

СОФТУЕР ЗА ПОДПОМАГАНЕ НА ПРОЦЕСА НА ОБУЧЕНИЕ ПО ПРОГРАМИРАНЕ

Стойчо Стоев¹, Пресиян Терзиев²

¹ Икономически университет – Варна/Катедра „Информатика“, Варна, България, s.stoev@ue-varna.bg

² Икономически университет – Варна/Катедра „Информатика“, Варна, България,
p.terziev@ue-varna.bg

РЕЗЮМЕ

Обучението по програмиране е свързано с изразходване на много интелектуална енергия, както при представяне на нов материал, така и при контролиране на придобитите знания. В този смисъл всяка възможност за облекчаване на тази дейност би подпомогнала процеса. За целта представяме приложение, като инструмент за анализ на сходни решения. С него преподавателят има възможност да отсее част от заданията, необходими за проверка и анализ.

КЛЮЧОВИ ДУМИ: *programing, python*

SOFTWARE TO SUPPORT THE PROGRAMMING LEARNING PROCESS

Stoycho Stoev¹, Presian Terziev²

¹ University of economics – Varna/Department of informatics, Varna, Bulgaria, s.stoev@ue-varna.bg

² University of economics – Varna/Department of informatics, Varna, Bulgaria, p.terziev@ue-varna.bg

ABSTRACT

Teaching programming involves spending a lot of intellectual energy, both when presenting new material and when controlling the acquired knowledge. In this sense, any possibility to ease this activity would help the process. For this purpose, we present an application as a tool for analyzing similar solutions. With it, the teacher has the opportunity to sift out part of the assignments necessary for checking and analysis.

KEYWORDS: *programming, python*

ВЪВЕДЕНИЕ

Процесът на обучение по програмиране предполага самостоятелно решаване на задачи с цел, обучаемият да натрупа умения по създаване на код, отстраняване на грешки и анализиране на резултатите. Тази дейност не винаги е свързана с присъствие на студентите в рамките на университета, а може да се извършва и дистанционно. Умението за програмиране, не се отдава в еднаква степен за всички студенти. Следователно съществува възможността за копиране на чуждо решение или генерирането му с помощта на AI. Проверката на задачите е трудоемък процес, при който всеки инструмент за облекчаването му би бил полезен. В този смисъл предлагаме софтуер за неговото подпомагане при филтриране на предадените решения.

1. ПРЕДПОСТАВКИ ЗА ИЗПОЛЗВАНЕ НА СОФТУЕР ЗА ПОДПОМАГАНЕ НА ПРОЦЕСА НА ОБУЧЕНИЕ ПО ПРОГРАМИРАНЕ

В рамките на една дисциплина по програмиране студентите имат възможност да покажат своите знания и умения чрез 6 домашни и 2 контролни. Домашните работи се предават дистанционно в рамките на една седмица, което понякога води до наличие на еднакви работи от

различни студенти. За да се подпомогне процесът на оценяване на предадените решения е необходимо да се редуцират тези, които са с еднакво или много подобно съдържание. Не е изключение и при контролните задания да съществуват решения със сходство.

Таблица 1

Предадени решения за оценяване в рамките на един семестър

Задание	Решения
Домашно 1	104
Домашно 2	90
Домашно 3	79
Контролно 1	110
Домашно 3	95
Домашно 4	86
Домашно 5	75
Контролно 2	115
	754

Източник: Собствена разработка

От таблица 1 е видно, че при голям брой предадени работи, вероятността за еднакво съдържание нараства, още повече, че всяко задание има отношение към текущият контрол. Отстраняването им, ще облекчи работата на преподавателя.

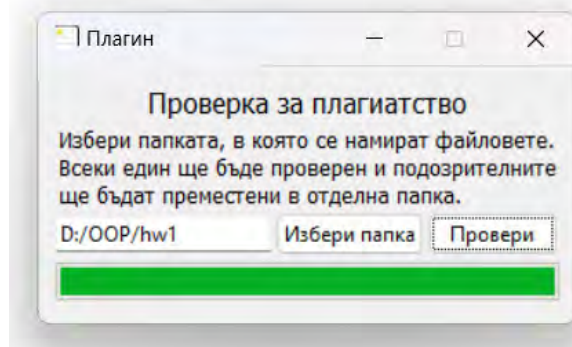
Приложението е подходящо в областта на обучението по програмиране, по няколко причини:

