

ВЪЗМОЖНОСТИ ЗА РАЗВИТИЕ НА АЛГОРИТМИЧНИ УМЕНИЯ НА УЧЕНИЦИТЕ В ОБУЧЕНИЕТО ПО ПРЕДМЕТА „КОМПЮТЪРНО МОДЕЛИРАНЕ И ИНФОРМАЦИОННИ ТЕХНОЛОГИИ“ В ПРОГИМНАЗИЯТА

Стефка Анева^{1,*}, Елена Тодорова²

^{1, 2} Факултет по математика и информатика, ПУ „Паисий Хилендарски“,
гр. Пловдив, бул. „България“ № 236

Имейли: stfaneva@uni-plovdiv.bg, etodorova@uni-plovdiv.bg

* Автор за кореспонденция: stfaneva@uni-plovdiv.bg

Резюме. В настоящата статия са дискутирани някои методически аспекти на обучението по програмиране в прогимназиален етап, свързани с развитието на алгоритмични умения на учениците чрез използване на език за блоково програмиране и скриптов текстов език. В разработката са представени примерни задачи, свързани с изучаване на елементи на програмиране в обучението по новия учебен предмет „Компютърно моделиране и информационни технологии“ в прогимназията. Предложени са техни конкретни реализации чрез езиците Scratch и Python.

Ключови думи: компютърно моделиране, език за блоково програмиране, скриптов текстов език, алгоритмични умения, методика на обучение.

Въведение

Процесът на формиране на алгоритмично мислене и дигитална креативност у учениците трябва да започва още в началното училище и да се надгражда и развива във всеки следващ етап на обучение. Към момента акцентът в обучението по компютърно моделиране в началните класове (3.-4. клас) е свързан с овладяване на основите на програмирането и формиране на практически и приложни умения за конструиране на последователни, циклични и разклонени алгоритми чрез използване на конкретна визуална среда за блоково програмиране [2, 5, 6]. Scratch е много подходяща среда за въведение в света на програмирането при

подрастващите поради неговата относителна лекота и възможност за лесна и бърза реализация на интересни програми и интерактивни проекти [3, 4, 5, 6]. Формираните алгоритмични знания и креативни умения у учениците в началното училище, усвоени чрез работа с конкретна среда за блоково програмиране, могат да бъдат успешно приложени и развити в прогимназиален и гимназиален етап на обучение при работа с други основни програмни езици като Python, JavaScript, Java и C#.

За новото учебно съдържание по учебния предмет „Компютърно моделиране и информационни технологии“ в прогимназията

От учебната 2021/2022 г. се предвижда учебният предмет „Информационни технологии“ в прогимназиален етап да се изучава с актуализирано учебно съдържание, променен хорариум и с ново наименование „Компютърно моделиране и информационни технологии“. Детайлният преглед на новите учебни програми за 5., 6. и 7. клас, одобрени през 2020 г., показва, че няма съществени промени спрямо всички включени за изучаване до момента учебни модули от предмета „Информационни технологии“, тъй като те са ключови за развитието на дигиталните компетентности на учениците. Новият момент касае добавянето на темата „Компютърно моделиране“, свързана с надграждане на базовите умения за програмиране, формирани в началния етап на обучение, и изграждане на алгоритмични знания и умения у учениците при изучаване и използване на различни видове езици за програмиране [1, 7].

Обучението в 5. клас по темата „Компютърно моделиране“ е насочено към:

- задълбочаване на знанията и уменията на учениците за работа с блоков език за програмиране, придобити в началния етап на обучение, и надграждане със създаване и използване на собствени блокове и подпрограми;
- формиране на знания и умения за прилагане на алгоритми за реализация на следните дейности: размяна на стойности, броене на елементи; намиране на минимален/максимален от три елемента; подреждане на три елемента по големина.

При изучаване на темата „Компютърно моделиране“ в 6. и 7. клас се предвижда надграждането на формираните знания и умения за програмиране чрез преминаване от език с блоково програмиране към скриптов текстов език. Въвеждането на скриптовия език се предвижда да се реализира чрез съпоставка и сравнение между конкретни примери за

линеен, разклонен и цикличен алгоритъм, реализирани чрез средствата на език с блоково програмиране и скриптов текстов език.

