

SOME POSSIBILITIES FOR USING SCRIPT TEXT LANGUAGE IN COMPUTER MODELING TRAINING

KRASIMIR V. HARIZANOV

ABSTRACT: This article discusses some guidelines related to the transition from block programming language to script text language, the main competence related to the teaching of Computer Modelling in 6th grade. Here are some sample tasks that may be useful in learning the Python scripting language.

KEYWORDS: education, interactive methods, computer modelling

DOI: <https://doi.org/10.46687/PSVY3940>

НЯКОИ ВЪЗМОЖНОСТИ ЗА ИЗПОЛЗВАНЕ НА СКРИПТОВ ТЕКСТОВ ЕЗИК В ОБУЧЕНИЕТО ПО КОМПЮТЪРНО МОДЕЛИРАНЕ*

КРАСИМИР В. ХАРИЗАНОВ

АБСТРАКТ: Настоящата статия разглежда някои насоки, свързани с преминаването от език с блоково програмиране към скриптов текстов език, основна компетентност свързана с обучението по Компютърно моделиране в 6 клас. Предложени са няколко примерни задачи, които могат да бъдат полезни при изучаването на скриптовия език Python.

1 Въведение

Обучението е един динамичен и сложен процес и като такъв, той непрекъснато се развива и променя. Учебните програми, като основна част от учебният процес, периодично се актуализират по отношение на тяхното тематично съдържание и очаквани резултати. Промените имат за цел да се развиват компетентностите на учениците, като се надграждат през всяка следваща учебна година и да достигат стандартите на водещите европейски държави. Всички тези промени водят след себе си нови предизвикателства пред учителя, както и нови учебници съобразени с новите изисквания на учебните програми.

2 Изложение

През учебната 2021/2022 година влязоха в сила новите, утвърдени програми по дисциплината „Информационни технологии“ за V-VII клас, под наименованието „Компютърно моделиране и информационни технологии“. Промяната в името на дисциплината е свързано с добавянето на още една глобална тема „Компютърно моделиране“, към вече съществуващото учебно съдържание на учебните програми. За новата тема са предвидени допълнителни часове като общият обем в препоръчаното годишно разпределение нараства от 36 до 52 учебни часа.

В V клас темата „Компютърно моделиране“ надгражда знанията на учениците от III и IV клас, като основният акцент попада върху изграждане на умения, свързани със

* Настоящата статия е частично финансирана по проект № РД - 08-146/02.03.2022

„Създаване и използване на собствени блокове или подпрограми“[[2]]. Тук целта на програмата е, да се изградят основни компетентности на учениците свързани с прилагането на алгоритми и усъвършенстване на задачи, които да се декомпозират чрез собствени блокове или подпрограми. На практика, основата, която се поставя в V клас има пряко отношение с целите, които са поставени в следващите програми за VI и VII клас. Изграждането на логическо мислене, което може да прилага различни видове алгоритми и подпрограми в блокова среда, по-лесно и интуитивно се адаптира към програмиране със скриптов текстове език.

В учебната програма за VI клас, темата „Дигитално творчество“ е свързана с изучаването на скриптов език като „Python, JavaScript и др.“[[3]]. В тези часове, учителят трябва да представи различни видове задачи, свързани с разглеждането на линеен, разклонен и цикличен алгоритъм чрез блоково програмиране и скриптов текстове език. За покриването на очакваните резултати в тази тема е необходимо да се използва среда, чрез която може да се демонстрират и сравняват кодовете от блоковото програмиране с тези на избрания скриптов текстове език.

В предложените учебни програми, няма строго определено изискване кой скриптов текстове език да се изучава в темата „Компютърно моделиране“. Изборът на подходящ език, до голяма степен е оставен на авторските колективи, които са ангажирани със създаването на учебниците и учебните материали към тях. Така например авторският колектив[[1]] са се спрели на езика Python. В уроците свързани с преминаването от език с блоково програмиране към скриптов текстове език, авторите на учебникът отделят внимание на няколко среди, чрез които се съпоставят решенията на задачи от блокова среда Scratch и скриптовия език Python. Дават се примери и за среди, в които може директно да се въвежда и транслира скриптов код като се „показват възникналите грешки или резултата от изпълнението на кода“[[1]].

