

# Основни алгоритми

Използване на променливи



SCRATCH



SoftUni Team  
Technical Trainers



Software University

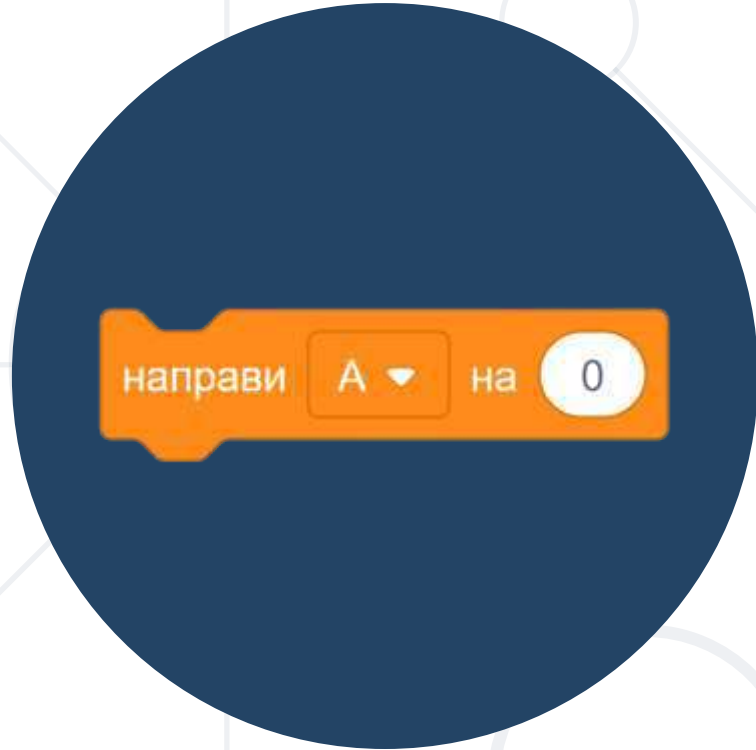
<http://softuni.bg>

## 1. Променливи

## 2. Алгоритми за:

- Размяна на стойности
- Броене на елементи
- Намиране на **най-големия от три** елемента
- Сортиране на елементи





# Променливи

Съхраняване на стойности

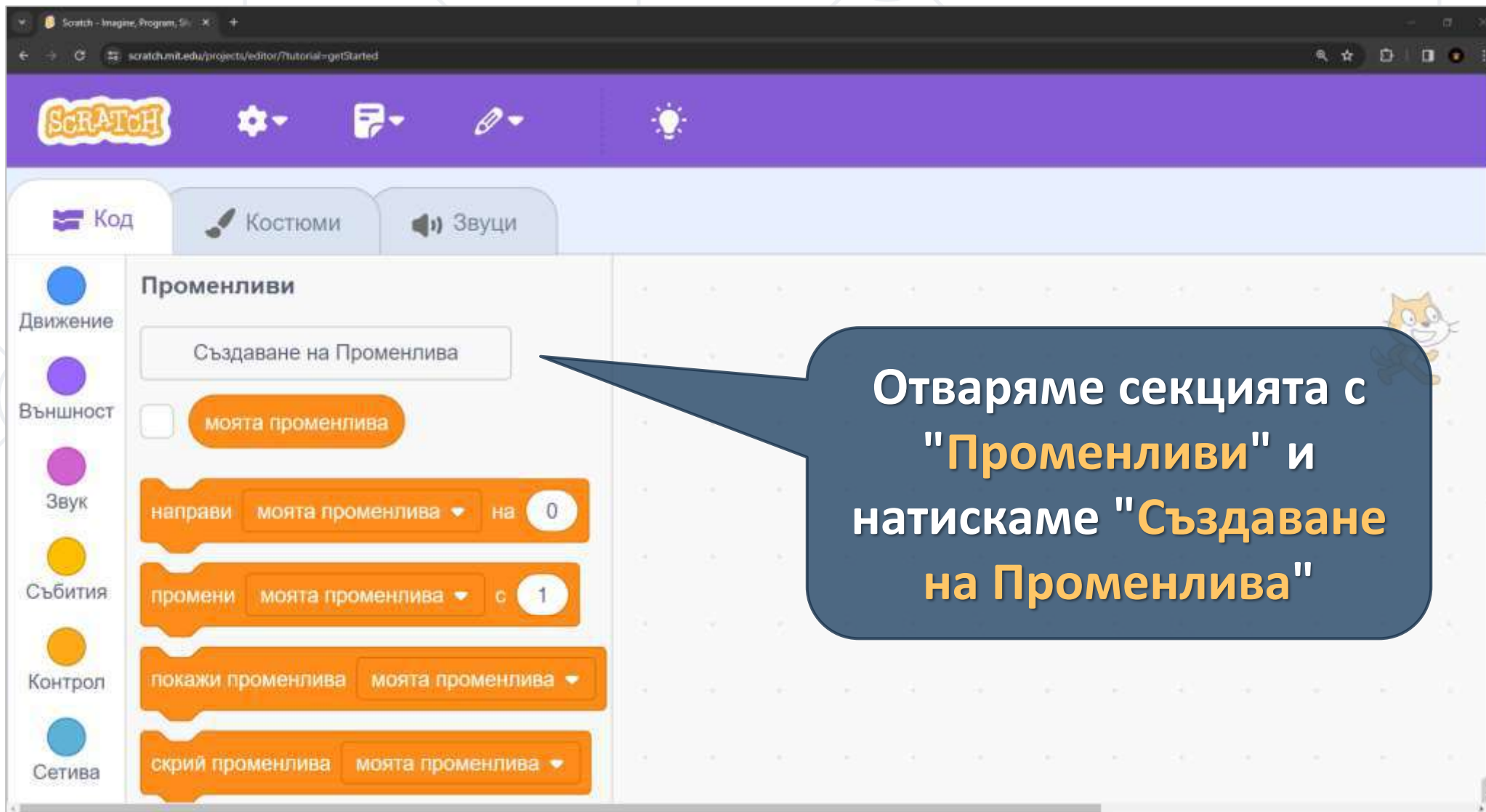
- **Променлива** - стойност, записана в паметта на Scratch
  - Може да променя **стойността си**
  - Променливите могат да съдържат **само една стойност в даден момент**
  - Стойностите на **променливите** могат да бъдат **числа или текст**

Име на променливата



Стойност на  
променливата

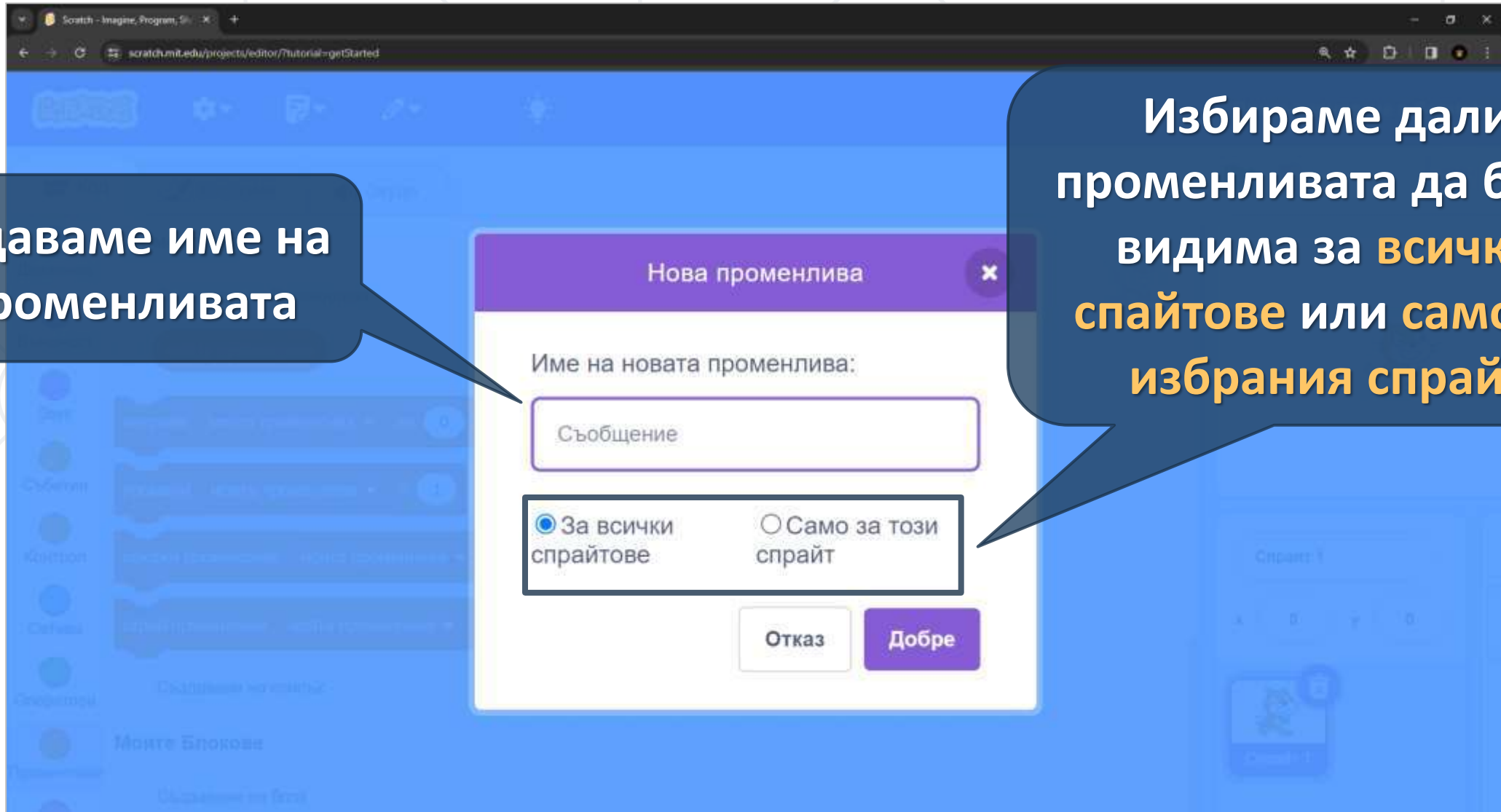
# Създаване на променлива



Отваряме секцията с "Променливи" и натискаме "Създаване на Променлива"