Примерни идеи за реализация на някои базови алгоритми в обучението по „Компютърно моделиране и информационни технологии“ в прогимназията

В настоящата разработка се представят примерни идеи за реализация на следните дейности – **размяна на стойности на две променливи и намиране на минимален/максимален от няколко елемента**, като съответните алгоритми ще се реализират и надграждат в процеса на обучение в 5., 6. и 7. клас с помощта на различни езици за програмиране. За целта се използва следният подход:

1. Реализация на базови алгоритми чрез средата за блоково програмиране Scratch – приложим за учениците от 5. клас и способства за задълбочаване на знанията и уменията им за използване на блоков език за програмиране, формирани в началното училище.
2. Реализация чрез езика Python на вече разгледани базови алгоритми в 5. клас, като в процеса на работа се извършва съпоставка с примерни варианти на техни решения чрез език за блоково програмиране Scratch – приложим за учениците от 6. и 7. клас.

При представяне на вариантите за решение на задачите чрез Scratch сме приложили визуализации само на конкретните реализации на алгоритмите чрез използване на собствени блокове. Предложените примерни модели на задачите в средата на Scratch са опростени като визия, тъй като целта е да се наблегне на самите алгоритми и вариантите за тяхната реализация. Тези модели са реализирани по такъв начин, че подсказват и самата функционалност на приложенията.

Задача 1. Размяна на стойностите на две променливи.

Реализация на алгоритъма за размяна на стойности чрез Scratch



Фигура 1. Примерен модел на задачата и алгоритъм за размяна на стойности на две променливи

Реализация на задачата с помощта на скриптов текстов език (Python)

Примерни варианти за реализация на алгоритъма за размяна на стойностите на две променливи с/без използване на помощна променлива:

```
1 print("Въведи две числа:")
2 a=int(input())
3 b=int(input())
4 swap=0
5 print("Преди размяната: a=",a," b=",b)
6 swap=a
7 a=b
8 b=swap
9 print("След размяната: a=",a," b=",b)
```

```
1 print("Въведи две числа:")
2 a=int(input())
3 b=int(input())
4 print("Преди размяната: a=",a," b=",b)
5 a,b=b,a
6 print("След размяната: a=",a," b=",b)
```

Задача 2. Намиране на минимален и максимален елемент от няколко въведени стойности.

Предложената задача е комплексна и е добре да бъде реализирана като отделни подзадачи с различна степен на сложност, като се използва следния примерен сценарий:

Първо ниво: Намиране на по-малкото/по-голямото от две числа.

Второ ниво: Намиране на минимален/максимален елемент от три въведени числа.

Трето ниво: Намиране на минимален/максимален елемент при повече от три въведени числови стойности.

Всеки един от компонентите на тази комплексна задача може да бъде дискутиран и надграждан на различен етап в процеса на обучение на учениците в прогимназията и да бъде реализиран както с блоков език за програмиране, така и със скриптов текстов език. По този начин може да бъде направена една пълноценна съпоставка и анализ на възможностите и спецификите при реализация на даден алгоритъм с помощта на различни езици за програмиране. Тези дейности биха подпомогнали процеса на формиране и развитие на базови знания, и умения на учениците в областта на програмирането.

Реализация на задачата с помощта на езиците Scratch и Python

Посочените по-долу примерни варианти за намиране на минимален елемент за всяко ниво от предложения сценарий са реализирани в Scratch с използване на собствени блокове. Аналогично може да се постъпи при определянето на максимален елемент при всяка една от разгледаните ситуации.

При запознаването на учениците със спецификите при използване на скриптов текстов език, разгледаните вече алгоритми за намиране на минимален и максимален елемент и техните примерни реализации чрез

блоков език за програмиране могат да се използват като основа при решаването на задачата чрез Python.

Реализация на първо ниво: Намиране на минимален/максимален елемент на две числа

При запознаване на учениците от 5. клас със спецификите за реализация на алгоритъма за намиране на минимален елемент от две числа (**a** и **b**) в среда на **Scratch** можем да представим следните примерни сценарии:

- в помощна променлива **min** първоначално запомняме стойността на **a**, след което се извършва проверка дали **b** е по-малко от **min** и ако това е така – стойността на **b** се запомня в променливата **min**.
- първоначално на променливата **min** се присвоява стойност **0**, след което се извършва проверка за определяне на по-малкото от числата **a** и **b**, и извличане на стойността му в **min**.

Накрая най-малкото число се получава в променливата **min**.

