

ЛЕКЦИЯ I © 2020 Нет Ит

# JAVA 101: Стрингове

Теодор Костадинов



SOFTWARE  
ACADEMY



# Съдържание

1. Особенности
2. Сравняване
3. Конкатенация
4. StringBuilder
5. Форматиране





# Особености НА СИМВОЛНИТЕ НИЗОВЕ



# Особености

- Символните низове (strings, стрингове) са поредица от СИМВОЛИ.
- В Java за обработка на низове се използва класът String.
- String course = "Intro to Java";
- Символните низове се записват като последователност от символи, оградена в кавички.
- Кавичките не са част от стойността на низа!
- String е клас, а не прост тип (каквито например са int или boolean).
  - Променливите могат да имат стойност null.
  - Сравняват се по различен начин от простите типове.
  - Класът String има методи за различни действия с низове (разгледайте ги).



# Инициализация НА СИМВОЛНИТЕ НИЗОВЕ



# Инициализация

Можем да присвоим стойност на променлива от тип String по няколко начина:

```
String name = "Ivan";  
String job = new String("programmer");  
String two = 2 + "";  
String three = Integer.valueOf(3);
```

Можем да преобразуваме друг тип в низ, като го съберем с низ.







# Събиране на низове

```
String firstName = "Ivan";  
String lastName = "Petrov";
```

Използване на знак „+“ за слепване на низове:

```
String fullName = firstName + " " + lastName; // Ivan Petrov
```

Използване на метода „concat()“ за слепване на низове:

```
String fullName2 = lastName.concat(", ").concat(firstName); // Petrov, Ivan
```





# Immutability

При класа String поредицата от символи , записана в паметта, не се изменя (нарича се immutable). При промяна на променливата, съдържанието не се променя, а се създава ново място в паметта, в което е записана новата стойност.

Затова не се препоръчва долепяне на низове в цикъл!

За работа в низове, когато имаме изменение на стойността на низа, се препоръчва използването на класа StringBuilder.

```
StringBuilder builder = new StringBuilder();  
builder.append("Text");  
String result = builder.toString();
```

