeba uwzględnić wszystkie możliwe wartości zmiennej.

Podstawowe idee

- Zmienne (ciąg dalszy):
  - Zmienne zawsze mają typ - określa on rodzaj wartości jaki może przyjąć zmienna.
  - W niektórzych językach wprost trzeba określić typ zmiennej i jest on stały. W innych typ jest ustalany na podstawie wartości zmiennej i może się zmienić.
  - Podstawowe typy:
    - Liczba całkowita
    - Liczba zmiennoprzecinkowa (ułamek dziesiętny)
    - Tekst
    - Znak (litera, cyfra)
    - Wartość logiczna (prawda/1/true lub fałsz/0/false)

Podstawowe idee

- Zmienne (ciąg dalszy 2):
  - var a = 2;
    var b = 3;
    var c = 2 + 3;
  - string str = "abc";
    string str2 = "def";
    cout << str << str2;

Podstawowe idee

- Instrukcje warunkowe:
  - Ich idea polega na tym, że na początku instrukcji sprawdza się warunek - warunek jest podany przez osobę piszącą.
  - Warunek zazwyczaj związany jest ze sprawdzeniem jaką wartość ma jakaś zmienna.
  - Dalej w zależności od tego czy warunek jest spełniony (jest prawdziwy) wykonywane są określone instrukcje - wpisane w odpowiednim miejscu. A jeśli nie jest spełniony (prawdziwy) to wykonywane są inne instrukcje (albo żadne).

Podstawowe idee

- Instrukcje warunkowe (ciąg dalszy):
  - Instrukcja warunkowa pozwala dostosować działanie programu do możliwych wartości zmiennych. Pozwala się zabezpieczyć na różne przypadki.
  - Podczas jednego wykonania programu warunek będzie albo prawdziwy, albo fałszywy. Ale w różnych wykonaniach raz może być prawdziwy, a raz fałszywy.

Podstawowe idee

- Pętle:
  - Ich ideą jest to, że pewne instukcje (te które się umieszcza „wewnątrz” pętli) mogą być wykonane wiele razy.
  - Poza instrukcjami wewnątrz pętli, pętla zawiera określenie warunków, na podstawie których określa się ile razy pętla będzie powtórzona.
  - Zazwyczaj w danym języku jest kilka rodzajów pętli, różnią się one sposobem określania warunków powtarzania pętli.

Podstawowe idee

- Pętle (ciąg dalszy):
  - Zazwyczaj warunek ile razy wykona się pętla uzależnione jest od wartości jakiejś zmiennej.
  - Z tego powodu podczas jednego wykonania programu pętla wykona się określoną ilość razy, ale w różnych wykonaniach pętla może się wykonywać różną ilość razy.

Podstawowe idee

- Funkcje:
  - Funkcja to blok instrukcji, który ma nazwę.
  - W kodzie zamiast pisać ciąg instrukcji z funkcji podaje się nazwę funkcji. Podczas wykonania programu w miejscu w którym jest nazwa funkcji wykonywane są instrukcje z funkcji.
  - Aby funkcja mogła się „kontaktować” z resztą programu można ją napisać tak, aby przekazywać do funkcji jakieś wartości oraz aby funkcja zwracała wartość. Wartość ta „trafia” w miejsce użycia funkcji (wywołania).