**State of art**

**Recognizing time in texts**

Aungurenci Andrei

Braneanu Valentin

Rauleac Robert

Senciuc Paul

Trufanda Bogdan

1. Tehnologia folosita

Am decis sa alegem ca mediu de lucru limbajul **Python**, datorita usurintei folosirii lui fata de alte limbaje de programare, de exemplu C++, Java sau C#. Pentru a intarii acest argument, priviti figura 1.

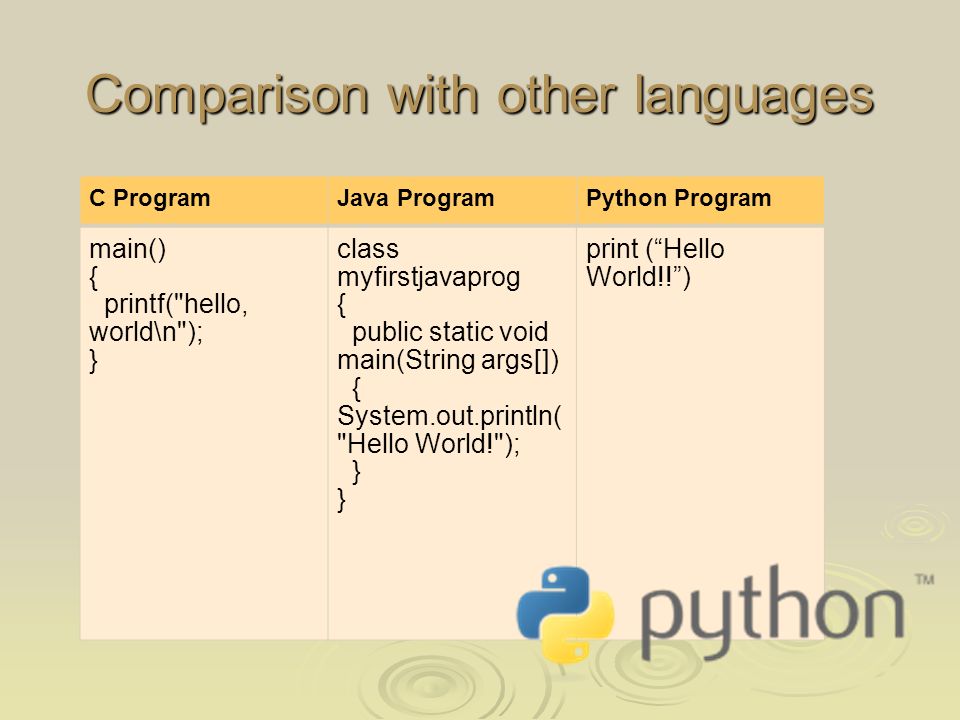


Fig 1.

Dupa cum se poate observa, in cele 3 coloane se intampla acelasi lucru: se afiseaza in consola “Hello World!”. In cazul limbajului Python, pentru afisarea unui text in consola tot ce este nevoie este sa folosim functia print() si atat, spre deosebire de C sau Java unde este nevoie de mult mai mult cod de scris.

Vom folosi Python 3.8.1 x64 (aceasta fiind ultima versiune la momentul actual) si ca IDE PyCharm deoarece are mai multe functionalitati fata de un simplu editor de text, precum Notepad++ sau Sublime text. Cat despre IDE, am ales sa folosim Pycharm de la cei de la JetBrains si nu alte IDE-uri precum Spyder, PyDev sau Eclipse deoarece avem licenta gratuita la el fiindca suntem studenti si am lucrat cu el la cursul de Python din semenstrul 1.

1. Librarii folosite

2.1 (Re – extragerea expresiilor regulate)

Pentru extragerea unor secvente de text vom folosi expresii regulate. In python exista 2 librarii care fac acest lucru si anume: Re si Regex. Libraria Re este libraria oficiala, cea oferita de cei de la Python, iar Regex este o librarie dezvoltata de o comunitate de developeri pasionati de Python. Aceasta librarie mai adauga unele functionalitati in plus fata de libraria Re. Insa, noi vom folosi libraria standard Re intrucat am lucrat cu ea la cursul de Python si avem mai multa experienta. Atasam un exemplu in figura 2 pentru a exemplifica usurinta folosirii librariei Re.