# Създаване на променлива

Задаваме име на променливата

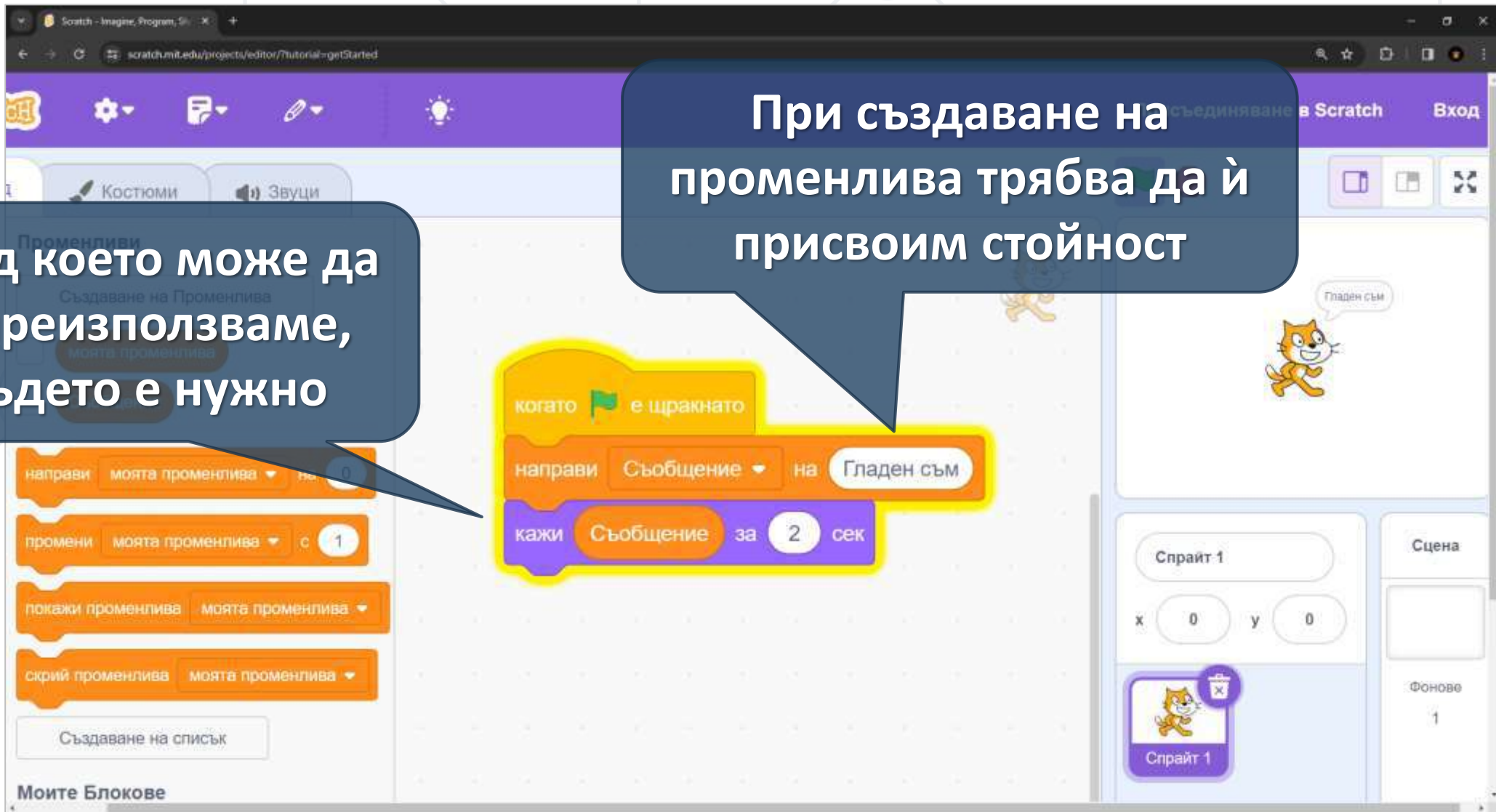


Избираме дали променливата да бъде видима за **всички спайтове** или **само за избрания спайт**

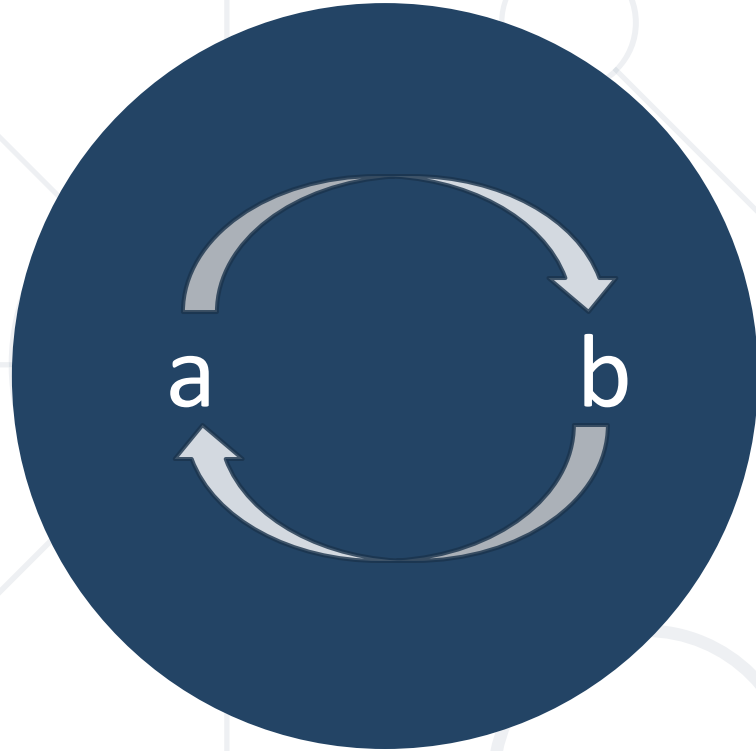
# Създаване на променлива

След което може да я използваме, където е нужно

При създаване на променлива трябва да ѝ присвоим стойност



The screenshot shows the Scratch IDE interface. On the left, the 'Variables' palette is open, displaying a 'Create Variable' dialog box for 'Моята променлива' (My Variable) with a value of 0. Below it are options to 'Change my variable to 1', 'Show variable', and 'Hide variable'. A 'Create on list' button is also visible. In the center workspace, a script block is assembled: 'when green flag clicked' (yellow), 'say Hello for 2 seconds' (purple), and 'say Hello' (orange). The 'say Hello' block is highlighted with a yellow border. On the right, the stage area shows the Scratch cat character with a speech bubble saying 'Гладен съм' (I'm hungry). The bottom right panel shows the 'Sprites' area with 'Sprite 1' selected.



**Алгоритъм за размяна на стойности**

# Алгоритъм за размяна на стойности

Целта ни е да  
разменим  
стойностите за  
**A** и **B**



Временно  
присвояваме  
стойността от **A**  
на помощната  
променлива **C**




Сменяме  
стойността  
на **A** с тази  
на **B**



Финалната стъпка е  
да прехвърлим  
стойността на **A**  
(която временно е  
запазена на **C**) в **B**



```
когато  е щракнато
  направи A на 100
  направи B на 80
  направи C на A
  направи A на B
  направи B на C
```

Алгоритъм за присвояване на стойности, приложен в Scratch

|   |     |
|---|-----|
| A | 80  |
| B | 100 |
| C | 100 |

Краен резултат

Помощната променлива остава със стойността на A

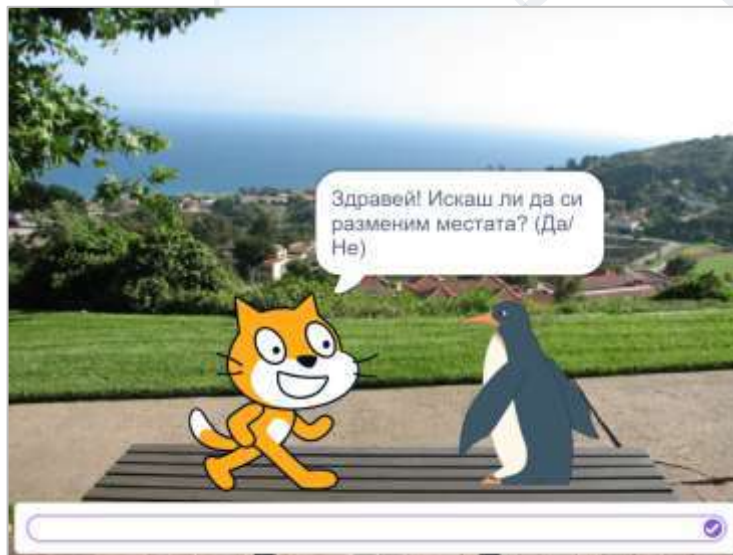


# Неудобната пейка

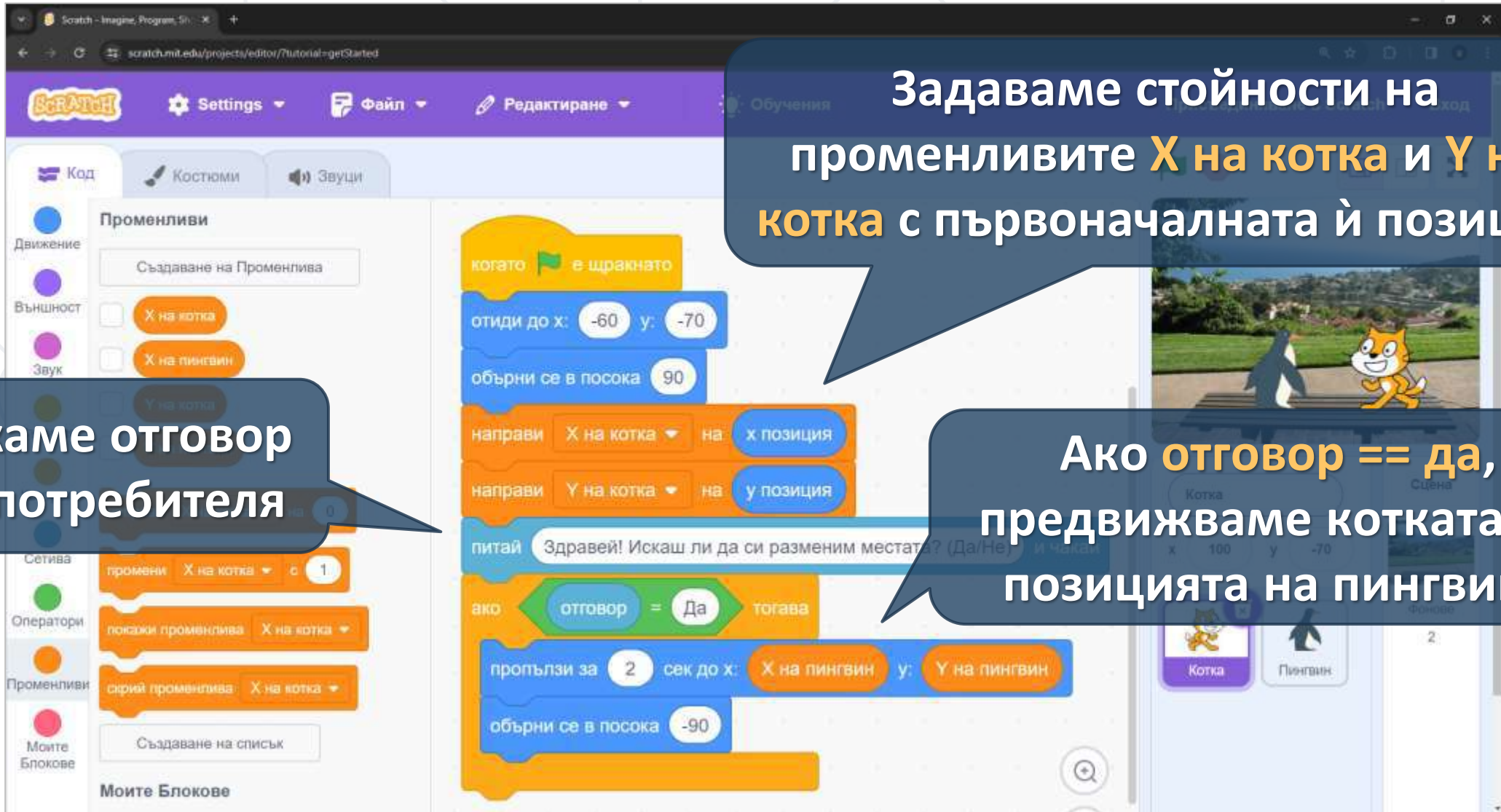
Използване на алгоритъм за размяна на стойности