Примерна реализация на алгоритъма за намиране на минимален елемент на две числа чрез Scratch



Фигура 2. Примерен модел на задачата и алгоритъм за намиране на минимален елемент на две числа

Примерна реализация на алгоритъма за намиране на минимален елемент на две числа чрез Python

```
1 print("Въведи двете числа:")
2 a=int(input())
3 b=int(input())
4 min=a
5 if b<min:
6     min=b
7 print("Минимален елемент:",min)
```

```
1 print("Въведи двете числа:")
2 a=int(input())
3 b=int(input())
4 min=0
5 if a<b:
6     min=a
7 else:
8     min=b
9 print("Минимален елемент:",min)
```

На учениците от 6. клас може да се представи и друг вариант за намиране на минимален и максимален елемент чрез директно използване на готовите функции **min()** и **max()**, например:

```
print("Минимален елемент: ", min(a,b))
```

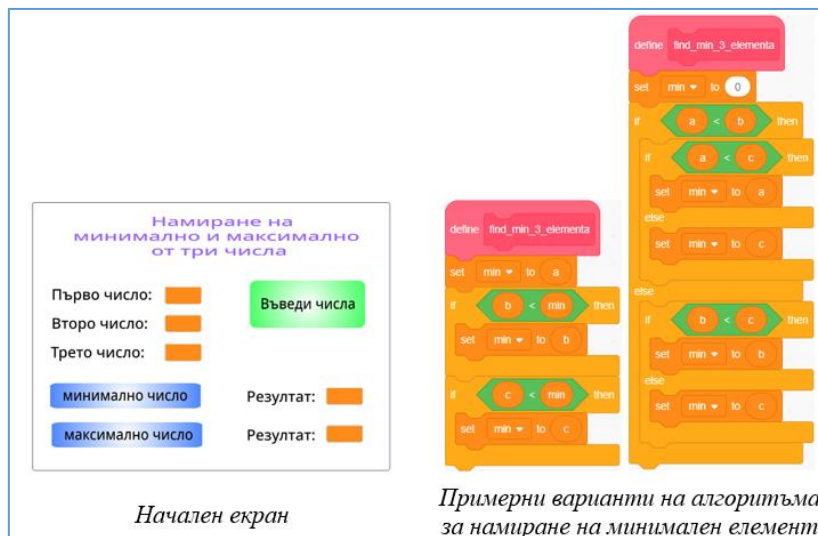
Реализация на второ ниво: Намиране на минимален/максимален елемент на три числа

При запознаване на учениците от 5. клас със спецификите за реализация на алгоритъма за намиране на минимален елемент от три числа (**a**, **b** и **c**) в среда на **Scratch**, можем да представим следните примерни сценарии:

- в помощна променлива **min** първоначално започваме с стойността на **a**, след което с помощта на отделни блокове за разклонение всяко от следващите числа **b** и **c** се сравнява с **min**. Ако се намери число по-малко от **min**, то се започва в **min**.
- първоначално на променливата **min** се присвоява стойност **0**, след което с помощта на вложени блокове за разклонение се извършва последователна проверка за определяне на най-малкото от числата **a**, **b** и **c**, и извличане на стойността му в помощната променлива **min**.

Накрая променливата **min** съдържа най-малкото число.

Примерна реализация на алгоритъма за намиране на минимален елемент на три числа чрез **Scratch**



Фигура 3. Примерен модел на задачата и алгоритъм за намиране на минимален елемент на три числа

Самото задаване на стойностите на трите числа може да бъде реализирано най-напред чрез ръчно въвеждане, а след това да бъде показан вариант и за автоматичното им попълване чрез случайно генериране на тези стойности в определен интервал (например, [1,100]).

Примерна реализация на алгоритъма за намиране на минимален елемент на три числа чрез **Python**

```

1 print("Въведи три числа:")
2 a=int(input())
3 b=int(input())
4 c=int(input())
5 min=a
6 if b<min:
7     min=b
8 if c<min:
9     min=c
10 print("Минимален елемент:",min)

```

```

1 print("Въведи три числа:")
2 a=int(input())
3 b=int(input())
4 c=int(input())
5 min=0
6 if a<b:
7     if a<c:
8         min=a
9     else:
10        min=c
11 elif b<c:
12     min=b
13 else:
14     min=c
15 print("Минимален елемент:",min)

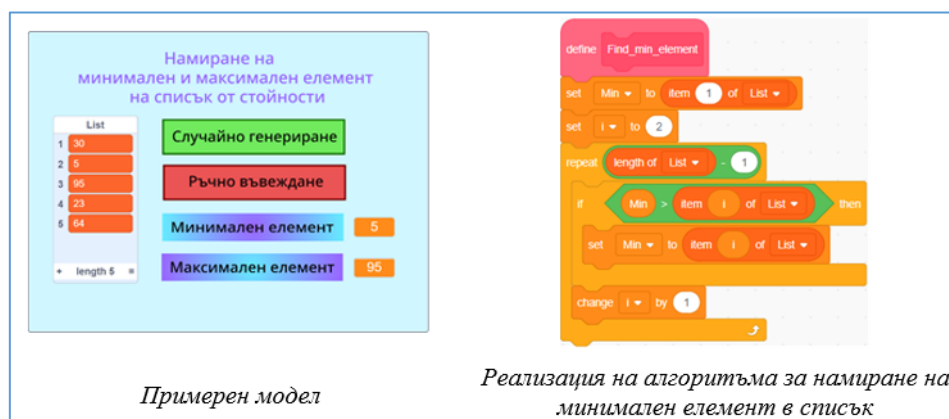
```