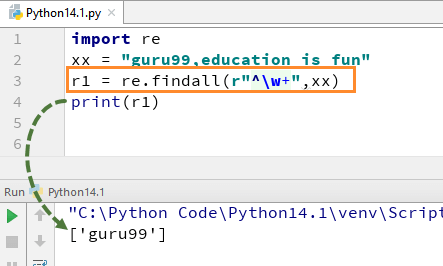


Fig 2.

2.2 XML

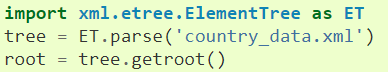
Pentru a adnota expresiile regulate intr-un XML ne folosim de xml.etree.ElementTree, acesta fiind un modul ce implementeaza un API simplu si eficient ce serveste la parsarea si crearea de date XML.

XML este forma de date inerent ierarhic cea mai naturala in reprezentarea unui arbore. ET are doua clase ce ne poate ajuta in acest sens, prima fiind ElementTree, ce reprezinta tot documentul XML ca un arbore, iar a doua fiind Element, ce reprezinta doar un singur nod in arbore. In general interactiunea cu documentul se realizeaza la nivelul ElementTree. In general interactiunea cu un singur element XML sau sub-elementele acestuia se realizeaza la nivelul Element.

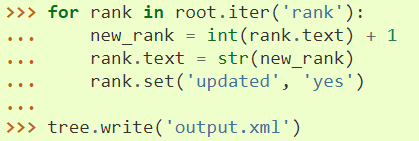
Vom avea ca suport pentru exemplificarea diferitor operatii urmatorul document XML :



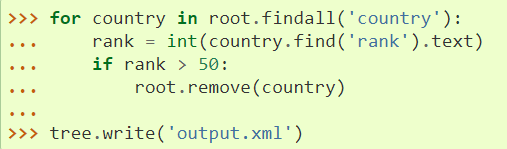
O metoda de parsare se poate realiza prin importarea datelor citite din fisier astfel :



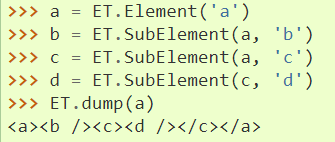
Modificarile documentului XML se pot realiza prin metoda ElementTree.write() oferita de ElementTree ce ne permite sa alcatuim documente XML si sa le scriem in fisiere. Atunci cand un este creat un obiect Element, acesta poate sa fie manipulat schimband fildul (de exemplu Element.text), astefl putem sa adaugam sau sa modificam atribute (Element.set()), sau sa adaugam noi copii (Element.append()). Un astfel de exemplu il putem observa in imaginea de mai jos :



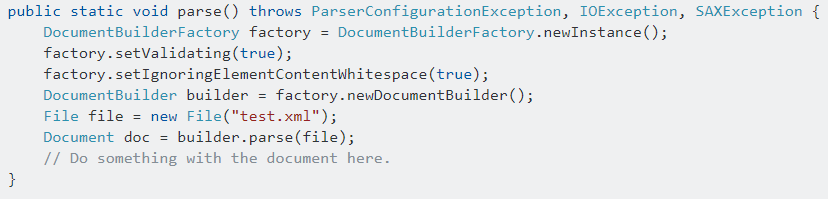
De asemenea, putem sa si eliminam diferite elemente cu functia Element.remove(), ca si in exemplul de mai jos :

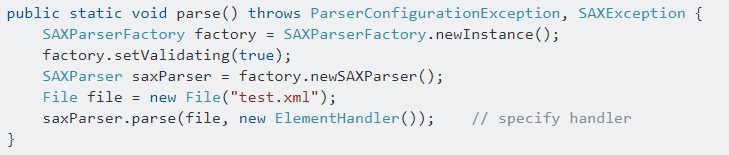


De asemenea, pentu a crea un sub-element unui element ales ne folosim de metoda SubElement(), ca si in exemplul de mai jos :



