JĘZYKI PROGRAMOWANIA

Aby przedstawić algorytm w postaci programu komputerowego, należy zapisać go jako ciąg instrukcji języka programowania. Każda instrukcja odpowiada określonej operacji lub ciągowi operacji.

Język programowania posiada swój:

- Zbiór instrukcji (w tym słowa kluczowe oznaczające określony rozkaz, instrukcję lub polecenie)
- Zasady składni
- Właściwe słownictwo.

Instrukcje każdego języka programowania realizują czynności takie, jak:

- Wprowadzanie danych,
- Wyprowadzanie wyników,
- Wykonywanie obliczeń,
- Sprawdzanie warunków,
- Powtarzanie operacji.

Zasady dotyczące języka programowania:

- Podlega konkretnym regułom,
- Precyzja - instrukcje błędne nie mogą być przetłumaczone, komputer nie wykona programu,
- Kolejność zapisywania instrukcji odpowiada kolejności operacji realizujących dany algorytm.

Języki programowania dzielimy ze względu na poziom wykonywania programu na:

- Języki wysokiego poziomu (np. Java, C++, Python, Pascal, PHP)
- Języki niskiego poziomu (wewnętrzne) (np. assemblery).

Kod źródłowy można zapisać w dowolnym edytorze tekstu, najwygodniej jest korzystać z edytora tekstowego. W edytorze piszemy zgodnie ze składnią wybranego języka programowania.

Program źródłowy nie może być wykonywany przez procesor. Procesor wykonuje kod maszynowy, a dokładniej wykonuje program w kodzie maszynowym (w języku wewnętrznym procesora).

Program komputerowy to:

- program (kod) źródłowy - instrukcje zapisane w języku programowania,

- program (kod) wynikowy (maszynowy), zrozumiały dla komputera, ciąg instrukcji zrozumiałych dla procesora.

Program napisany w języku wysokiego poziomu, musi zostać przetłumaczony na język niskiego poziomu (wewnętrzny język komputera). Proces ten nazywamy translacją.

Napisany przez siebie program sprawdzamy w kompilatorze (C++) lub interpreterze (Python) - sprawdzamy poprawność zapisania poleceń, zgodnie z regułami składni tego języka.

Implementacja – powstanie programu (kodu) źródłowego za pomocą algorytmu z ciągiem instrukcji języka programowania (kodu źródłowego)

Translacja – przetłumaczenie programu źródłowego napisanego w języku wysokiego poziomu na program w języku niskiego poziomu (język wewnętrzny komputera).

Translacja przebiega w formie kompilacji (C, C++) lub interpretacji (LOGO, Scratch, Python), wykonywanych w translatorze (programie do tłumaczenia programu na kod maszynowy).

Interpretacja – tłumaczenie programu tworzonego w języku programowania instrukcja po instrukcji, każdorazowo przy uruchomieniu programu. Piszemy program składający się z wielu poleceń (skrypt), zapisujemy do pliku, uruchamiamy. Np. Python - język interpretowany.

Kompilacja – przetłumaczenie programu w całości na język zrozumiały dla procesora, tak by mógł go wykonać komputer. Jeśli kompilacja przebiegnie poprawnie, można uruchomić program (nie trzeba go już powtórnie tłumaczyć). Piszemy program, zapisujemy w pliku, kompilujemy, uruchamiamy poprawną (!) wersję. Przykładem jest C++ - język kompilowany.

