Od problemu do algorytmu – warsztat programisty

Agenda:

- Organizacja i weryfikacja narzędzi.
- Od problemu do projektu: Wspólne tworzenie schematu blokowego i pseudokodu.
- Rozgrzewka z kodem: Ćwiczenia z ‘print’ i ‘input’
- Implementacja projektu: Przekładamy plan na działający kod
- Samodzielne zadania i dobre praktyki: Ćwiczenia, omówienie i refektoryzacja
- Podsumowanie

Nasz warsztat i problem do rozwiązania

Weryfikacja narzędzi

- Czy masz dostęp do **draw.io** (lub innej aplikacji do schematów)?
- Czy masz zainstalowany **PyCharm**?
- Czy PyCharm poprawnie wykrywa interpreter Pythona?
- Tworzenie projektu: **File** > **New Project…**
- Tworzenie pliku: **prawy klik** > **New** > **Python**

Nasz algorytm: Od słów do schematu

```
--------------------------
Jan Kowalski
Programista Back-end
--------------------------
```

Nasz algorytm: Od słów do schematu

- **Przepis** (opis słowny):

1. Start
2. Wyświetl pytanie o imię i pobierz je.
3. Wyświetl pytanie o nazwisko i pobierz je.
4. Wyświetl pytanie o stanowisko i pobierz je.
5. Przygotuj dane do wyświetlenia (połącz imię i nazwisko).
6. Wyświetl sformatowaną wizytówkę.
7. Stop

- **Cel**: Zwizualizować nasz algorytm
- **Narzędzia** (podstawowe bloki):

a) Owal (Start/Stop)

b) Równoległobok (Dane Wejścia/Wyjścia - I/O)

c) Prostokąt (Operacja/Proces)

Ćwiczenie 1: Budowa schematu blokowego

Logika schematu (krok po kroku):

Ćwiczenie 2: Most do kodu – Pseudokod

1. Czym jest pseudokod?

- Ustrukturyzowany opis algorytmu w języku naturalnym.
- Używa słów kluczowych, ale ignoruje ścisłą składnię (:, (), "").
- Ostatni, najważniejszy krok przed napisaniem prawdziwego kodu.

2. Nasze słowa kluczowe:

- Nasze słowa kluczowe:
- WYŚWIETL -> operacja wyjścia (jak print)
- POBIERZ -> operacja wejścia (jak input)
- USTAW -> przypisanie wartości do zmiennej (jak =)

3. Nasz algorytm w pseudokodzie:

START

WYŚWIETL "Podaj swoje imię: ”

POBIERZ imie

WYŚWIETL "Podaj swoje nazwisko: ”

POBIERZ nazwisko

WYŚWIETL "Podaj swoje stanowisko: ”

POBIERZ stanowisko

USTAW linia_danych = imie + " " + nazwisko

WYŚWIETL sformatowaną wizytówkę z linia_danych, stanowisko

STOP

Ćwiczenie 3: Rozgrzewka z print()

Cel: Oswojenie się z funkcją print() i edytorem kodu.

Zadanie:

1. W pliku main.py napisz kod, który wyświetli w konsoli Twoje imię i nazwisko

2. W kolejnej linii, wyświetl nazwę swojego kierunku studiów

Ćwiczenie 4: Rysowanie w konsoli (ASCII Art)

Cel: Zrozumienie, jak sekwencja instrukcji print() buduje wieloliniowy obraz.

Zadanie: Odtwórz poniższy rysunek, używając trzech instrukcji print(). Każda linia rysunku to osobna instrukcja.

```
 /\_/\
( o.o )
 > ^ <
```

Ćwiczenie 5: Nasz pierwszy dialog (input i print)

Cel: Nauczyć program słuchać i odpowiadać.

Nowe narzędzie:

input("Twoje pytanie") - zatrzymuje program i czeka na tekst od użytkownika.

Zadanie:

- Użyj **input()**, aby zapytać użytkownika o jego imię. Zapisz odpowiedź w zmiennej.
- Użyj **print()**, aby przywitać użytkownika po imieniu (np. *"Witaj, Anna!"*).
- W kolejnych liniach, zapytaj o ulubiony kolor i skomentuj go.