Alegerea noastra de a adnota expresiile regulate intr-un XML folosindu-ne de libraria xml.etree.ElementTree din limbajul Python este datorata faptului ca avem mai multa experienta folosind acest limbaj, cat si atingerea scopului final intr-un timp cat mai scurt decat prin folosirea altor limbaje sau librarii precum DOM sau SAX din JAVA, Xerces-C++ din C++. De asemenea suportul gasit pe internet fiind unul mult mai vast pentru acest limbaj, totodata cantitatea de cod necesara pentru a realiza anumite operatii in cadrul limbajului Python este una cu mult redusa fata de alte libaje. De exemplu aceasta reprezinta un mod de parasare a unui document XML in limbajul JAVA folosind DOM si SAX :





2.3 PySimpleGUI si Tkinter (pentru interfata grafica)

**Tkinter** este inclus in libraria standard din Python fiind documentat in amanunt si punand la dispozitie multe exemple de creare si utilizare a widgeturilor.

Posibile resurse: [Python Documentation](https://docs.python.org/3.8/library/tkinter.html), [TkDocs](https://tkdocs.com/tutorial/index.html), [PythonWikiTkinter](https://wiki.python.org/moin/TkInter).

**PySimpleGUI** este un wrapper pentru mai multe module de interfata grafica cum ar fi Tkinter sau Qt. Un avantaj deosebit este simplitatea si cantitatea mica de cod necesar pentru a configura interfata dorita.

In figura 3 prezentam un program ce afiseaza calea catre un fisier ales de utilizator.

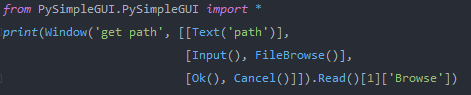


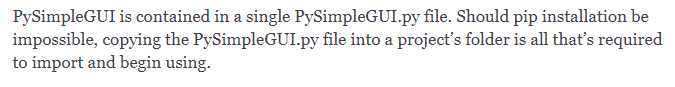
Fig 3.

Rezultatul este cel din figura 4.



Fig 4.

Conform [The Hitchhiker's Guide To Python](https://docs.python-guide.org/scenarios/gui/#pysimplegui):



Python ofera mai multe alternative pentru interfata grafica, cateva dintre ele fiind(**Kivy, PyQT, WxPython, PySide** ). Alegerea noastra de a folosi PySimpleGui si Tkinter vine in urma experientei pe care o avem folosind aceste librarii si putem rezolva cerintele acestei aplicatii cu ususinta folosind aceste librarii.

Fig 5.

Posibile resurse: [PySimpleGUIDocumentation](https://pysimplegui.readthedocs.io/en/latest/), [PySimpleGUIGithub](https://github.com/PySimpleGUI/PySimpleGUI).

1. Cerintele problemei

Scrieți o gramatică de expresii regulate capabilă să adnoteze într-un XML, în maniera TimeML, expresii temporale în texte.

Exemple de expresii temporale: mâine, după două zile, de luni până vineri, săptămâna trecută, acum un an, la anul, în secolul al XIX-lea, la mijlocul deceniului ‘90, 16.12.2019, ieri la ora 12:00, în data de 20 a lunii, în primul weekend al fiecărei luni etc.

1. Abordarea

Initial, citim textul din fisier intr-un string. Apoi vom extrage utilizandexpresiile temporale pe care le vom stoca intr-un dictionar in functie de tipul de expresie, iar mai apoi vom construi cu ajutorul librariei xml fisierul TimeML.

Bonus : Vom face un GUI.

1. Task-uri
   * Rauleac – Adnoteze intr-un XML
   * Senciuc, Aungurenci – Inerterfata grafica
   * Trufanda, Braneanu – Extragere expresii temporale