# Неудобната пейка – условие

- Създайте програма, в която двама герои - **котка** и **пингвин** си **разменят местата** на пейка. Направете 4 променливи – **X** и **Y** на котката, както и **X** и **Y** на пингвина. Котката трябва да **пита пингвина** дали искат да си разменят местата. Ако котката (потребителят) отговори с "**Да**", героите си **разменят местата** (координатите).



# Неудобната пейка – код на котката



Задаваме стойности на променливите **X на котка** и **Y на котка** с първоначалната ѝ позиция

Чакаме отговор от потребителя

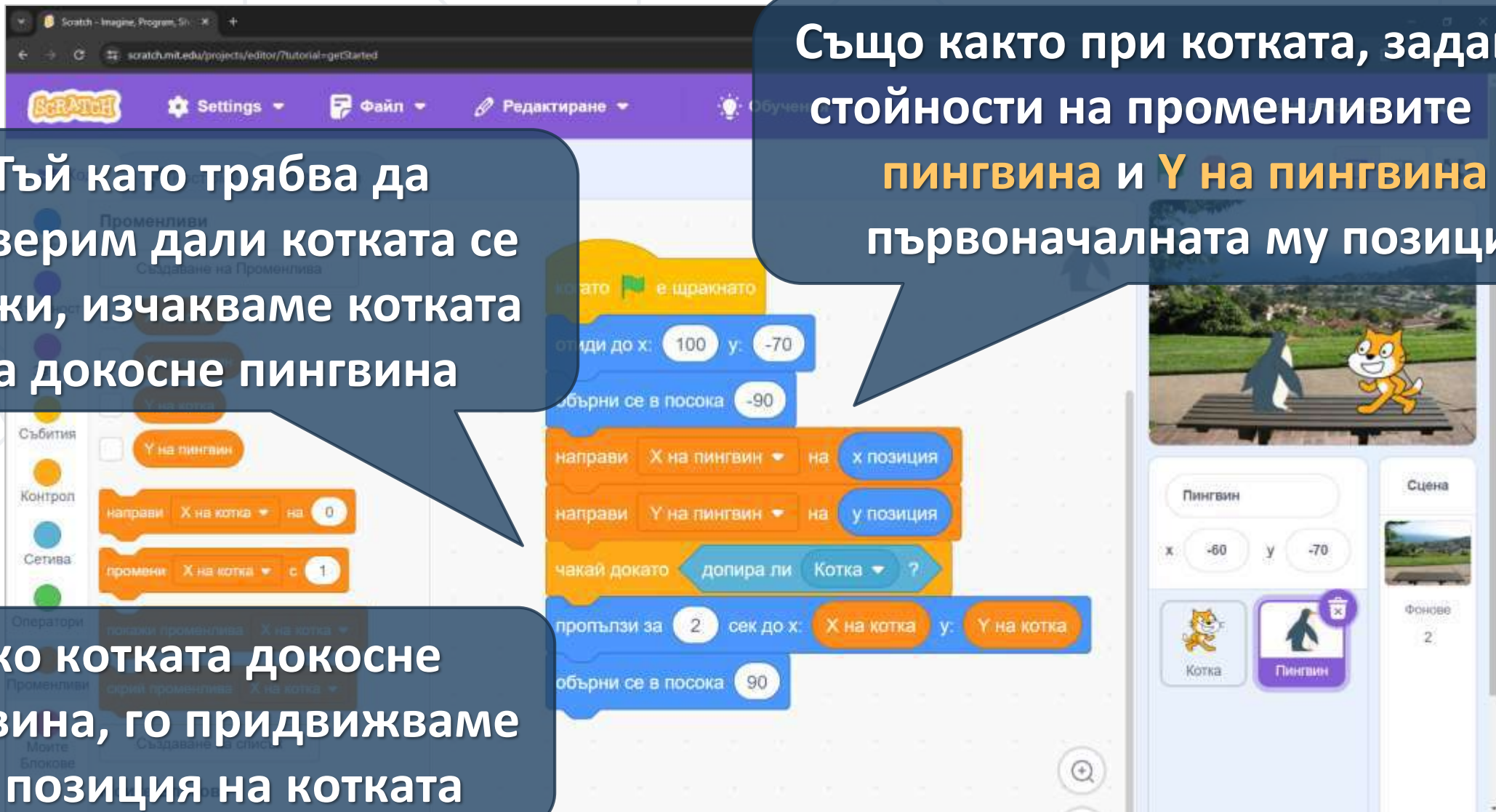
Ако **отговор == да**, предвижваме котката до позицията на пингвина

# Неудобната пейка – код на пингвина

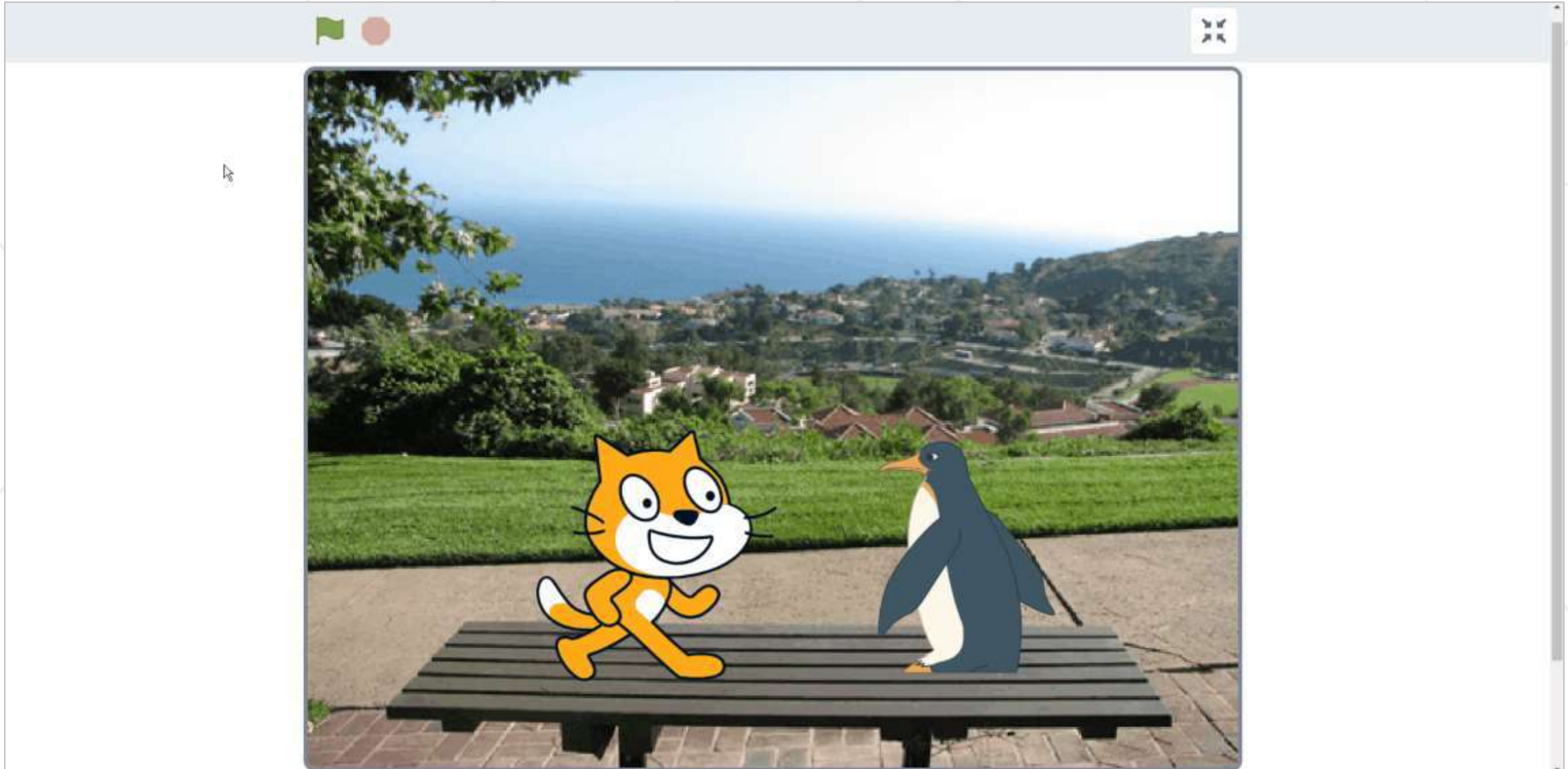
Тъй като трябва да проверим дали котката се движи, изчакваме котката да докосне пингвина