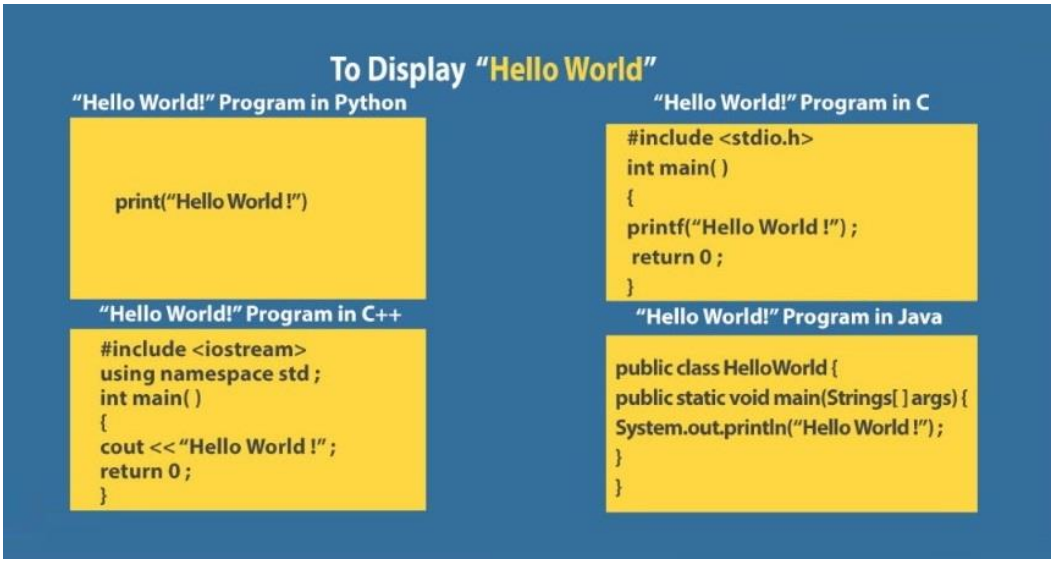
Aby pisać programy w języku Python należy zainstalować aplikację Python, w skład której wchodzi zintegrowane środowisko programistyczne **IDLE.**

IDLE - Integrated Development and Learning Environment.

IDLE Pythona zawiera **powłokę Python Shell**, **edytor kodu źródłowego** i **interpreter** oraz inne narzędzia wspomagające programowanie.

Okno powłoki Pythona umożliwia pracę w dwóch trybach: **interaktywnym** (od razu widzimy wynik, znak zachęty >>>) i **skryptowym** (do tworzenia programów).

Program piszemy w edytorze kodu źródłowego, zapisujemy na dysku, a następnie uruchamiamy poprzez wywołanie interpretera Pythona.



Etapy tworzenia programu w języku Python:

1. W oknie powłoki Pythona (Python Shell) wybieramy opcję **File/New File**, po czym nastąpi otwarcie okna edytora kodu źródłowego.
2. W oknie edytora piszemy program.
3. Zapisujemy program w pliku (**File/Save As** i rozszerzenie .py)
4. Uruchamiamy program poprzez wybór w menu **Run/Run Module** (lub F5). Wynik programu pojawi się oknie powłoki.
5. Jeżeli wystąpią błędy – zostaną zaznaczone, należy je poprawić

Ćwiczenie 1.

1. Utwórz nowy plik źródłowy i napisz program wyświetlający na ekranie napis „Hallo word!” `print("Hallo word!”)`
2. Zapisz program w pliku pod nazwą **Hallo**.
3. Uruchom program.
4. Jeśli interpreter wykrył błędy, popraw je (literówka lub np. brak nawiasu). Zapisz plik pod tą samą nazwą i ponownie uruchom
5. Zapamiętaj gdzie na komputerze program został zapisany.

Ćwiczenie 2.

1. Zmodyfikuj program utworzony w ćwiczeniu 1, aby wyświetlał oprócz pierwszego wiersza również drugi wiersz "Programuję w Python”.
2. Jeśli interpreter wykrył błędy, popraw je. Zapisz plik pod tą samą nazwą. Ponownie uruchom program.

ZMIENNE W JĘZYKU PYTHON

Zasady dotyczące nazw zmiennych w języku Python:

1. Wielkie i małe litery w nazwach traktowane są odmiennie (np. suma i Suma oznaczać będą różne zmienne). Pisząc program, należy zwracać uwagę na poprawne używanie małych i wielkich liter.
2. W nazwach zmiennych powinno się używać liter, znaku podkreślenia i cyfr.
3. Nazwa nie może zaczynać się od cyfry. Przyjęte jest stosowanie małych liter i niestosowanie polskich liter.
4. W nazwach zmiennych nie wolno stosować spacji. W przypadku nazw kilkuczłonowych zamiast spacji stosujemy znak podkreślenia.
5. Należy nadawać nazwy, które określają znaczenie danej zmiennej, np. suma, liczba_elementow.

Zmiennej stosowanej w programie możemy nadać konkretną wartość za pomocą **instrukcji przypisania**. W instrukcji przypisania zmiennej podanej po lewej stronie instrukcji zostanie przypisana obliczona przez komputer wartość wyrażenia znajdującego się po prawej stronie instrukcji. ZMIENNA = WYRAŻENIE.