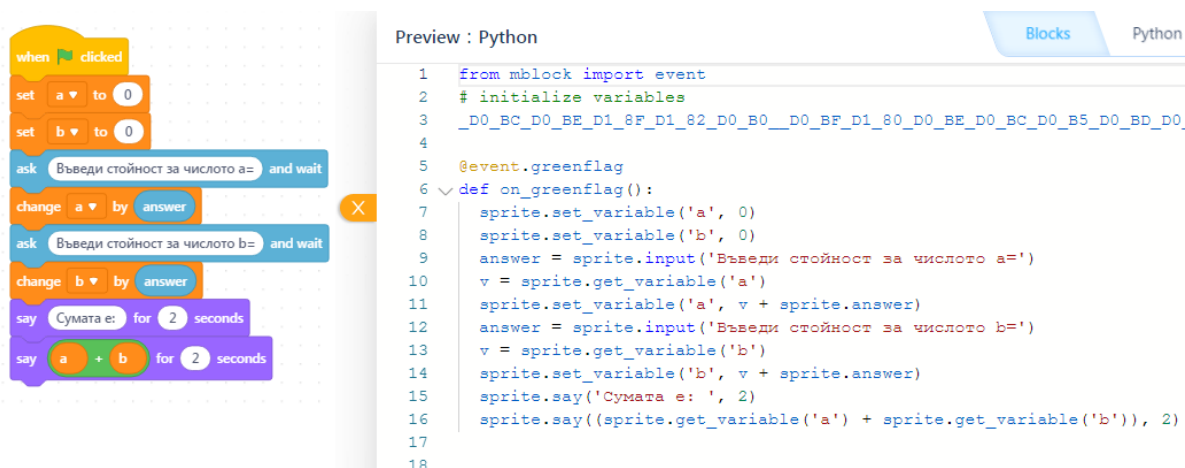
mBlock

Онлайн средата за програмиране mBlock[[5]], е една много добра възможност, всеки потребител, който се е занимавал с блоково програмиране, бързо и лесно да се запознае с особеностите на езика Python. Както е описано в [1], тази среда има много прилики с познания за много ученици и учители интерфейс на Scratch. Средата дава възможност да се отварят задачи и проекти реализирани на Scratch, като за удобство при работа, потребителят може да използва някой от най-популярните езикови визуализации (английски, руски, френски и др.), с изключение на българския.



Фиг. 1 Работен екран на mBlock. 1 – Сцена с герои/спрайтове; 2 – категории с блокове; 3 – област за кодиране; 4 – инструмент за скриване и показване кода на Python

След стартиране на приложението, потребителят може да види познатите области от блоковата среда Scratch, да започне нова задача или да отворите вече създадена такава. Средата разполага с голямо разнообразие от спрайтове и декори, чрез които могат да се създават забавни и интересни задачи, от познатите до сега възможни сюжети.



Фиг. 2 Паралелен изглед на блоков код и код на Python

Изгледът, който се вижда като интерпретация на езика Python, позволява неговото изучаване, на ниво, на което учениците да съпоставят функцията на конкретен блок с неговият скриптов код. Така категория по категория, блок по блок, изучаването на езика Python, става по един по-лесен и мотивиращ за учениците начин.

Основният проблем при изучаването на този скриптов език, чрез платформата mBlock е, че кодовата интерпретация е свързана със спрайтовете използвани в задачите. Генерираният код е пряко отнесен към конкретен спрайт, който в практическото използване на езика има някои разминавания, като например:

Пример 1

mBlock

```
answer = sprite.input('Въведи стойност за числото a=')
v = sprite.get_variable('a')
sprite.set_variable('a', v + sprite.answer)
```

Python

```
a=input("Въведи стойност за числото a=")
```

Пример 2

mBlock

```
sprite.say('Сумата е: ')
```

Python

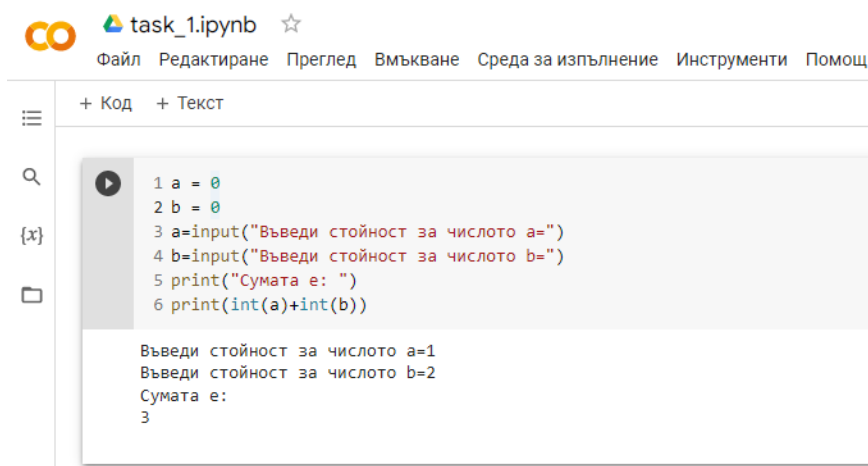
```
print("Сумата е: ")
```

Макар да се видни някои различия между интерпретираният код и реалния скриптов код, средата mBlock дава една много добра основа за това какво представляват скриптовите езици и кои са особеностите при въвеждането на такъв код. При изучаването на скриптовият език тук, много бързо се проследяват отделните редове заради тяхното номериране, а чрез автоматичното цветно кодиране, визуално лесно се разграничава кода

свързан потребителя и тази, която служи за логическата функционалност. Тази среда може да бъде един отличен помощник на всеки преподавател в присъствена или онлайн среда.