Реализация на трето ниво: Намиране на минимален/максимален елемент на повече от три числови стойности

При реализация на алгоритъма за намиране на минимален/максимален елемент при повече от три въведени елемента е най-добре да се използва списък при съхраняване на въведените стойности. В случая може да се реализират два възможни варианта за въвеждане на стойностите в списъка: чрез случайно генерирани стойности или чрез ръчно въвеждане на елементите на списъка.

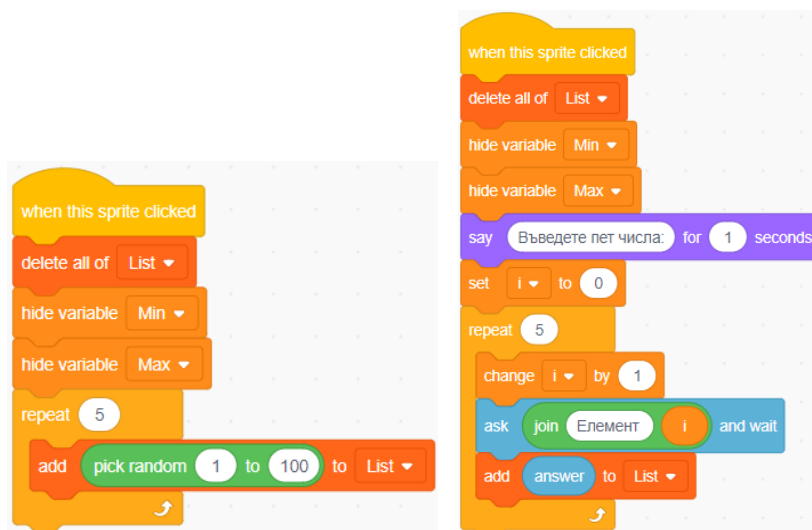
Примерна реализация на алгоритъма за намиране на минимален елемент на повече от три числа чрез Scratch

Най-напред може да бъде разгледан конкретен вариант на реализация при използване на списък с пет числови стойности. За самостоятелна работа може да се възложи на учениците да предложат вариант на решение, при използване на списък с произволен брой стойности.



Фигура 4. Примерен модел на задачата и алгоритъм за намиране на минимален елемент на повече от три числа

Реализиране на дейностите по попълване на списъка с пет числови стойности – чрез случайно генериране на петте елемента със стойности от 1 до 100 или чрез ръчно въвеждане на произволна числова стойност за всеки един от елементите на списъка:



Примерна реализация на алгоритъма за намиране на минимален елемент на повече от три числа чрез Python

За надграждане и развитие на знанията и уменията на учениците от 7. клас за използване на циклични конструкции може да бъде представен вариант за реализация на този алгоритъм чрез **Python**. Първоначално може да бъде разгледана ситуация, при която се използва списък с предварително въведени числови стойности, за които в режим на изпълнение да се определи само кой е минималният елемент от така въведения набор от числа.

```

1 List_Numbers=[11,33,5,7,22]
2 def min_element(numbers):
3     min=numbers[0]
4     for i in numbers:
5         if i<min:
6             min=i
7     return min
8 print("min=", min_element(List_Numbers))

```

След това задачата може да бъде надградена, като се реализира възможност за произволно въвеждане на краен брой числови стойности в списъка в режим на изпълнение на програмата – както чрез ръчно въвеждане, така и чрез случайно генериране на числови стойности в интервал от 1 до 100.

```

1 List_Numbers = []
2 print("Въведете пет числа:")
3 for x in range(5):
4     List_Numbers.append(int(input()))
5 print("Введените стойности са:", List_Numbers)
6 def min_element(numbers):
7     min=numbers[0]
8     for i in numbers:
9         if i<min:
10            min=i
11     return min
12 print("min=", min_element(List_Numbers))