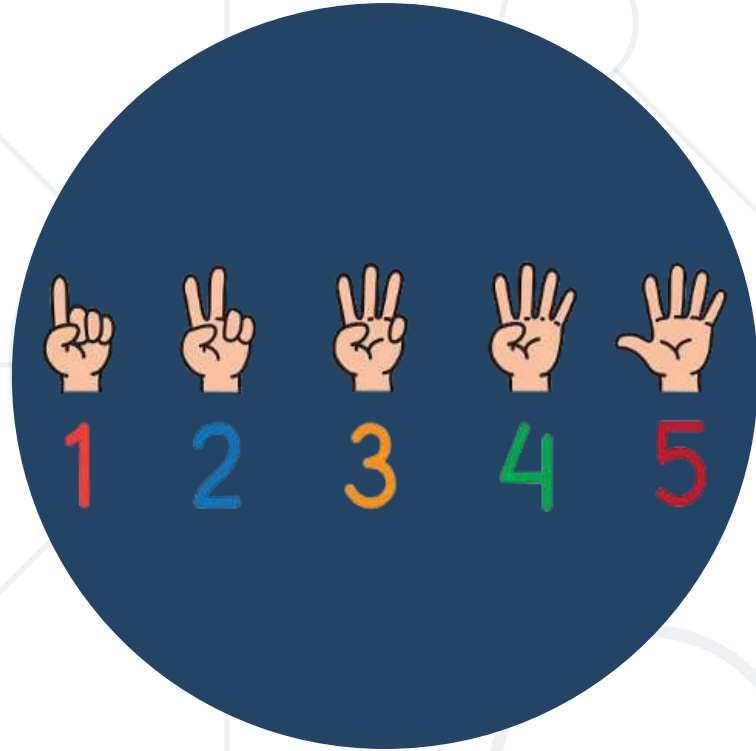
Ако котката докосне пингвина, го придвижваме до позиция на котката

Също както при котката, задаваме стойности на променливите **X на пингвина** и **Y на пингвина** с първоначалната му позиция



# Неудобната пейка – видео

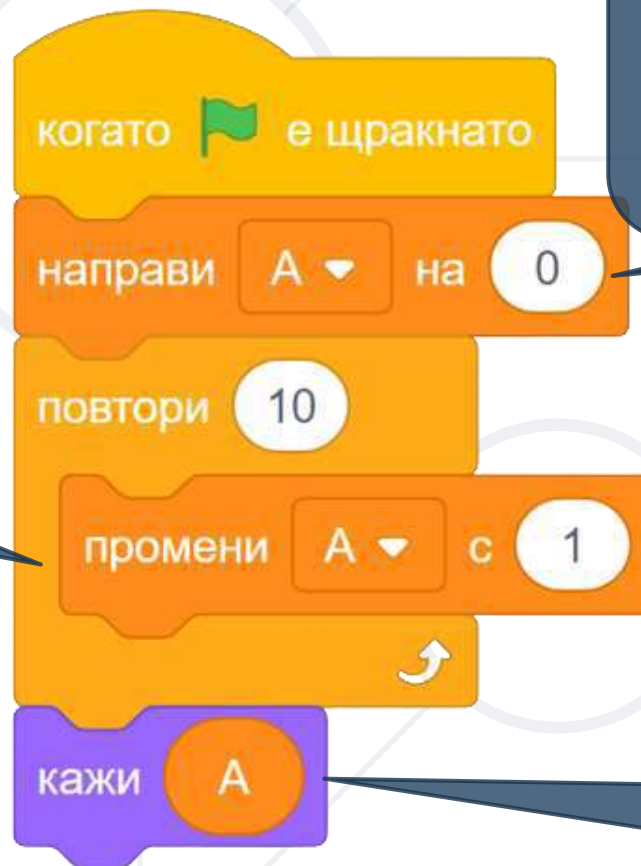




# Алгоритъм за броене на елементи

# Алгоритъм за броене на елементи

След което трябва да увеличаваме или намаляваме стойността на **променливата** според елементите, които следим



Когато броим дадени елементи, трябва да зададем начална стойност на променливите

Какво ще каже героят накрая?

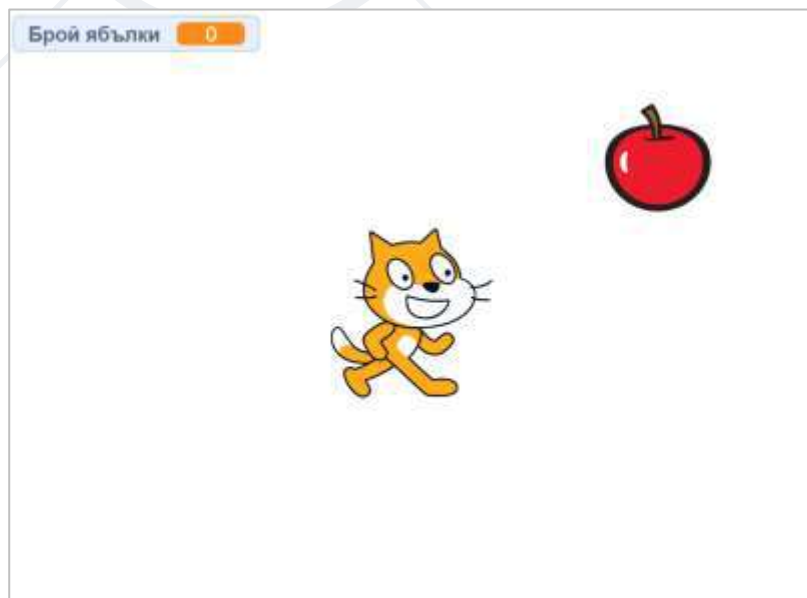


# Крадецът на ябълки

Използване на алгоритъм за броене на елементи

# Крадецът на ябълки – условие

- Създайте игра, в която котка яде ябълки, появяващи се на **случайно място**. Броят на изядените ябълки да се вижда на сцената, а котката **да следва показалеца на мишката**.



# Крадецът на ябълки – код на ябълката

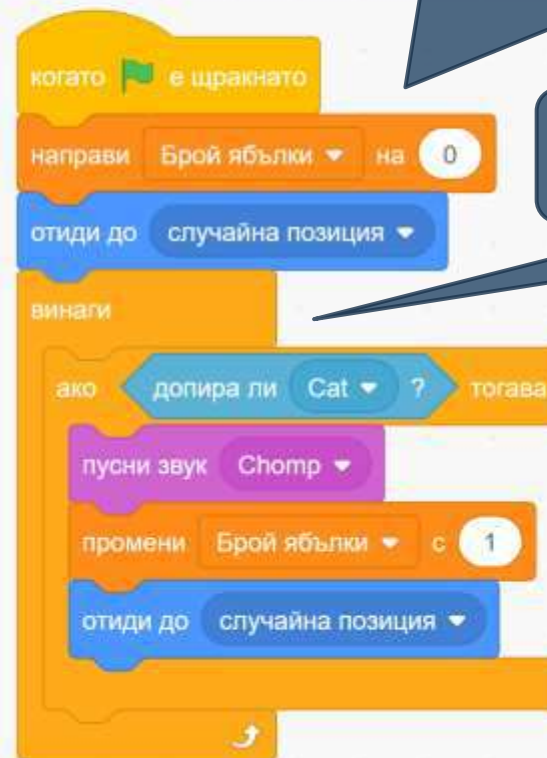
Поставяме я на  
**случайна позиция** на  
сцената

Създаваме променлива на ябълката  
и ѝ присвояваме начална **стойност 0**

Ако условието е  
изпълнено, се изпълнява  
кодът вътре (**издава се  
звук, увеличаваме броя  
на изядените ябълки и  
слагаме нова позицията  
на ябълката**)

Създаваме **безкраен цикъл**

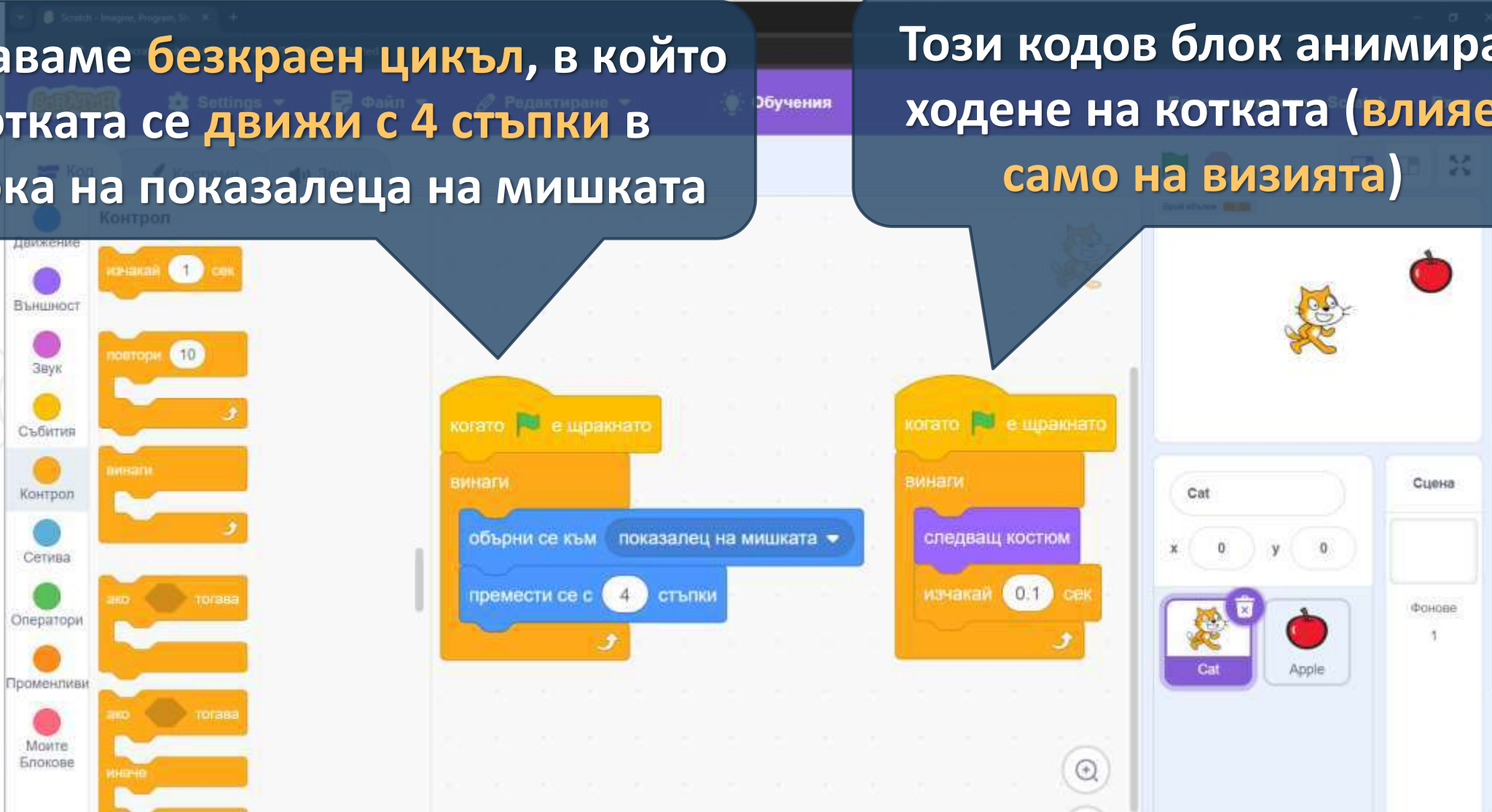
В цикъла проверяваме  
дали **котката допира  
ябълката**



