Proiect pentru obținerea certificatului de competență profesională

Tema

Sudoku Solver

Nume și prenume: Dumitru Sebastian

Clasa a XII-a D

Profesor Coordonator: Ghiordunescu Grațiela

**Cuprins:**

-Motivația alegerii temei

-Introducere

-Noțiuni de bază

-Descrierea softului utilizat

-Structura aplicației

-Resurse hardware și software necesare

-Concluzii

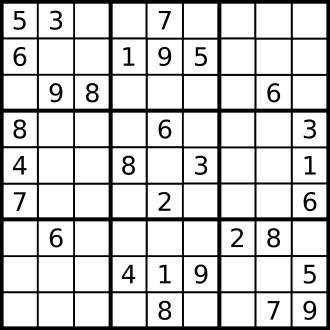
-Bibliografie

**Motivația alegerii temei:**

Tema acestui proiect la informatică este -Sudoku Solver-

Am ales aceasta temă deoarece mereu mi-am dorit să creez un program funcțional utilizând alt limbaj de programare decât C++, însă nu am avut niciodată motivația necesară să duc proiectul până la final.

Această dorință de a crea un program combinată cu pasiunea mea pentru Sudoku, un joc în formă de grilă inventat în 1979 și inspirat de pătratul latin și de problema celor 36 ofițeri a lui Leonhard Euler, a dus la realizarea acestui proiect.



**Introducere:**

Python este un limbaj de programare dinamic [multi-paradigmă](https://ro.wikipedia.org/wiki/Limbaj_de_programare_multiparadigm%C4%83), creat în [1989](https://ro.wikipedia.org/wiki/1989) de programatorul olandez [Guido van Rossum](https://ro.wikipedia.org/w/index.php?title=Guido_van_Rossum&action=edit&redlink=1). Van Rossum este și în ziua de astăzi un lider al comunității de dezvoltatori de software care lucrează la perfecționarea limbajul Python și implementarea de bază a acestuia, CPython, scrisă în [C](https://ro.wikipedia.org/wiki/C). Python este un limbaj multifuncțional folosit de exemplu de către companii ca [Google](https://ro.wikipedia.org/wiki/Google) sau [Yahoo!](https://ro.wikipedia.org/wiki/Yahoo!) pentru programarea aplicațiilor web, însă există și o serie de aplicații științifice sau de divertisment programate parțial sau în întregime în Python. Popularitatea în creștere, dar și puterea limbajului de programare Python au dus la adoptarea sa ca limbaj principal de dezvoltare de către programatori specializați și chiar și la predarea limbajului în unele medii universitare. Din aceleași motive, multe sisteme bazate pe Unix, inclusiv [Linux](https://ro.wikipedia.org/wiki/Linux), [BSD](https://ro.wikipedia.org/wiki/BSD) și [Mac OS X](https://ro.wikipedia.org/wiki/Mac_OS_X) includ din start interpretatorul CPython.

Python pune accentul pe curățenia și simplitatea codului, iar sintaxa să le permite dezvoltatorilor să exprime unele idei programatice într-o manieră mai clară și mai concisă decât în alte limbaje de programare ca [C](https://ro.wikipedia.org/wiki/C). În ceea ce privește paradigma de programare, Python poate servi ca limbaj pentru software de tipul object-oriented, dar permite și programarea imperativă, funcțională sau procedurală. Sistemul de tipizare este dinamic iar administrarea memoriei decurge automat prin intermediul unui serviciu „gunoier” (garbage collector). Alt avantaj al limbajului este existența unei ample biblioteci standard de metode.

Implementarea de referință a Python este scrisă în C și poartă deci numele de CPython. Această implementare este software liber și este administrată de fundația Python Software Foundation.

**Noțiuni de bază:**

**Tipizare**

Limbajele dinamice sunt deseori caracterizate prin tipizare dinamică, sunt interpretate, au management de memorie automatizat (garbage collecting) și au un nivel înalt de abstractizare. Tipul variabilelor nu este fixat la momentul declarării acestora (de fapt, declararea variabilelor nu este necesară ca în C), ci este determinat de interpretator după conținutul lor sau după operațiile efectuate. Într-un asemenea caz se vorbește de tipizare dinamică.