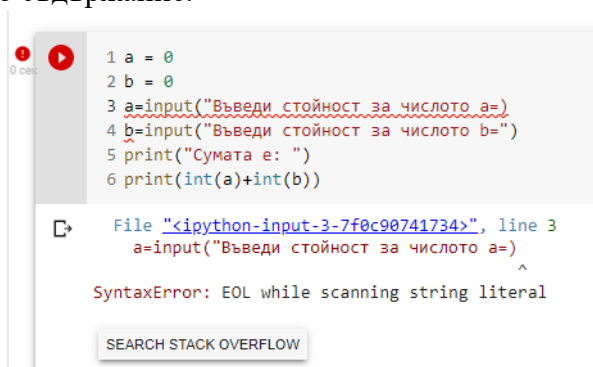
Colab

В следващият етап на обучение си, учениците в VII клас ще се запознаят с приложението на различните типове данни и цикличните конструкции в скриптовите езици[[4]]. На този етап няма ясно открити платформи и приложения, чрез които трябва да се обезпечи глобалната тема „Компютърно моделиране“, като на този етап се търсят решения, които да бъдат достатъчно ефективни и подходящи за възрастовата група на обучаемите.



Фиг. 3 Примерна задача въведена в облачната платформа Colab

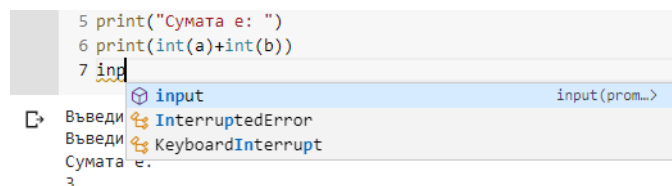
Въвеждането на нова задача, започва със създаването на *Нов бележник* от менюто Файл. Всеки бележник се съхранява с разширение *ipynb* като достъпът до него може да бъде споделен или да се изтегли локално на потребителския компютър. Стартирането и проверката на въведеният код се изпълнява от инструмента *Изпълнение (Play)* като получения резултат се визуализира в бялата област, разположена под кода. Ако е допусната синтактична грешка, редакторът показва от какъв вид е грешката и на кой ред се намира тя. Като всеки облачен документ, така и този се съхранява автоматично, при всяка промяна в неговото съдържание.



Фиг. 4 Пример за допусната синтактична грешка

Създаването на скриптов код в платформата се подпомага от автоматичен помощник, който при въвеждане на първите символи от определена функция се генерира

списък с предложения, от които потребителя може да избере желаната от неговата функция.



```
5 print("Сумата е: ")
6 print(int(a)+int(b))
7 ind
input
input(prom...>
Въведи InterruptedError
Въведи KeyboardInterrupt
Сумата е.
3
```

Фиг. 5 Автоматичен помощник към редактора Colab

Ролята на този помощник е обучаемия по-бързо да въвежда необходимият код, без да допуска синтактични грешни. В допълнение и тук интерфейсът предлага удобството да се разграничават отделните редове, чрез номерация, което улеснява процеса на преподаване, а също цветното кодиране на отделните логически области.

3 Заключение

Използването на онлайн среди, свързани с обезпечаването на учебният процес има своите положителни приноси, като достъпност, актуалност и работа в споделен режим. До голяма степен разгледаните платформи могат успешно да се използват в темата „Компютърно моделиране“, която е заложена в новите учебни програми в прогимназиален етап. Те са подходящи именно за тази възрастова група, както по своя дружелюбен интерфейс, така и заради липсата на допълнителна инсталация в домашни условия.

Разбира се остава въпросът какъв ще бъде изборът на учителя за скриптов текстов език за тези глобална тема и какви ще бъдат изискванията към него и условията за работа в конкретния компютърен кабинет.

ЛИТЕРАТУРА:

- [1] Ангелов А., Ковачева, Е., Иванова, В., Харизанов, Кр., Сребрева, Т., учебник по „Компютърно моделиране и информационни технологии“ 6 клас, Издателство Клет България, ISBN 978-954-18-1657-8, 2022
- [2] URL: https://mon.bg/upload/24186/UP_PC-modelirane-5kl.pdf
- [3] URL: https://www.mon.bg/upload/24187/UP_PC-modelirane-6kl.pdf
- [4] URL: https://www.mon.bg/upload/24188/UP_PC-modelirane-7kl.pdf
- [5] <https://ide.mblock.cc/>
- [6] <https://colab.research.google.com/>

