

Proiect final

Matei Alexandra Loredana

25.09.2024

Partea teoretica

- **Variabila** – reprezinta o zona din memoria unui calculator care stocheaza anumite valori.

Variabila are nume unic, fiind creata in momentul in care i se atribuie o valoare.

Prin suprascriere, variabilele isi pot schimba valoarea pe parcursul executiei programului. De asemenea, isi pot schimba si tipul de date (type casting). Se poate atribui aceeasi valoare mai multor variabile, sau, se pot atribui, mai multe valori unei variabile. Numele variabilelor se scrie cu litere mici.

```
# declarare si initializare variabila  
fruct = 'mar'
```

- **Constanta** – reprezinta, asemeni variabilei, un identificator pentru o valoare, dar, spre desosebire de variabile, valorile constantelor nu se pot schimba in timpul rularii programului. Ele sunt utile pentru a mentine o valoare constanta in intreaga aplicatie. Prin conventie, numele constantelor sunt mereu scrise cu majuscule.

```
# declarare si initializare constanta  
PI = 3.14
```

- **Tipuri de date:**

- **int** – numar intreg

```
# initializare variabila de tip int  
numar = 5
```

- **float** – numar zecimal

```
# initilizare variabila de tip float  
numar_zecimal = 8.65
```

- **bool** – adevarat/fals

```
# initializare variabila de tip boolean  
valoare_adevarata = True
```

- **string** – sir de caractere de la tastatura, delimitate de ' ' sau " "

```
# initializare variabila string  
string = 'curs'
```

- **Conditionalul if-elif-else**

- **If:** reprezinta o declaratie in cod prin care programul executa o bucata de cod in functie de o

conditie/ un criteriu. Dupa cele : ale unei ramuri, urmate de 'Enter' se va realiza indentarea, cu scopul de a transmite programului de unde pana unde tine blocul de cod corespunzator acelei ramuri. Codul se executa doar in situatia in care conditia data la 'if' este evaluata ca fiind adevarata.

- **If-elif-else:** se foloseste atunci exista mai mult de doua situatii posibile, se evalueaza conditiile de sus in jos si se executa codul aferent primei conditii adevarate.

```
# Pretul biletului la film in functie de varsta  
varsta = 15  
if varsta >=12 and varsta <=18:  
    print("Poti viziona acest film fara plata biletului."  
elif varsta >= 18:  
    print('Pretul biletului este 25lei.'  
else:  
    print('Nu poti viziona acest film.'  
# -> Poti viziona acest film fara plata biletului.
```

- **Structuri de date**

➤ **Lista:** reprezinta o colectie de elemente, separate prin virgula si definita folosind []; se pot pastra diferite tipuri de date in aceeași lista.

Lista este ordonata (elementele fiind pastrate in ordinea in care au fost adaugate), indexata (fiecare element din lista are index, care incepe de la 0) si mutabila (se pot adauga, sterge si modifica elemente). De asemenea, valorile pot fi duplicate intr-o lista. Lungimea unei liste (cate elemente exista in lista) se poate afla cu ajutorul functiei len().

```
lista1 = [1, 2, 3, 'nume', True, 'prenume', True]
print(lista1) # -> [1, 2, 3, 'nume', True, 'prenume', True]
```

➤ **Dictionarul:** reprezinta o colectie de perechi cheie:valoare; se defineste folosind {} si are structura generala {cheie1:val1, cheie2:val2}.

Dictionarul este ordonat, mutabil, iar cheile sunt unice (nu pot exista chei duplicate). Se poate folosi functia len() pentru a afla dimensiunea unui dictionar.

```
ex_dictionar = {
    'prenume': 'Alexandra',
    'varsta': 30
}
print(ex_dictionar['varsta']) # -> 30 # accesare valoarea aferenta cheii 'varsta'
```

➤ **Setul:** reprezinta o colectie de elemente unice, separate prin virgula; se defineste folosind {}.

Setul nu este ordonat (elementele sale nefiind pastrate in ordinea in care au fost adaugate), nu este indexat (nu se pot accesa elementele dupa index). Este mutabil (se pot adauga si sterge elemente), dar elementele din set sunt imutabile (nu pot fi modificate).

Dimensiunea unui set se poate afla cu ajutorul functiei len().

```
ex_set = {10, 'element', True, 20}
print(ex_set) # -> {'element', 10, 20, True}
```

➤ **Tuplul:** pastreaza mai multe valori imutabile intr-o singura variabila, elementele fiind separate prin virgula; se defineste folosind ().

Tuplul este ordonat, indexat si imutabil (elementele sale nu pot fi modificate, adaugate sau sterse); accepta valori duplicate.

Dimensiunea unui tuplu se afla folosind functia len().

```
ex_tuplu = (1, 2, 3, 'element', True)
print(ex_tuplu) # -> (1, 2, 3, 'element', True)
```

- **Funcția:** reprezintă un bloc de cod care se execută doar atunci când este apelată.

Funcția se poate apela în cod ori de câte ori este nevoie, eliminând astfel copierea în program, aceasta fiind și utilitatea ei principală: este reutilizabilă.

Există două tipuri de funcții:

- funcții built-in (disponibile pentru utilizare: print, sum, max etc.)
- funcții definite de utilizator — se pot crea propriile funcții care să îndeplinească propriile nevoi.