Avantajul acestei metode de a lucra cu tipuri este că codul produs este mai clar și mai intuitiv. De exemplu:

***a = 10 # a va fi considerat un număr întreg (integer)***

***b = 11.5 # b va fi considerat un număr rațional (double sau float)***

***c = "un rând de text" # c va fi considerat un șir (string)***

În ciuda tipizării dinamice, Python este puternic tipizat, adică interpretatorul nu permite operații cu obiecte de tip diferit dacă acestea nu sunt bine definite, cum ar fi operațiile de tip cast în C. Există totuși unele excepții logice, cum ar fi în cazul numerelor întregi și celor raționale (float), între care sunt permise operații ca adunarea sau împărțirea, tipul rezultatului ținând cont de natura operației, cum se poate vedea mai jos:

***d = 10 + 11.5 # rezultatul va fi un număr rațional, 21.5***

***e = "o brioșă" + 4 # această comandă va genera o eroare.***

Alt concept important în Python este cel al tipurilor mutabile și nemutabile. Așa cum implică numele, datele cu un tip mutabil pot fi alterate după inițializare, în vreme ce la date nemutabile lucrul acesta este imposibil. Luând ca exemplu o listă (mutabilă), constatăm că putem adăuga elemente la listă sau le putem modifica fără probleme. În cazul unui șir de caractere (nemutabil), acest lucru nu este permis:

***lista\_mea = ["măr", "pară"]***

***lista\_mea.append("strugure") # adaugă la listă***

***print lista\_mea[2] # afișază: strugure***

***șirul\_meu = "varză"***

***șirul\_meu[0] = "b" # schimbă cuvântul în "barză" -> Eroare***

### **Structuri de date**

Python oferă tipuri tradiționale de date, cum ar fi numărul întreg (integer, int) sau cel rațional (float), dar introduce totodată și concepte noi.

De exemplu, un grad mare de flexibilitate îl oferă listele (tablourile) în Python. Acestea nu sunt statice, ci pot conține orice tipuri de date (în cadrul aceleiași liste) și pot fi modificate pe loc adăugând și eliminând elemente fără a declara sau utiliza funcții de manipulare a memoriei.

***lista\_mea = [10, "șir de caractere", variabilă, ["altă", "listă"], chiar\_și\_o\_funcție]***

Alte structuri de date sunt tuplurile/perechile (tuples) și dicționarele (dictionaries, mappings). Tuplurile sunt liste care au un număr prestabilit de elemente, și nu pot fi modificate parțial. Tuplurile pot fi utilizate în cazuri în care este nevoie de o anumită structură de date specializată, de exemplu coordonate în spațiul cartezian. Dicționarele sunt liste neordonate în care fiecare element are asociat o cheie, care poate fi număr sau șir de caractere. Dicționarele au foarte multe aplicații, inclusiv crearea structurilor de tip hash-tables.

### **Stil**

Python este un limbaj multi-paradigmă, concentrându-se asupra programării imperative, orientate pe obiecte și funcționale, ceea ce permite o flexibilitate mai mare în scrierea aplicațiilor. Din punctul de vedere al sintaxei, Python are un număr de construcții și cuvinte cheie cunoscute oricărui programator, dar prezintă și un concept unic: nivelul de indentare are semnificație sintactică. Blocurile de cod sunt delimitate prin simplă indentare.

În C un astfel de blocuri sunt deseori desemnate prin acolade, {<cod>}, dar în Python nu este nevoie de astfel de construcții. Nivelele de indentare îndeplinesc această funcție. Această importanță a indentării este foarte suprinzătoare pentru mulți utilizatori noi ai limbajului Python, chiar dacă sunt programatori cu experiență. Dar o astfel de utilizare a indentării permite codului să fie mai ușor de citit și mai compact. Programatorii cu experiență vor indenta implicit codul sursă, oricare ar fi limbajul, fiindcă astfel se permite structurarea codului sursă și evidențierea funcționalității. Python face din această deprindere folositoare în acest sens o cerință strictă.