# Крадецът на ябълки – код на котката

Създаваме **безкраен цикъл**, в който котката се **движи с 4 стъпки** в посока на показалеца на мишката

Този кодов блок анимира ходене на котката (**влие само на визията**)

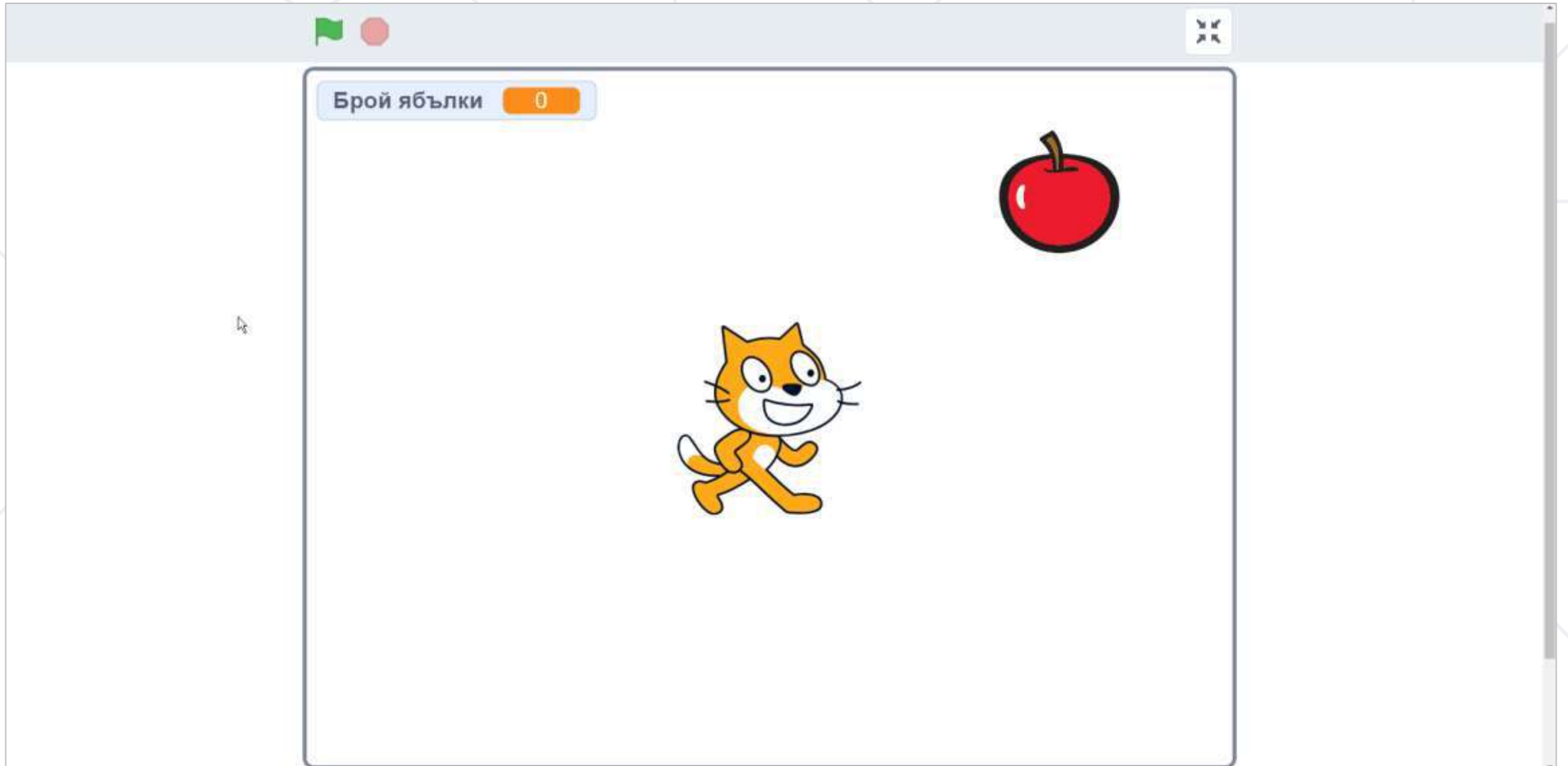


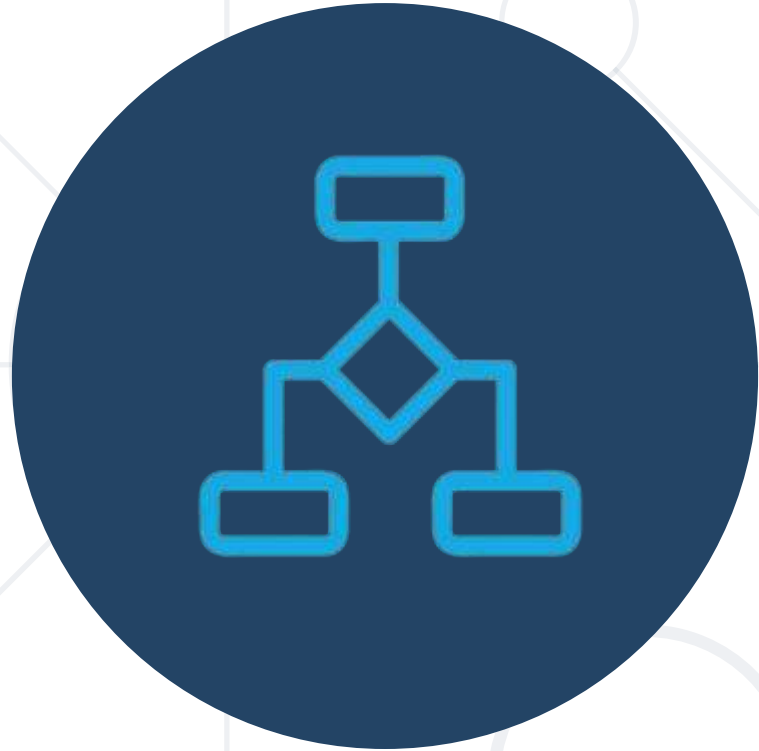
The image displays the Scratch code editor interface. On the left, the 'Control' category is selected in the block palette. Two code blocks are shown in the workspace:

- Block 1:** A yellow 'когато е щракнато' (when clicked) block with a green flag icon. It contains a blue 'винаги' (forever) loop block with two sub-blocks: 'обърни се към показалец на мишката' (turn to mouse cursor) and 'премести се с 4 стъпки' (move 4 steps).
- Block 2:** A yellow 'когато е щракнато' (when clicked) block with a green flag icon. It contains a blue 'винаги' (forever) loop block with two sub-blocks: 'следващ костюм' (next costume) and 'изчакай 0.1 сек' (wait 0.1 seconds).

The background shows a Scratch stage with a cat character and an apple. The 'Cat' sprite is selected, and its 'x' and 'y' coordinates are 0. The 'Apple' sprite is also visible. The 'Сцена' (Stage) and 'Фонове' (Backgrounds) panels are also visible.

# Крадецът на ябълки – видео





**Алгоритъм за намиране на най-големия от три елемента**

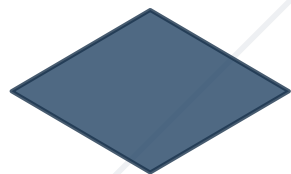
# Какво представлява блок-схема?



– Начало и край на алгоритъм



– Описание на вход и изход



– Проверка на условие (Да/Не)



– Описание на действие



– Посочване на следваща стъпка в алгоритъма

# Алгоритъм за намиране на най-големия от три елемента

Ако  $A > B$ , си разменят стойностите ( $B$  вече ще съдържа по-голяма стойност от  $A$ )