Typ zmiennej całkowitej (**int**) i jej wartość: waga = 88, rok = 2021, dlugosc = 100
Typ zmiennej rzeczywistej (**float**) i jej wartość: a = 1,5, srednia = 4,75, wys = 180

INSTRUKCJA WEJŚCIA

Instrukcja wejścia (wartość wpisywana z klawiatury): **input()**

Aby dana wprowadzona za pomocą instrukcji input() została zapamiętana jako liczba, musimy dodać instrukcję, która zamieni ciąg znaków na liczbę:
int() – w przypadku liczb całkowitych, float() – w przypadku liczb rzeczywistych.

a = input("Wprowadź liczbę: ")

- jeśli wpiszemy z klawiatury liczbę 346, w zmiennej a zostanie zapamiętany ciąg znaków „346”, a nie liczba 346,

a = int(input("Wprowadź liczbę: "))

- jeśli wpiszemy z klawiatury liczbę 346, w zmiennej a zostanie zapamiętana liczba całkowita 346,

srednia = float(input("podaj średnią ocen: "))

- jeśli wpiszemy z klawiatury liczbę 4.3, w zmiennej srednia zostanie zapamiętana liczba rzeczywista 4,3,

miasto = input("Wprowadź nazwę miasta: ")

- jeśli wpiszemy z klawiatury nazwę miasta „Kraków”, w zmiennej miasto zostanie zapamiętany ciąg znaków „Kraków”.

Ćwiczenie 3.

1. Utwórz nowy plik źródłowy.
2. Wprowadź polecenia:
```
a = int(input(„Podaj pierwszą liczbę: ”))
b = int(input(„Podaj drugą liczbę: ”))
iloczyn = a * b
print(„iloczyn a*b wynosi: ”, iloczyn)
```
3. Zapisz program pod nazwą iloczyn i uruchom program
4. Jeśli interpreter wykrył błędy, popraw je. Zapisz plik pod tą samą nazwą. Ponownie uruchom program.

Podstawowe operatory arytmetyczne w języku Python:

Operator	Działanie	Przykład instrukcji przypisania	Wynik działania dla danych: a = 11 i b = 4
+	dodawanie	`suma = a + b`	11 + 4 = 15
-	odejmowanie	`roznica = a - b`	11 – 4 = 7
*	mnożenie	`iloczyn = a * b`	11 * 4 = 44
//	dzielenie całkowite (z zaokrągleniem części ułamkowej „w dół” – do największej liczby całkowitej mniejszej od wyniku dzielenia)	`iloraz = a // b`	11 // 4 = 2
/	dzielenie zmiennoprzecinkowe (z zachowaniem części ułamkowej)	`iloraz = a / b`	11 / 4 = 2.75
%	obliczenie reszty z dzielenia dwóch liczb całkowitych	`reszta = a % b`	11 % 4 = 3

Wyprowadzanie komunikatów i wyników - funkcja `print()`

Przykłady:

```
print("Zaczynamy lekcje z programowania")
print(p)
print(a + b)
print(23 + 89)
print("Obwód =", 2 * a + 2 * b)
print("Obwód wynosi:", obwod)
```

Umieszczenie sekwencji znaków **\n** w tekście wyprowadzanym za pomocą instrukcji print() wymusza przejście do nowego wiersza.

Instrukcja:
```
print("Pole prostokąta wynosi:", "\nbok1 * bok2 = ", pole)
```
i instrukcje:
```
print("Pole prostokąta wynosi:")
print("bok1 * bok2 =", pole)
```
dają ten sam wynik.

Aby okno nie zamknęło się automatycznie po wyświetleniu wyniku, należy napisać na końcu programu instrukcję oczekiwania na naciśnięcie klawisza Enter: `input("\nAby zakończyć, naciśnij Enter").`

Ćwiczenie 4.

1. Za pomocą typów wprowadzanych przez użytkownika danych (float, int) wykonaj obliczenia:
   a. Średnia arytmetyczna liczb 40 oraz 4
   b. Średnia arytmetyczna liczb 40,4 oraz 4,5
2. Zapisz plik jako srednia.

Ćwiczenie 5.

1. Napisz program którego wynikiem będzie informacja:

   Dla boku o długości (wartość wprowadzona przez użytkownika) pole kwadratu wynosi: ..

2. Zapisz program jako pole

Jeśli zamknęliśmy okno powłoki Pythona, a chcemy edytować program, można z menu kontekstowego nazwy pliku w oknie Eksploratora plików wybrać polecenie **Edit with IDLE/ Edit with IDLE 3.10**

Link do informacji podsumowujących:
https://youtu.be/BDwly79zLTI