Od projektu do kodu

Cel: Zaimplementować nasz projekt w całości, używając poznanych do tej pory narzędzi.

Zadanie: Odtwórz poniższą wizytówkę, korzystając z danych, które użytkownik wprowadzi do programu. Skorzystaj z opracowanego wczesniej planu.

```
----------------------------
Jan Kowalski
Programista Back-end
----------------------------
```

Zadanie 1: Kreator Historyjki (Mad Libs)

Cel: Samodzielne przejście przez cały proces projektowy dla nowego problemu.

Problem: Napisz program, który prosi użytkownika o rzeczownik, czasownik i przymiotnik, a następnie wstawia je do zdania:
"Pewnego dnia [przymiotnik] [rzeczownik] postanowił [czasownik] dookoła świata."

Etapy do wykonania:

- Plan (w głowie): Zastanów się, jakie kroki musi wykonać program (jakie pytania zadać, w jakiej kolejności).
- Implementacja (w PyCharm): Przełóż swój plan na działający kod.

Omówienie zadania i Dobre Praktyki

Cel: Zrozumieć, że pierwszy działający kod to dopiero początek.

Refaktoryzacja – Poprawianie struktury kodu bez zmiany jego zewnętrznego działania.

Ćwiczenie Finalne: Mini-Wywiad

Cel: Zaprojektuj i zaimplementuj od A do Z bardziej złożony program interaktywny.

Problem: Stwórz program, który przeprowadzi z użytkownikiem krótki wywiad, a następnie wyświetli podsumowanie.

Kroki do wykonania:

1. **Plan:** Zastanów się nad sekwencją pytań i formatem wyjściowym.
2. **Implementacja:**
    a. Program powinien zapytać o 3 rzeczy: imię, miasto pochodzenia i wymarzony zawód.
    b. Po zebraniu danych, program powinien wyświetlić sformatowane, wieloliniowe podsumowanie, np.:

```
--- PODSUMOWANIE ---
Imię: Anna
Miasto: Gdańsk
Marzenie: Zostać Astronautką.
--------------------
```

Omówienie ćwiczenia: Mini-Wywiad

Analiza przykładowego rozwiązania i dobrych praktyk formatowania

Kluczowe techniki:

- Użycie **n** do stworzenia pustej linii (nowa linia).
- Użycie **f-stringów** do czytelnego łączenia tekstu i zmiennych.
- Konsekwentne formatowanie dla profesjonalnego wyglądu.

Wyzwanie Końcowe: Generator prostego CV

Cel: Utrwalenie umiejętności zbierania wielu danych i formatowania złożonego, wieloliniowego tekstu.

Problem: Stwórz program, który zbierze od użytkownika dane do prostego CV, a następnie je wyświetli.

Wymagane dane:

- Imię i nazwisko
- Adres e-mail
- Numer telefonu
- Krótkie podsumowanie zawodowe (jedno zdanie)
- Trzy kluczowe umiejętności (każda osobno)

```
========================================
CV: Jan Kowalski
========================================
Kontakt:
  - E-mail: jan.kowalski@example.com
  - Telefon: 123-456-789

Podsumowanie:
  Doświadczony programista Python z pasją do czystego kodu.

Umiejętności:
  - Python
  - Git
  - SQL
========================================
```

Omówienie Wyzwania: Generator CV

Przykładowe rozwiązanie.

Podsumowanie

Co dzisiaj zrobiliśmy?

- Nauczyliśmy się, jak rozbijać problem na kroki (myślenie algorytmiczne).
- Zaprojektowaliśmy nasz pierwszy algorytm (schemat blokowy i pseudokod).
- Napisaliśmy kompletny, działający program, który realizuje nasz projekt.
- Przećwiczyliśmy cały proces na samodzielnych zadaniach.

Kluczowa lekcja: Dobry projekt to 90% sukcesu. Kodowanie to tylko tłumaczenie.

Na następnych laboratoriach:

- Damy naszym programom inteligencję (instrukcje warunkowe if).
- Nauczymy je powtarzać zadania (pętle while i for).
- Zaczniemy budować bardziej złożone programy.