Създаваме променливите и им даваме стойности

Начало

$A, B, C$

$A > B$

Размяна на  $A$  и  $B$

$B > C$

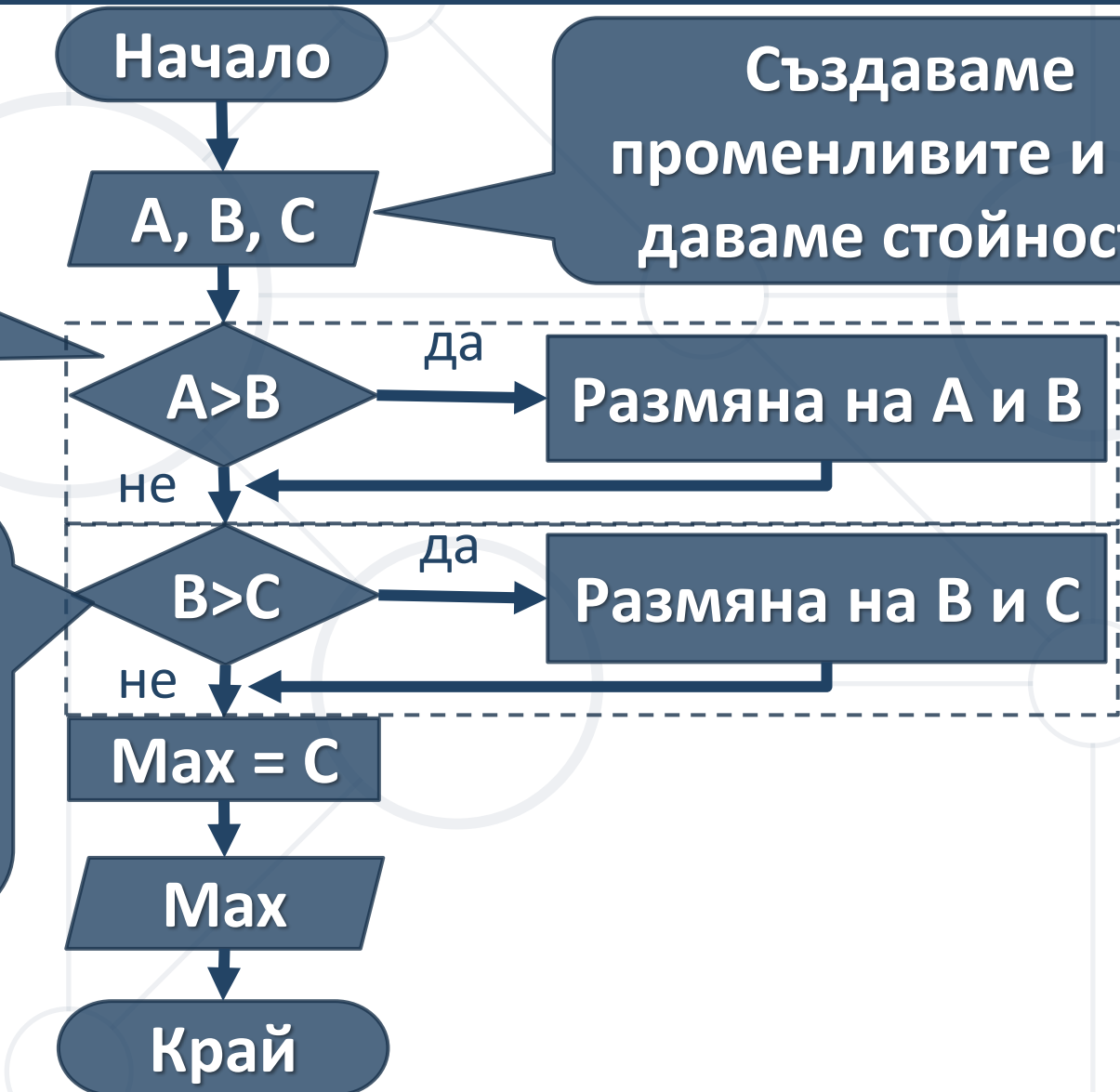
Размяна на  $B$  и  $C$

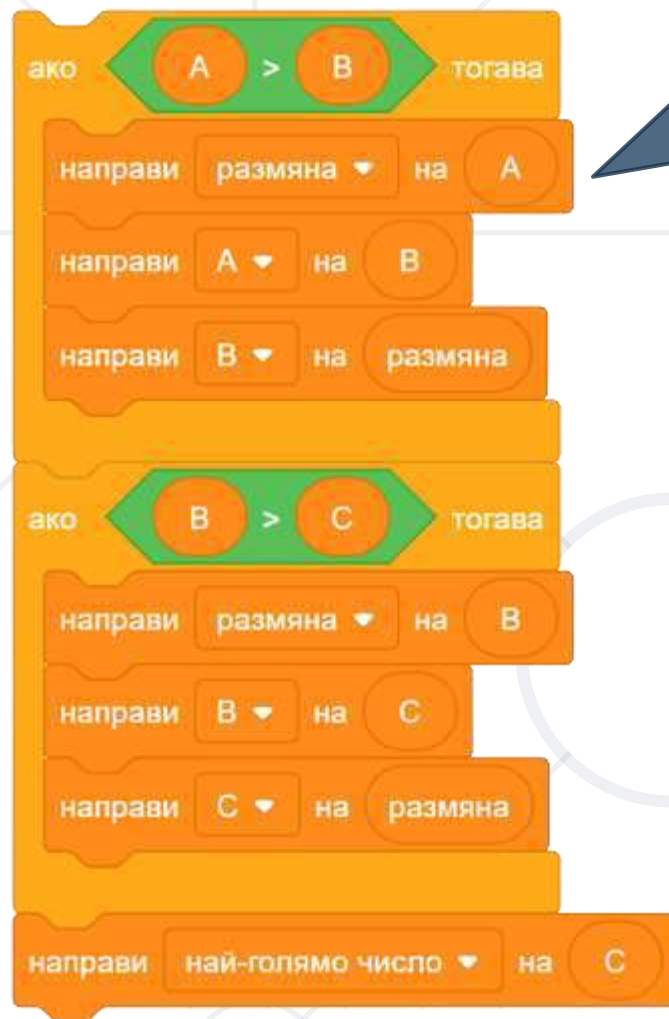
$Max = C$

$Max$

Край

По същия начин се сравняват стойностите на  $B$  и  $C$ . Ако  $B > C$ , ще си разменят стойностите ( $C$  вече ще е с най-голямата стойност)





```
ако (A > B) тогава
  направи размяна на A
  направи A на B
  направи B на размяна
ако (B > C) тогава
  направи размяна на B
  направи B на C
  направи C на размяна
направи най-голямо число на C
```

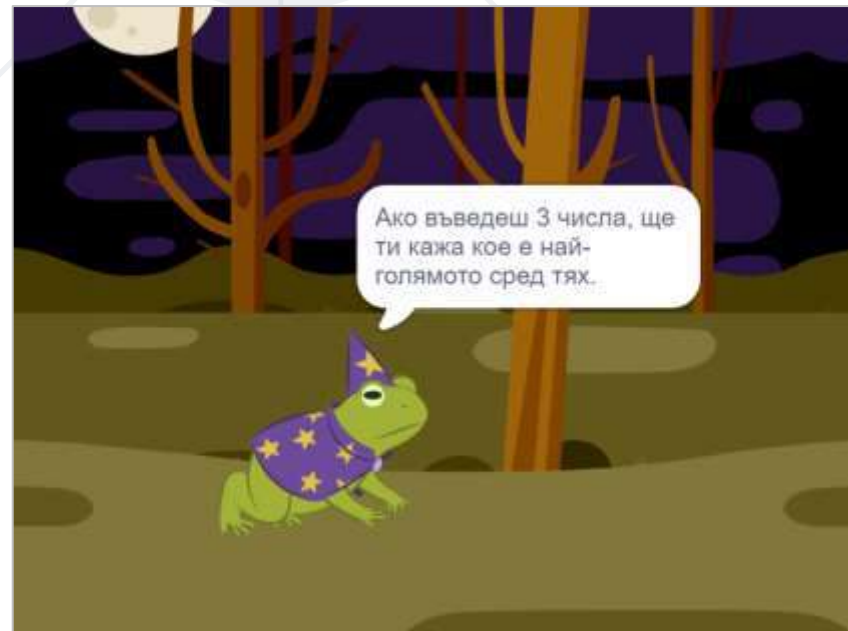
Алгоритъм за намиране на най-големия елемент, проложен в Scratch



# Магическата жаба

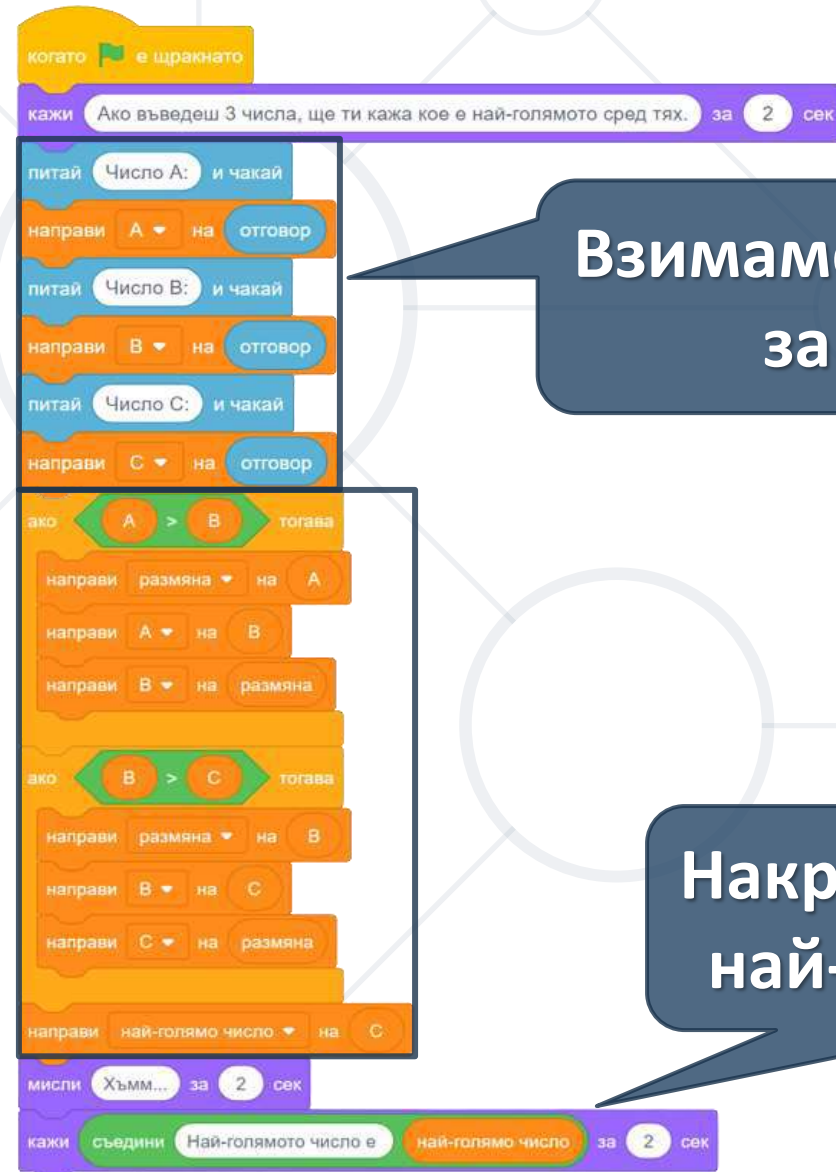
Използване на алгоритъм за намиране на  
най-големия от три елемента

- Направете програма за намиране на най-големия от три елемента. Героят трябва да е **магическа жаба**, която взима **3 числа** от потребителя. Накрая жабата трябва да **каже** кое е **най-голямото число**. Добавете и подходящ фон.