# Методи НА СИМВОЛНИТЕ НИЗОВЕ





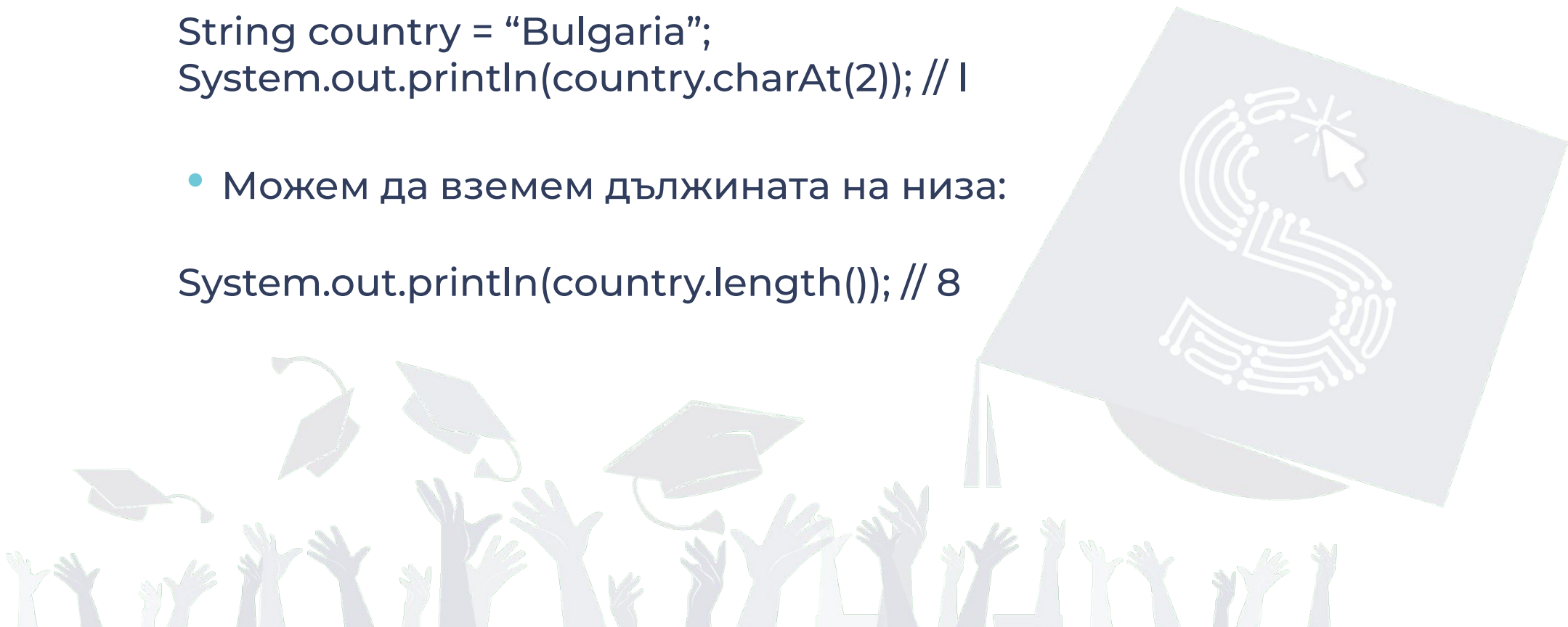
# charAt

- Можем да достъпваме символите на низа по индекси. Индексирането започва от 0, както при масивите:

```
String country = "Bulgaria";  
System.out.println(country.charAt(2)); // I
```

- Можем да вземем дължината на низа:

```
System.out.println(country.length()); // 8
```



- Прочетете името на потребителя от конзолата. Отпечатайте в конзолата първата и последната буква и дължината му.
- За четене на символен низ от конзолата може да се използва методът `readLine()` на класа `Scanner`



## Ами сега?

Задача за упражнение



# Важно - NULL

---

Винаги, когато работите с променливи от тип String, имайте предвид, че може да имат стойност null и винаги правете проверки!