- Решенията на задачите се предават в текстови файлове без допълнителни таблици, обекти, изображения или различни формати. Тоест отпадат проблемите с различните версии на документите, анализирането на различни от текст елементи или смисловото им съдържане;
- В зависимост от задачите, разликите в решенията може да са малки. Следователно приликите в отделните файлове, като съотношение трябва да се съобрази с големината на възможния програмен код. Тоест да не се търси 100% съвпадение, а варианти като 90%, 95% и други;
- Програмният код е съхранен в текстови файлове или лесно може да се пренесе в такива. Това улеснява използване на код за тяхното сравнение, като се анализират отделните редове в различните файлове. Във всяка среда за разработка на софтуер има инструменти за обработка на текстови файлове, което улеснява изграждане на софтуер за сравнение;
- Към анализа на предадените решения може да се включи код генериран от AI. Все по-често някои студенти се изкушават да използват AI за намиране на готови решения на възложените задачи. В този смисъл към анализа може да се включи и такъв вариант, за да се провери има ли сред предадените генериран код.

Следователно, софтуерът е необходимо да обработва стандартна текстова информация и да предлага възможност за обработка на резултатите.

2. ПРИМЕРНО РЕШЕНИЕ ЗА СОФТУЕР ЗА СРАВНЕНИЕ

Разработено е приложение „Плагин“ за сравнение на текстови файлове, с търсене на процентно съвпадение. То е разработено на Python, версия 3.6 и работи автономно, без необходимост от допълнителни инсталации. Интерфейсът на приложението е показан на фиг.1.



Фигура 1. Интерфейс на приложението

Източник: Собствена разработка

Софтуерът предлага следните възможности:

- Съобразен е с платформата e-learn.ue-varna.bg. Тоест за използването му е необходимо от платформата да се свалят решенията по определено задание, без да е необходимо допълнителни действия;
- Опростен интерфейс, който може да се използва и от преподаватели, незапознати с програмният език Python. Трябва само да се посочи папката, в която са свалените решения от платформата;
- Може да се настройва, до какъв процент съвпадения да се търси между отделните файлове;
- Предлага се възможност, тези със съвпадения да се отделят в папка „Еднакви“, за полесното им идентифициране и последващ анализ;
- Създава се файл „Rezult.csv“ със съвпадения между двойки файлове над зададеното съотношение. Файлът може да се отвори в Excel и да се прегледат резултатите.

Сърцевината на приложението е библиотеката **DIFFLIB** и по специално класът **SequenceMatcher**¹, който реализира алгоритъмът на Ratcliff-Obershelp². Той поддържа евристика, която автоматично третира определени елементи от последователността като нежелана. Евристиката брой колко пъти всеки отделен елемент се появява в последователността от текста. Основният метод в класът е `ratio()`, който връща float в диапазона [0, 1], измервайки сходството на последователностите. Като правило, `ratio()` - стойност над 0,6 означава, че последователностите имат близки съвпадения.

Примерният код на приложението е представен на фиг. 2.

¹ <https://docs.python.org/3/library/difflib.html>

² <https://xlinux.nist.gov/dads/HTML/ratcliffObershelp.html>

```

for root1, dirs1, files1 in os.walk(_path):
    current+=1
    progressbar['value']=round(current/max*100)
    root.update_idletasks()
    checked.append(root1)
    for filename1 in files1:
        with open(os.path.join(root1,filename1),'r',encoding='utf-8',errors='ignore') as f:
            source1=f.read()
        for root2, dirs2, files2 in os.walk(_path):
            if root1 == root2 or root2 in checked: continue
            for filename2 in files2:
                with open(os.path.join(root2,filename2),'r',encoding='utf-8',errors='ignore') as f:
                    source2=f.read()
                s=difflib.SequenceMatcher(None, source1,source2)
                ratio=round(s.ratio(),3)
                if ratio>0.5:
                    triggered.add(root1); triggered.add(root2)
                    file1.write(root1+';'+root2+';'+str(ratio)+'\n')
    if triggered:
        if messagebox.askquestion('Плагин',f'Намерени са {len(triggered)} файла с подозрително
съдържание. Да бъдат ли преместени в „еднакви“?') == 'yes':
            if not os.path.isdir(os.path.join(_path,'еднакви')):
                os.mkdir(os.path.join(_path,'еднакви'))
            for items in triggered:
                move(items, os.path.join(_path,'еднакви/'))
            messagebox.showinfo('Плагин',f'{len(triggered)} файла са преместени успешно.')
        else: messagebox.showinfo('Плагин','Не са открити файлове.')
    messagebox.showinfo('Плагин','Общо файлове:'+str(current))
file1.close()

```

Фигура 2. Примерен код на приложението

Източник: Собствена разработка

В предложения програмнен код е заложено процентно съвпадение от 0.5. При такова съотношение се предполага голяма част от двете решения да са еднакви.