```

```

1 import random
2 List_Numbers = []
3 print("Генериране на пет числа в интервала от 1 до 100: ")
4 for x in range(5):
5     List_Numbers.append(random.randint(1,100))
6 print("Введените стойности са:", List_Numbers)
7 def min_element(numbers):
8     min=numbers[0]
9     for i in numbers:
10        if i<min:
11            min=i
12     return min
13 print("min=", min_element(List_Numbers))

```

В процеса на работа по задачата може да бъде демонстриран вариант с приложение на оператор за цикъл `while` при реализиране на дейността по въвеждане на елементите на списъка.

```
x=1
while x<=5:
    List_Numbers.append(int(input()))
    x+=1
```

```
x=1
while x<=5:
    List_Numbers.append(random.randint(1,100))
    x+=1
```

За самостоятелна работа може да бъде зададена задача на учениците да усъвършенстват реализираната до момента програма за търсене на минимален и максимален елемент за набор от числови стойности, като реализират възможността техния брой (т.е. дължината на списъка) да се определя (уточнява) от потребителя след стартиране на програмата.

Заклучение

Предложеният в настоящата разработка примерен подход и реализираните учебни задачи могат да бъдат успешно използвани в обучението по предстоящия нов учебен предмет „Компютърно моделиране и информационни технологии“ от учебната 2021/2022 г. и да подпомогнат процеса на развитие на знанията и уменията на учениците в областта на програмирането.

Благодарности

Това изследване е частично подкрепено от Национална научна програма „Информационни и комуникационни технологии за единен цифров пазар в науката, образованието и сигурността (ИКТвНОС)“, финансирана от МОН.

Литература

- [1] Е. Стефанова, К. Стефанов, *От костенурката до питона: обучението на учители по информатика и информационни технологии и промените в учебните програми*, сборник доклади на Петдесетата юбилейна пролетна конференция на Съюза на математиците в България, Математика и математическо образование, 2021, 123–132, ISSN: 1313-3330.
- [2] К. Гъров, С. Анева, Е. Тодорова, *Някои методически аспекти на обучението по програмиране в началното училище*, сборник доклади на научна конференция „Иновационни ИКТ в бизнеса и обучението: тенденции, приложения и разработване“, Пампорово, 2016, 109–116, ISBN: 978-954-8852-72-2.

- [3] К. Гъров, Н. Тодорова, *Образователни игри и приложението им в обучението по „Компютърно моделиране“*, сборник доклади на научна конференция „Иновационни ИКТ за дигитално научноизследователско пространство по математика, информатика и педагогика на обучението“, Пампорово, 2019, 77–92, ISBN: 978-619-202-572-4.
- [4] К. Гъров, Г. Колева, Н. Тодорова, *Компютърното моделиране в помощ на обучението на деца със специални образователни потребности*, сборник доклади на юбилейна международна научна конференция „Синергетика и рефлексия в обучението по математика“, Пампорово, 2020, 235–244, ISBN: 978-619-202-595-3.
- [5] Р. Папанчева, Т. Глушкова, К. Димитрова, *Компютърно моделиране за 3. клас общообразователна подготовка*, изд. Изкуства, София, 2018, ISBN: 978-619-7243-51-2.
- [6] Р. Папанчева, Т. Глушкова, *Компютърно моделиране за 4. клас общообразователна подготовка*, изд. Изкуства, София, 2019, ISBN: 978-619-7243-64-2.
- [7] МОН, Учебни програми по компютърно моделиране и информационни технологии (2020): <https://mon.bg/bg/100884> (посетен на 20.08.2021 г.)

PROSPECTS FOR DEVELOPING STUDENTS' ALGORITHM SKILLS IN TEACHING THE SUBJECT “COMPUTER MODELING AND INFORMATION TECHNOLOGIES” IN MIDDLE SCHOOL

Stefka Aneva^{1,*}, Elena Todorova²

^{1,2} Faculty of Mathematics and Informatics,
University of Plovdiv “Paisii Hilendarski”, 236, Bulgaria Blvd., Plovdiv, Bulgaria
Emails: stfaneva@uni-plovdiv.bg, etodorova@uni-plovdiv.bg

* Corresponding author: stfaneva@uni-plovdiv.bg

Abstract. In this current paper we discuss some methodical aspects of teaching programming in the middle school which are related to the development of algorithm skills among the students by using a block-based programming language and a scripting language. In the research we present model problems related to studying elements of programming in teaching the new school subject “Computer Modeling and Information Technologies” in the middle school. Their specific realization is presented via the languages Scratch and Python.