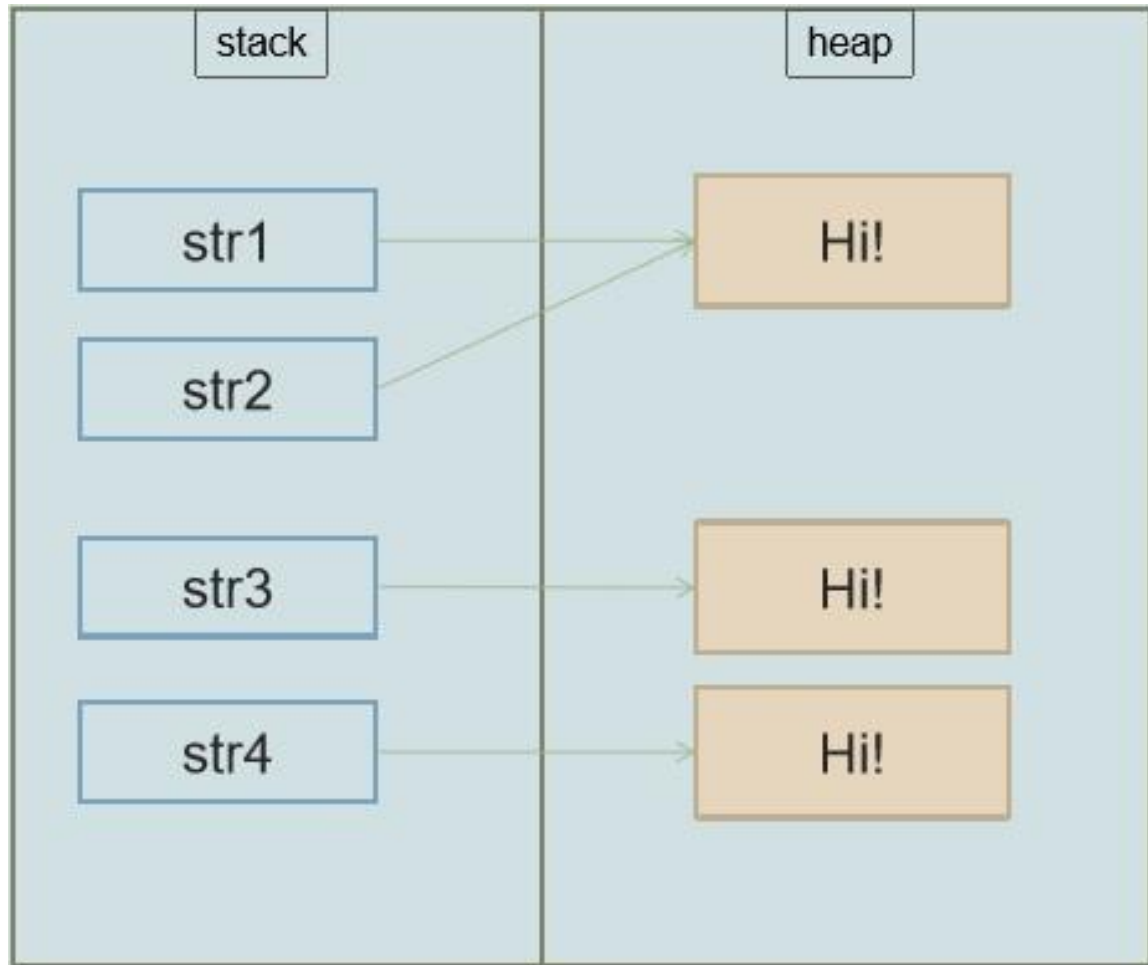
# equals

Тъй като String е клас, символните низове в Java не се сравняват с оператора „==“. При използване на „==“, се сравняват техните адреси, а не стойности.

```
String str1 = "Hi!";  
String str2 = "Hi!";  
  
boolean equal = str1 == str2;  
  
System.out.println(equal); // true  
  
String str3 = new String("Hi!");  
String str4 = new String("Hi!");  
  
equal = str3 == str4;  
  
System.out.println(equal); // false
```