# Магическата жаба – код

Прилагаме алгоритъма  
за най-голямо число

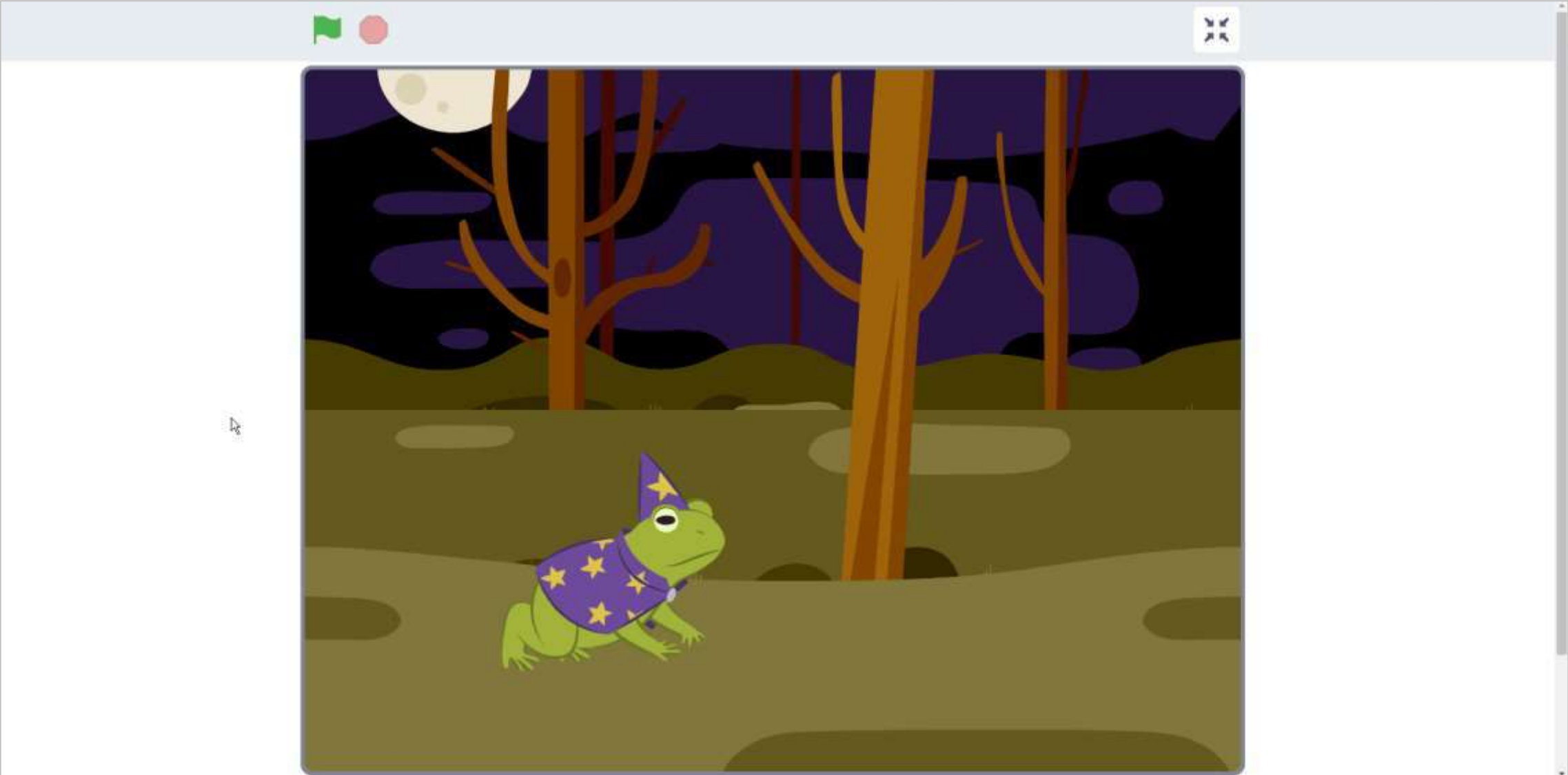


```
когато е щракнато
кажи Ако въведеш 3 числа, ще ти кажа кое е най-голямото сред тях. за 2 сек
питай Число A: и чакай
направи A на отговор
питай Число B: и чакай
направи B на отговор
питай Число C: и чакай
направи C на отговор
ако A > B тогава
направи размяна на A
направи A на B
направи B на размяна
ако B > C тогава
направи размяна на B
направи B на C
направи C на размяна
направи най-голямо число на C
мисли Хъмм... за 2 сек
кажи съедини Най-голямото число е най-голямо число за 2 сек
```

Взимаме стойностите  
за A, B и C

Накрая казваме кое е  
най-голямото число

# Магическата жаба – видео





# Алгоритъм за сортиране на елементи

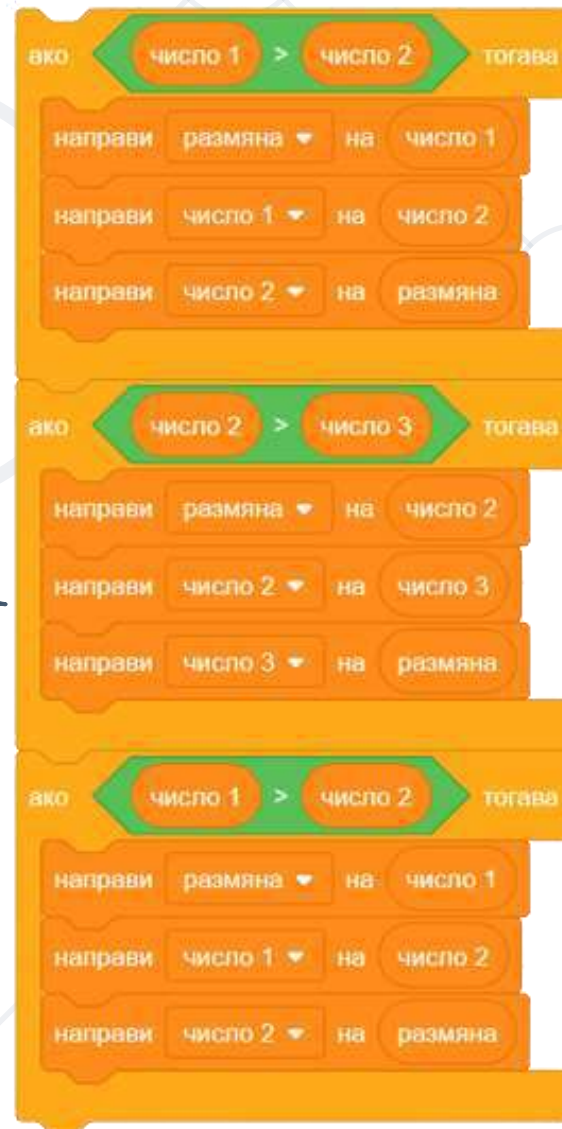
# Алгоритъм за сортиране на елементи

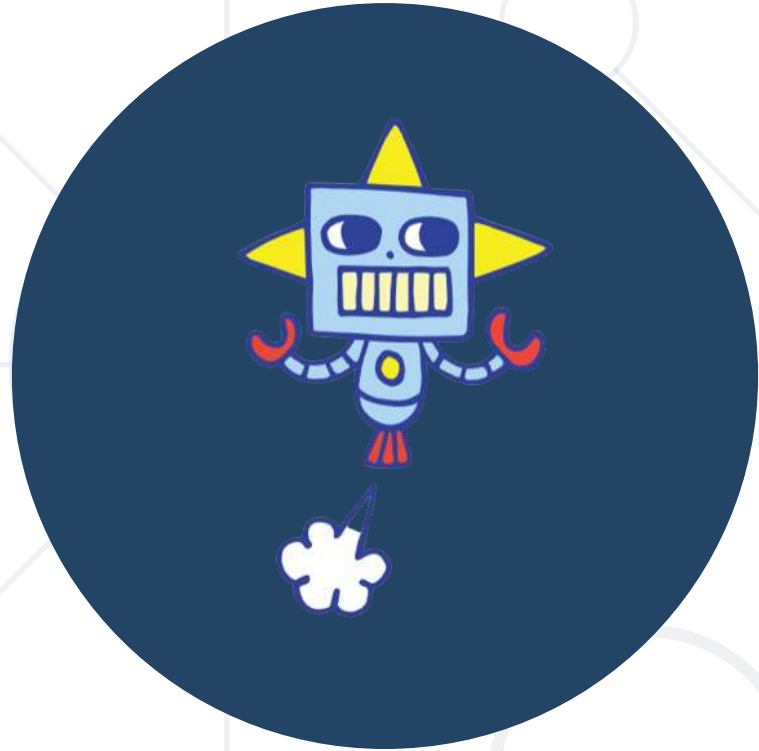
Прилагаме алгоритъма за намиране на **най-голямото** от трите числа (в **C** ще се съдържа **най-голямото** число)

Последно сравняваме **A > B** и ги разменяме, ако е изпълнено условието



Алгоритъм за сортиране на елементи, приложен в Scratch





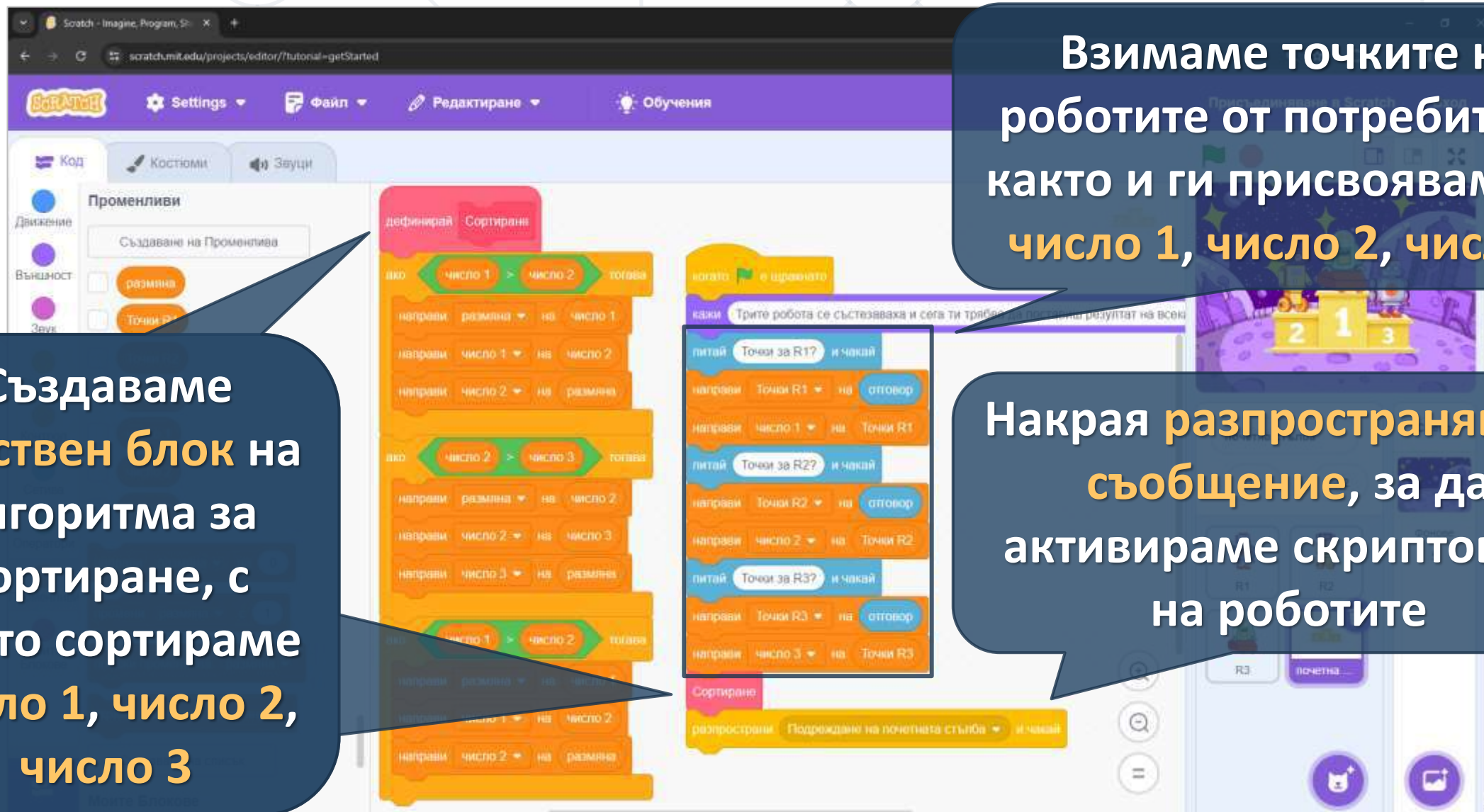
# Робо маратон

Използване на алгоритъм за сортиране

- Създайте **сортираща игра**, която да подрежда **3 робота** на почетната стълба. Намерете подходяща **снимка на подиум** в интернет и **създайте герой** със снимката. Подиумът трябва да **вземе резултатите** на роботите от **потребителя** и да ги **сортира**. Накрая състезателите трябва да се **наредят по подиума** според техния резултат (**низходящо**). Добавете подходящ фон.



