

**Възможности за развитие на алгоритмични умения на
учениците в обучението по предмета „Компютърно
моделиране и информационни технологии“ в
прогимназията**

Стефка Анева, Елена Тодорова



Anniversary International Scientific Conference "Computer Technologies and Applications"
15-17 September 2021, Pamporovo, Bulgaria



Въведение

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.

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.

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.

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.

Изучаване на темата „Компютърно моделиране“ в прогимназията

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 следния подход:

- 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.
- 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, които могат да се ползват като основа при решаването на съответната задача чрез Python – приложим за учениците от 6. и 7. клас.

Задача 1. Размяна на стойностите на две променливи.

Scratch



Примерен модел на задачата и алгоритъм за размяна на стойности на две променливи

Python

```
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)
```

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

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.

Примерни варианти на алгоритмите за намиране на минимален елемент на две и на три числа

Scratch

Намиране на минимално и максимално от две числа

Първо число: Въведи числа
Второ число: Резултат:
минимално число Резултат:
максимално число Резултат:

Примерни варианти на алгоритъма за намиране на минимален елемент

Начален екран

Намиране на минимално и максимално от три числа

Първо число: Въведи числа
Второ число: Резултат:
Трето число: Резултат:
минимално число Резултат:
максимално число Резултат:

Примерни варианти на алгоритъма за намиране на минимален елемент

Начален екран

Примерни модели на задачите и алгоритми за намиране на минимален елемент на две и на три числа

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 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)
```

Примерна реализация на алгоритмите за намиране на минимален елемент на две и на три числа чрез Python

Примерни варианти на алгоритмите за намиране на минимален елемент на повече от три числови стойности

Scratch

Намиране на минимален и максимален елемент на списък от стойности

Случайно генериране
Ръчно въвеждане
Минимален елемент: 5
Максимален елемент: 95

```
define Find_min_element
set Min to item 1 of List
set i to 2
repeat (length of List - 1)
if (Min > item i of List) then
set Min to item i of List
change i by 1
```

Примерен модел

Реализация на алгоритъма за намиране на минимален елемент в списък

```
when this sprite clicked
delete all of List
hide variable Min
hide variable Max
repeat 5
add (pick random 1 to 100) to List
```

```
when this sprite clicked
delete all of List
hide variable Min
hide variable Max
say "Въведете пет числа:" for 1 seconds
set i to 0
repeat 5
change i by 1
ask (join "Елемент " i) and wait
add (answer) to List
```

Python

```
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))
```

```
1 List_Numbers = []
2 print("Въведете пет числа:")
3 for x in range(5):
4     List_Numbers.append(int(input()))
5 print("Въведените 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("Въведените 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))
```

Реализиране на възможности по въвеждане на краен брой числови стойности (в случая 5) в списъка – чрез **случайно генериране** на петте елемента със стойности в интервала от 1 до 100 или чрез **ръчно въвеждане** на произволна числова стойност за всеки един от елементите на списъка.



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.

Благодаря за вниманието!

Автори:

Стефка Анева, stfaneva@uni-plovdiv.bg

Елена Тодорова, etodorova@uni-plovdiv.bg

Факултет по математика и информатика, ПУ „Паисий Хилендарски“

Настоящата 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.