O impunere similară există și în limbajul de programare Java, care forțează programatorii să delimiteze clasele în fișiere aparte, din motive de organizare și sporire a eficienței de scriere a softului în echipe.

**Biblioteci și Extindere**

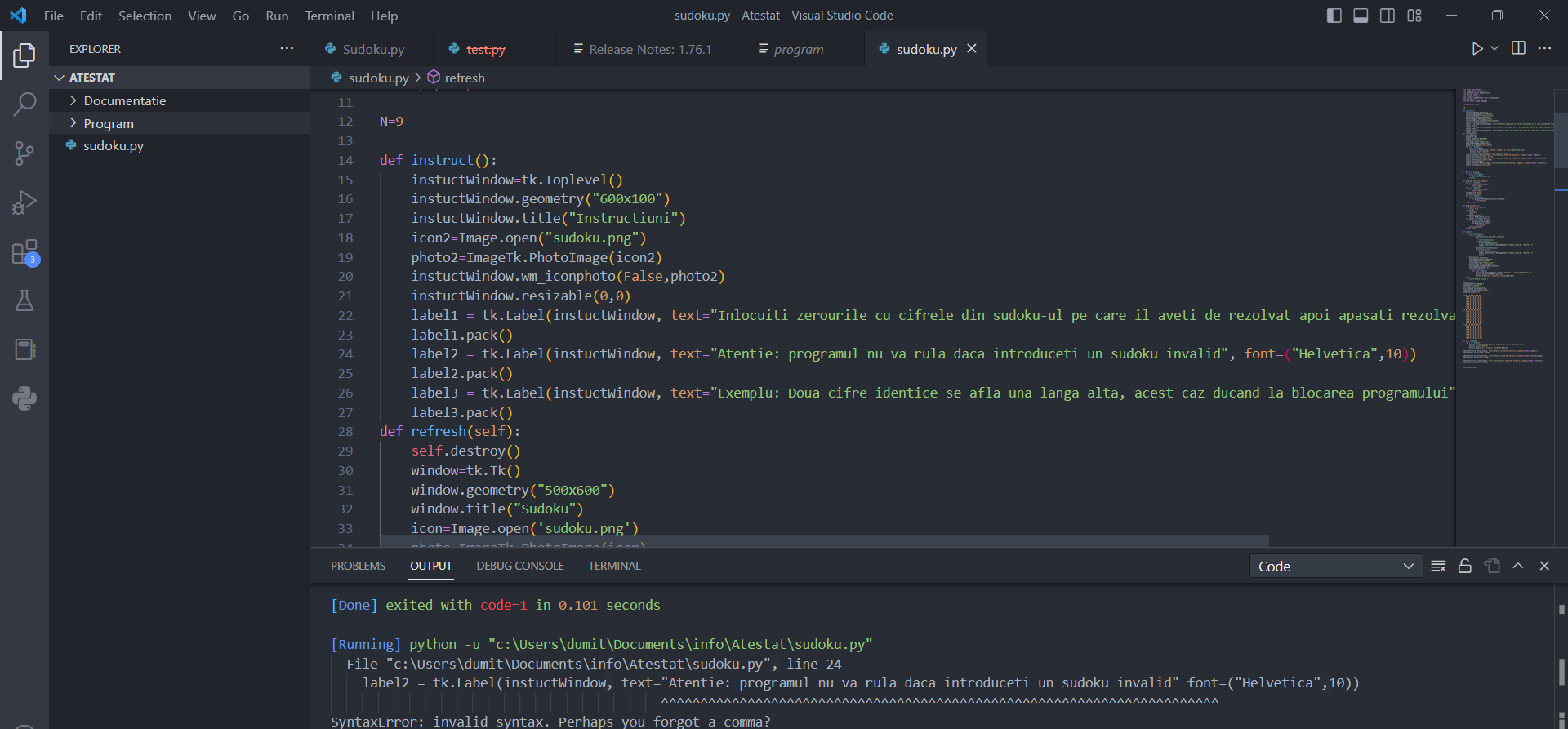
Includerea tuturor acestor structuri, precum și a funcțiilor ce permit manipularea și prelucrarea lor, precum și multe alte biblioteci de funcții sunt prezente datorită conceptului “Batteries Included”, ce poate fi explicat prin faptul că Guido van Rossum și comunitatea ce s-a format în jurul limbajului cred că un limbaj de programare nu prezintă utilitate practică dacă nu are un set de biblioteci importante pentru majoritatea dezvoltatorilor.