# Робо маратон – код на подиума



Създаваме собствен блок на алгоритма за сортиране, с който сортираме число 1, число 2, число 3

Взимаме точките на роботите от потребителя, както и ги присвояваме на число 1, число 2, число 3

Накрая разпространяваме съобщение, за да активираме скриптовете на роботите

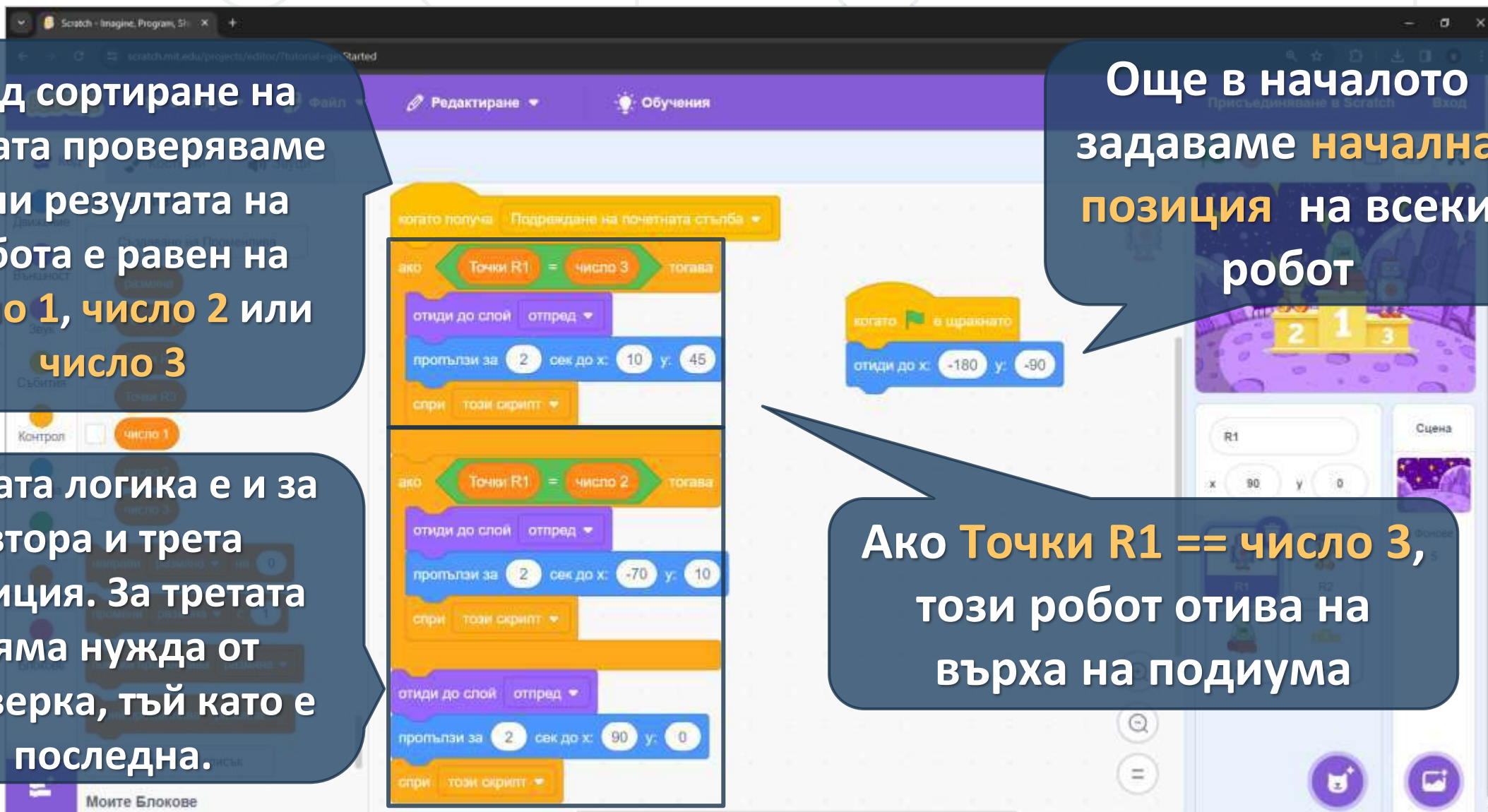
# Робо маратон – код на роботите

След сортиране на числата проверяваме дали резултата на работа е равен на **число 1**, **число 2** или **число 3**

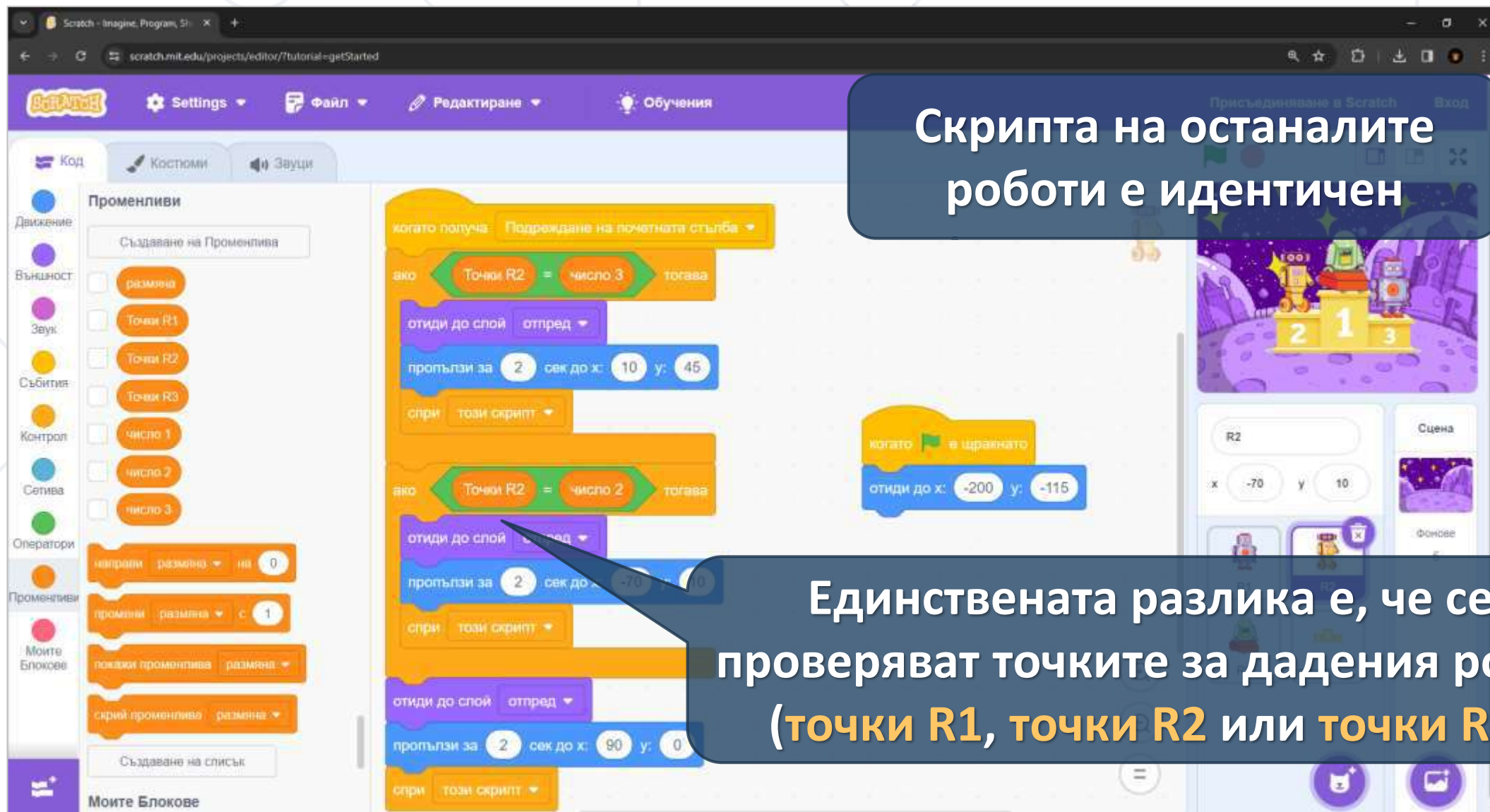
Същата логика е и за втора и трета позиция. За третата няма нужда от проверка, тъй като е последна.

Още в началото задаваме **начална позиция** на всеки робот

Ако **Точки R1 == число 3**, този робот отива на върха на подиума



# Робо маратон – код на роботите



Скрипта на останалите роботи е идентичен

Единствената разлика е, че се проверяват точките за дадения робот (точки R1, точки R2 или точки R3)

# Робо мартон – видео



- **Променлива** – стойност, записана в паметта на Scratch
- **Елементи** на променливата:
  - Име
  - Стойност
- **Алгоритми** за:
  - Размяна на стойности
  - Броене на елементи
  - Най-голям елемент
  - Сортиране на елементи



# Въпроси?



SoftUni



Software  
University



SoftUni  
Svetlina



SoftUni  
Creative



SoftUni  
Digital



SoftUni  
Foundation



SoftUni  
Kids

- Този курс (слайдове, примери, демонстрации, видеа, домашни и др.) притежават лиценза "[Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International](#)"

