Lekcja 20.10.2023

Temat: Przypomnij sobie, podstawy tworzenia algorytmów.

**Przykład działania algorytmu na przykładzie ciasta**

Dane wejściowe algorytm dane wyjściowe

Składniki przepis gotowe ciasto

**Algorytm** – jest to skończony i uporządkowany opis precyzyjnie sformułowanych czynności prowadzących do przetworzenia danych wejściowych na dane wyjściowe.

Sposoby reprezentacji algorytmu

Zapis słowny (zdanie po zdaniu)

Lista kroków

Schemat blokowy

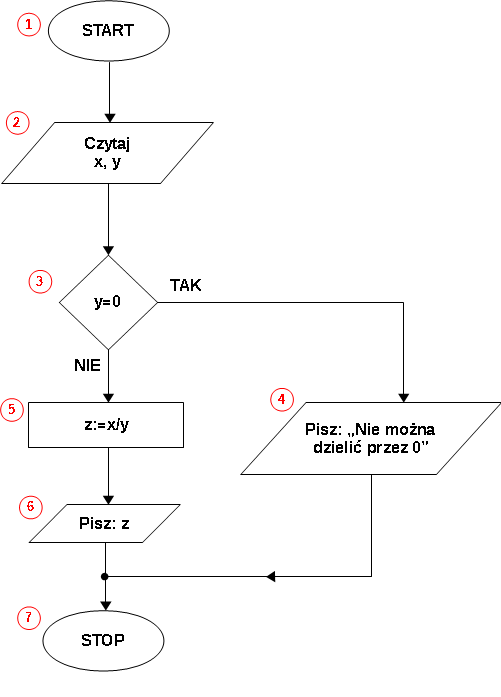
-bloki graniczne (start/stop)

-blok wejścia/wyjścia (wczytywanie/wyświetlanie danych)

Pseudokod

Jest to sposób reprezentacji algorytmu jak najbardziej zbliżony do języka programowania.

.

blok graniczny ->

blok wejścia/wyjścia ->

blok decyzyjny (warunkowy) ->

przykład schematu blokowego

**Cechy dobrego algorytmu**:

* **Precyzyjne zdefiniowanie** - algorytm nie może pozostawiać dowolności w interpretacji zapisu. Każdy, kto z niego skorzysta, musi go rozumieć w taki sam sposób.
* **Skończoność** - algorytm musi po pewnej liczbie kroków dać wynik, czyli wygenerować dane wyjściowe. Liczba operacji musi być skończona.
* **Wykonywalność -** wszystkie operacje opisane w algorytmie muszą być możliwe do wykonania.
* **Efektywność** - algorytm powinien być tak ułożony, by wynik jego działania był dostępny po jak najmniejszym zaangażowaniu zasobów komputera i w jak najkrótszym czasie. Zależy to jednak nie tylko od algorytmu, lecz tak że od właściwości komputera, języka użytego do ułożenia programu, systemu operacyjnego i wielu innych parametrów.

**Lekcja**

**Temat:** Narzędzia, czyli jak dobrać i konfigurować środowisko programistyczne.

**IDE –** (Integral Development Environment) Zintegrowane Środowisko Programistyczne, jest to program lub zespół programów służących do tworzenia, testowania, modyfikowania i konserwacji programów komputerowych w danym języku.

**MinGW –** (Minimalistic GNU for Windows) Podstawowe narzędzie służące do kompilacji

**Kompilacja –** Jest to zamiana kodu źródłowego (kodu pisanego) na język wynikowy (język maszynowy)

**Lekcja**

**Temat: Podstawy programowania.**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**Return 0;**

**}**

**#** - dyrektywa preprocesora; jest to wykonanie operacji przed analizą kodu źródłowego.

**include <iostream>** - dołączenie biblioteki (zestawu instrukcji) strumienia wejścia i wyjścia

**using namespace std** – stosowanie standardowych przestrzeni nazw (nie musimy zapisywać std::

w każdej linii kolumn).

**Int main() –** Jest to główna funkcja programu

**return 0** – informacja dla kompilatora o poprawnym zakończeniu programu.

**Deklaracja zmiennej w programowaniu**

**Zmienna** to informacja zapisana w komórce pamięci RAM

Typ nazwa = wartość

**Tworzenie komentarzy C++**

**Komentarz jednoliniowy** - **//**

**Komentarz wieloliniowy** -  **/\*** … **\*/**

**Operatory arytmetyczne w języku C++**

|  |  |
| --- | --- |
| Dodawanie | + |
| Odejmowanie | - |
| Mnożenie | \* |
| Dzielenie | / |
| Reszta z dzielenia (modulo) | % |

**Typy danych stało-przecinkowych**

**Nazwa typu Liczba bajtów Zakres wartości**

|  |  |  |
| --- | --- | --- |
| **short** | 2 | Od -32 768 do 32 767 |
| **int** | 4 | Od -2 147 483 648 do 2 147 483 647 |
| **unsigned int** | 4 | Od 0 do 4 294 967 295 |
| **long** | 4 | X |
| **unsinged long** | 4 | X |
| **long long** | 8 | X |