Din acest motiv Python include biblioteci pentru lucrul cu fișiere, arhive, fișiere XML și un set de biblioteci pentru lucrul cu rețeaua și principalele protocoale de comunicare pe internet (HTTP, Telnet, FTP). Un număr mare de platforme Web sunt construite cu Python. Abilitățile limbajului ca limbaj pentru programarea CGI sunt în afara oricăror dubii. De exemplu YouTube, unul din site-urile cu cea mai amplă cantitate de trafic din lume, este construit pe baza limbajului Python.

Totuși, Python permite extinderea funcționalității prin pachete adiționale programate de terți care sunt axate pe o anumită funcționalitate. De pildă, pachetul wxPython conține metodele și structurile necesare creării unei interfețe grafice.

**Descrierea softului utilizat:**

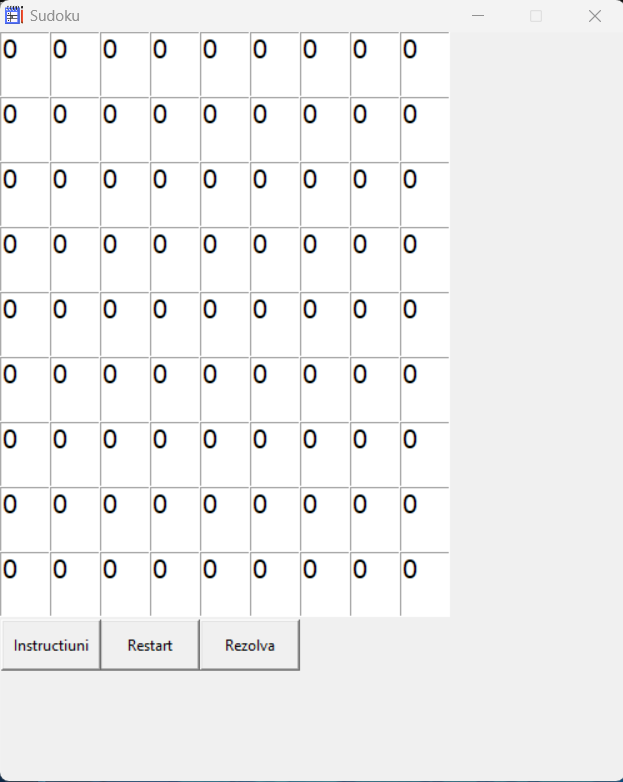
Visual Studio Code (VS Code) este un editor de cod sursă ușor, dar puternic, care rulează pe desktop și este disponibil pentru Windows, macOS și Linux. Acesta vine cu asistență încorporată pentru JavaScript, TypeScript și Node.js și are un ecosistem bogat de extensii pentru alte limbi (cum ar fi C+, C#, Java, Python, PHP și Go) și runtimes (cum ar fi .NET și Unity).



**Structura aplicației:**

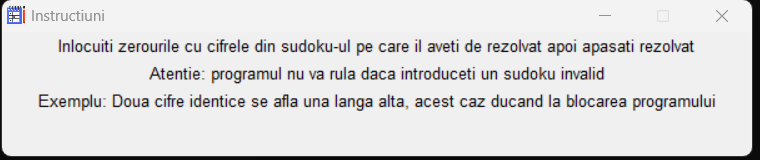
Aplicația a fost realizată în totalitate în Python 3.0 în aplicația Visual Studio Code.

GUI-ul a fost realizat cu ajutorul extensiei Tkinter.