Atunci când o funcție este apelată, programul se duce la definiția funcției, are loc executarea codului din interiorul ei, iar apoi programul se continuă de la linia de după apelarea funcției.

```
# exemplu definire functie
1 usage
def function_name():
    print('Bine ati venit!')
# apelare functie
function_name() # -> Bine ati venit!
```

- **Parametrul:** reprezintă datele de intrare(input) într-o funcție.

O funcție poate avea unul sau mai mulți parametri, aceștia fiind opționali; practic, ei reprezintă variabile declarate, dar neinitializate, fiind initializate la apelarea funcției.

```
# exemplu functie cu parametri
1 usage
def suma_nr(a, b):
    print(f'Suma: {a + b}')
suma_nr(a: 2, b: 3) # -> Suma: 5
```

- **Clasa:** este o reteta (un blueprint) pentru crearea obiectelor; contine elemente descriptive(campuri) si actiuni posibile (metode), ea defineste comportamentul si starea obiectelor. O clasa se creeaza folosind cuvantul cheie “class” urmat de denumirea clasei (scrisa cu cuvinte capitalizate).

Dupa definirea clasei, se pot crea obiecte de tipul respectiv.

```
class Produs:  
    # fields-uri  
    nume = 'Trotineta'  
    pret = 200  
    stoc = 3  
    # metode  
    def functioneaza(self):  
        print('Functionala!')
```

- **Obiectul:** este o instanta a clasei. Toate obiectele dintr-o clasa vor avea acelasi comportament (campuri, metode).

Dupa initializarea obiectului, attributele pot suferi modificari.

```
produs1 = Produs()  
print(produs1.nume) # -> Trotineta
```

➤ **Selectorul** – reprezinta un sir de caractere cu rolul de a identifica unul sau mai multe elemente intr-o pagina web, avand scopul de a interactiona cu ele in procesul de automatizare.

➤ Tipuri de selectori:

- ID – identificator unic pentru elemente ;
- Class – selector care de cele mai multe ori nu este unic, fiind necesara combinatia sa cu alti selectori pentru a identifica elementele in mod unic;
- Name – identifica elemente pe baza perechilor de atribut=valoare (valoarea reprezinta elementul);
- Link Text – cautarea se face dupa textul pus peste link;
- Partial Link Text – cautarea se face dupa o parte din textul pus peste link;
- CSS Selector – sir de caractere folosit pentru identificarea elementelor;
- XPATH – se foloseste, in general, atunci cand niciuna din celelalte metode de identificare nu a ajutat la identificarea unui element in mod unic.

Alegerea unui selector depinde de elementul cautat, in general, se foloseste cea mai scurta si eficienta metoda de a identifica un element web.

- **TDD (Test Driven Development):** este un process implementat, de obicei, de catre echipa de dezvoltare, folosind testele unitare care vor testa ceea ce se presupune ca ar trebui sa faca sistemul. Practic, cerintele de business sunt transformate in teste inainte de a avea codul dezvoltat. Apoi se vor crea liniile de cod necesare pentru functionalitate.

➤ Avantajele utilizarii TDD:

- se creeaza minimul de cod necesar implementarii unei functionalitati;
- se concentreaza pe teste, asigurand astfel o acoperire mai mare a aplicatiei prin intermediul lor; aplicatia devine mai apropiata de nevoile clientului;
- intretinerea usoara a codului.

➤ Testarea unitara (testare de componente) reprezinta un proces prin care fiecare parte din cod este testata in vederea verificarii ei pentru utilizare.

Testul unitar reprezinta testarea celei mai mici parti dintr-o aplicatie (functii, clase, proceduri).

- **BDD:** reprezinta un proces de dezvoltare software, derivat din TDD, fiind bazat insa pe o atentie mai mare asupra scenariilor de testare.

➤ **Avantajul utilizarii BDD:** peste codul de testare automata, se adauga fisiere descriptive ale scenariilor de business (feature files) fiind scrise intr-un limbaj inteles si de utilizatori care nu au cunostiinte tehnice.

Fisierele descriptive sunt primele care se creeaza in procesul de BDD, fiind creat ulterior codul pentru a valida testele descrise.

➤ **Sintaxa Gherkin:** limbajul in care sunt scrise fisierele descriptive; in python, limbajul Gherkin se implementeaza prin intermediul libreriei behave.

Cele mai folosite elemente in limbajul Gherkin sunt: Given, When, Then, And, Scenario. Cu ajutorul sintaxei Gherkin, se definesc scenariile care urmeaza a fi automatizate.

- **API (Application Programming Interface):** reprezinta un set de reguli si protocoale care permit aplicatiilor software sa comunice intre ele; permite comunicarea intre client si server.

In practica, API-urile se utilizeaza pentru a permite aplicatiilor din partea clientului sa aiba acces la o baza de date.

API-urile stabilesc comunicarea intre doua aplicatii: ele nu pot comunica direct pentru ca nu au acces una la codul celeilalte, astfel ca este necesar sa se transmita semnale intre ele. Modul in care trebuie transmise aceste semnale, este definit de API.

- **HTTP (Hypertext Transfer Protocol):** reprezinta un protocol de comunicare intre client si server.

➤ Principalele metode HTTP:

- GET: este cea mai folosita metoda, atunci cand serverului i se cere o resursa;
- POST: prin intermediul acestei metode se trimit date catre server pentru a crea resurse noi;
- PUT: actualizarea unei resurse;
- PATCH: actualizarea partiala a unei resurse;
- DELETE: metoda prin care se sterg date de pe server.

Va multumesc!