**Typy danych zmienno-przecinkowych**

**Nazwa typu Liczba bajtów Zakres wartości**

|  |  |  |
| --- | --- | --- |
| **Float** | 4 | 3.4E +/- 38 (dokładność do 7 cyfr) |
| **Double** | 8 | 1.7E +/- 308 (dokładność do 15 cyfr) |

**Pierwsza postać instrukcji warunkowej**

**if(warunek)**

**{**

**instrukcja1;**

**instrukcja2;**

**…**

**instrukcjaN;**

**}**

**Druga postać instrukcji warunkowej**

**if(warunek)**

**{**

**instrukcja1;**

**instrukcja2;**

**}**

**else**

**{**

**instrukcja3;**

**instrukcja4;**

**}**

**Trzecia postać instrukcji warunkowej**

**If(warunek1)**

**{**

**Instrukcja1;**

**Instruckja2;**

**}**

**Else if(warunek2)**

**{**

**Instrukcja3;**

**}**

**Else instrukcja4;**

**Pętle w C++**

**Pętla for**

**for(1;2;3)**

**{**

**3;**

**}**

**1 – jest to wartość początkowa zmiennej pomocniczej**

**2 – jest to warunek powtórzenia pętli**

**Pętla while**

**int a=5**

**while (a<10)**

**{**

**cout << a << endl;**

**A++;**

**}**

**Pętla wyświetli na ekranie liczby od 5 do 9**

**Pętla dowhile**

**int a=10;**

**do**

**{**

**cout << a << endl;**

**a+=10;**

**}**

**while(a<=50);**

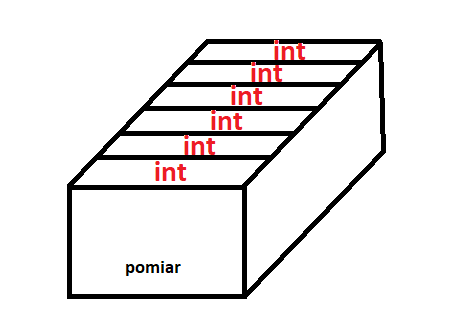
**Program wyświetli liczby 10, 20, 30, 40 i 50.**

**Pętla dowhile wykona się przynajmniej raz w odróżnieniu do pętli while, która może wykonać się 0 razy.**

**Tablica jest rodzajem zmiennej w programowaniu, która posiada wiele innych zmiennych tego samego typu. Przykład deklarowania tablicy jednowymiarowej:**

**int pomiar[10]**

**program utworzy tablicę o nazwie „pomiar” która posiada 10 elementów (liczb całkowitych).**

****

**int n;**

**int pomiar[n]**

**cin >> n;**

**Odwoływanie się do poszczególnych komórek w tablicy nazywamy indexowaniem.**

**Dla tablicy 10-elementowej odwołujemy się do komórek przez pomiar[0] do pomiar[9] (n-1).**

**Do poruszania się po komórkach tablicy wykorzystujemy pętle:**

**int pomiar[10];**

**for(int i=0;i<10;i++)**

**{**

**cin >> pomiar[i];**

**}**

**Deklarowanie tablicy dwuwymiarowej**

**Int tab[4][3];**

**4 \/ 3 >**

|  |  |  |
| --- | --- | --- |
| **[0][0]** | **[0][1]** | **[0][2]** |
| **[1][0]** | **[1][1]** | **[1][2]** |
| **[2][0]** | **[2][1]** | **[2][2]** |
| **[3][0]** | **[3][1]** | **[3][2]** |

**Definicja funkcji jest to podprogram który możemy wywołać w dowolnej części programu głównego.**

**Na sprawdzian:**

* **Definicja algorytmu**
* **Sposoby reprezentacji algorytmu (5) (lista kroków, drzewko decyzyjne, zapis słowny, schemat blokowy, pseudokod)**
* **Cechy dobrego algorytmu (4)**
* **Przykłady danych wejściowych algorytmu (temat 6 podręcznik)**
* **Definicje (temat 9) IDE**
* **(Temat 10) Kompilacja (programowanie)(definicja z notatek)**
* **Co to MinGW**
* **Co oznacza | # | include <iostream> | using namespace std; | int main() | return 0; | (C++)**
* **Zapis komentarzy w C++**
* **Co to Debugger**
* **(temat 11) Operatory arytmetyczne (matematyczne) C++ (kilka wymienić)**
* **Operatory relacyjne C++ (<, >, =np.)(6 jest)**
* **Operatory logiczne C++ (są 2)**
* **Przykłady danych stałoprzecinkowych (int np.)**
* **Przykłady danych zmiennoprzecinkowych (float, double)**
* **3 instrukcje wrunkowe w C++ (if, if else, i coś jeszcze)**
* **3 rodzaje pętli w C++**
* **Definicja tablicy C++**
* **Deklaracja tablicy C++ (jak tworzymy tablice dwuwymarową np.)**
* **Definicja funkcji C++**