3. ПРОБЛЕМИ ПРИ РАЗРАБОТКАТА И ТЕСТОВЕТЕ

При тестовите на приложението, възникнаха няколко проблема свързани със съдържанието на текстовите файлове. Те са както от техническо естество, така и от смисловото представяне на информацията. Може да ги обобщим в следните направления:

- Кодиране на текстовата информация. Въпреки, че всички решения се предават в текстови файлове, то съществува възможността те да са с различна кодировка. Това зависи от приложението, с което са обработени или от начина им на съхранение. Тъй като класът SequenceMatcher изисква като параметри да се подават два стринга с информацията от двата текстови файла за сравнение, то от съществено значение е коректното им прочитане. Възможно е файловете да са с различно кодиране на текста: ANSI, UTF -8, UTF-16 и т.н.
- Добавяне на съдържание. Добавянето на допълнителен текст към решението (например имена, фак.номер, текста на задачата и др.) променя процента съвпадение между

оригинал и копията му. Въпреки, че в заданията изрично се указва във файловете да се копира само програмният код, то не са изключения допълнителното „разкрасяване“ на текста. Включително и добавяне на коментари.

- Подредба на решенията. Ако заданието включва повече от една задача, то предадените решения може да са в различен порядък. Това променя съотношението на съвпаденията и съответно класът понижава стойността на еднаквост.
- Еднаквост на файлове. На текущия момент приложението отделя намерените еднакви файлове в една папка „Еднакви“, без да се групират по съвпадение. Тоест допълнително трябва да се провери, кои файлове с кои съвпадат. Предложен е вариантът с файл „Rezultat.csv“, в който са съхранени имената на двойките текстови файлове с процент съвпадение по-голям от зададения. При усъвършенстване на приложението е предвидена възможност за групиране на еднаквите решения.

За съжаление езикът Python не предлага големи възможности при изграждане на потребителски интерфейс. Затова приложението има само основните елементи за неговото функциониране. При разширяване на неговите функционалности, най-подходящ вариант за конфигуриране на различните опции е чрез външен файл с формат Json или Xml.

ЗАКЛЮЧЕНИЕ

Разбира се предложеният софтуер не е универсално решение. То е инструмент с цел да облекчи работата по оценяване на решенията на студентите. Не може да замени експертизата на преподавателите при анализа на програмният код. По скоро предлага улесняване на процеса на търсене на еднакви задания, като се предлага допълнителна информация за възможно съвпадение. Вероятно с развитието на AI допълнително ще улесни преподавателската работа.

REFERENCES / ИЗПОЛЗВАНА ЛИТЕРАТУРА

1. Ashwin Pajankar, (2019) Python 3 Image Processing: Learn Image Processing with Python 3, NumPy, Matplotlib, and Scikit-image, BPB Publications, 9388511727, 9789388511728.
2. Chris Meyer, Niklas Dellby, Jordan A. Hachtel, Tracy Lovejoy, Andreas Mittelberger, Ondrej Krivanek, (2019) Nion Swift: Open Source Image Processing Software for Instrument Control, Data Acquisition, Organization, Visualization, and Analysis Using Python, Microscopy and Microanalysis, Volume 25, Issue S2, 1 August 2019, Pages 122–123.
3. Jason M. Kinser (2018) Image Operators: Image Processing in Python, CRC Press, 0429835949, 9780429835940.
4. Jie Chen, Tao Jiang, Dongjin Yu, Haiyang Hu (2023) Pattern-based circular reference detection in Python. *Science of Computer Programming*, Volume 227, April 2023, 102932.
5. Samira Gholizadeh, Top Popular Python Libraries in Research (2022) Published in Journal of Robotics and Automation Research volume 3 issue 2.
6. Sandipan Dey. (2018) Hands-On Image Processing with Python: Expert techniques for advanced image analysis and effective interpretation of image data, Packt Publishing Ltd., 178934185X, 9781789341850.
7. Vaibhav Kant Singh, Santosh Kumar Soni , Dr. Nageshwar Dev Yadav, Pankaj Chandra (2023) A SYSTEM FOR CARTOONIFYING AN IMAGE USING PYTHON. *A Jurnal of Physical Sciences, Engeneering and Tegnology*, volume 14, pp.84-100.