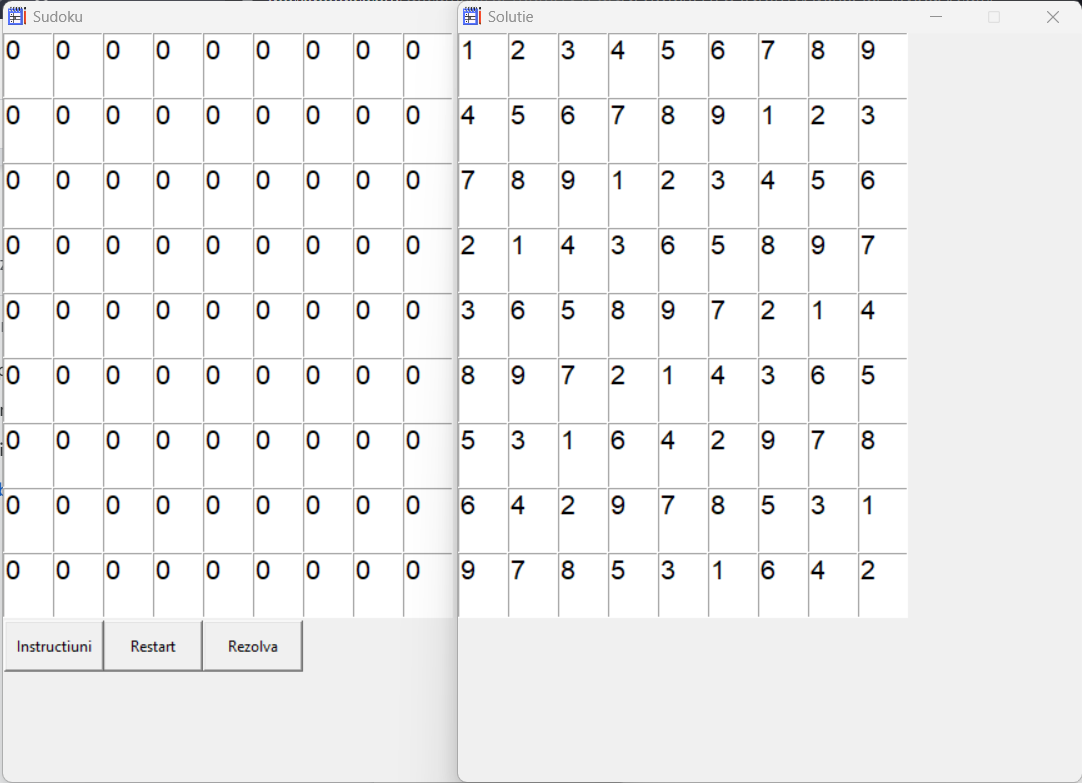
Interfața este una simpla, fiind compusă din 81 căsuțe care au rolul de a stoca cifrele sudokului care urmează a fi rezolvat, și trei butoane cu următoarele roluri:

-Instrucțiuni: Deschide o fereastră cu instrucțiuni și restricții cu privire la utilizarea programului;



-Restart: Reseteaza programul, constituind o metodă simplă de a șterge cifrele deja introduse și a rezolva un sudoku nou;

-Rezolvă: Așa cum sugerează numele, butonul deschide o nouă fereastră cu soluția pentru sudokul introdus, sau o eroare dacă cifrele introduse sunt necorespunzătoare (userul a introdus litere sau cifre în afara intervalului 1-9)



**Resurse hardware și software necesare:**

Programul nu necesită un PC modern pentru a rula. Este necesar un computer cu un procesor Intel Pentium sau AMD Athlon, GPU integrat și cel puțin 2 GB RAM.

**Concluzii:**

Este foarte folositor să cunoaștem mai multe limbaje de programare și crearea acestui atestat a fost momentul potrivit pentru a-mi îmbunătăți cunoștințele despre Python, un limbaj de programare foarte versatil, un limbaj de programare care îmi va fi util în viitoarele mele proiecte.

**Bibliografie:**

->Pentru partea de proiectare:

-<https://www.w3schools.com/python/default.asp>

-<https://www.youtube.com/watch?v=XKHEtdqhLK8>

->Pentru documentație:

-<https://code.visualstudio.com/docs/editor/whyvscode>

-<https://learn.microsoft.com/ro-ro/power-apps/maker/portals/vs-code-extension>

-<https://ro.wikipedia.org/wiki/Python>

-<https://ro.wikipedia.org/wiki/Sudoku>

-https://en.wikipedia.org/wiki/Tkinter