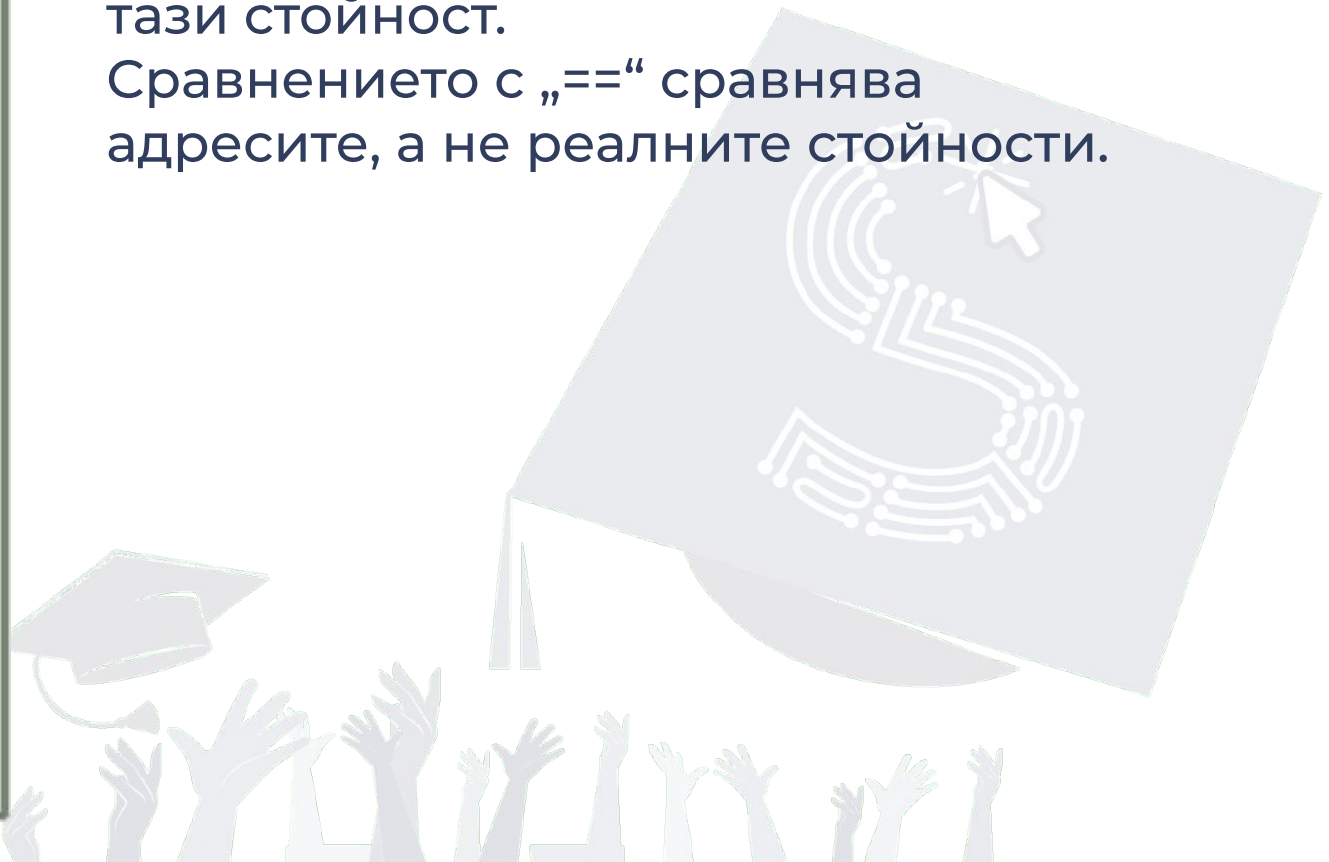


# equals



Променливата не запазва в себе си стойността на низа, а сочи към място в паметта, където се пази тази стойност.

Сравнението с „==“ сравнява адресите, а не реалните стойности.



# equals

За сравнение на стойностите на низове се използва методът equals()

```
String str3 = new String("Hi!");  
String str4 = new String("Hi!");  
  
equal = str3.equals(str4);  
  
System.out.println(equal); // true
```





Напишете програма, която очаква от потребителя да въведе името на любимия си език за програмиране в конзолата.

Ако въведеният език е Java, извежда „Good choice!“ и „Try again!“ в противен случай.



Ами сега?

Задача за упражнение

# format

Метод `String.format()` :

Задава се шаблон с места, в които да се попълват данните от различни типове.

Задават се като следващи параметри на метода данните в последователността, в която трябва да бъдат попълнени.

```
String.format("Name: %s, Age: %d years", "Ivan", 25);  
//Name: Ivan, Age: 25 years
```



Използвайки `String.format()`, напишете следното (подчертаните думи да се зададат като места за попълване в шаблона).

```
//Name: Ivan, Age: 25 years, Lives in Vratsa
```



## Ами сега?

Задача за упражнение



Почивка

до 19:50



# split

Методът split разделя един Стринг на по-малки стрингове, спрямо знак, който му е подаден.

Резултатът се записва в масив от стрингове.

Пример:

```
String text = "This is text. Look at that text. Text text."  
String[] result = text.split("t");  
// result = {"This is ", "ex", ". Look a", " ", "ha", " ", "ex", " ", "ex", "."}
```



# indexOf

Методът `indexOf` му се подава текст и знак или дума, която трябва да намери в този текст.

Резултатът е число - индексът на първото срещане на търсеният символ в текста.

Пример:

```
String text = "This is text. Look at that text. Text text."  
int search = text.indexOf("t"); //search = 8
```





- Напишете метод, който приема като параметри 2 символни низа и връща първата позиция, на която вторият низ се съдържа в първия.
- Пример:  
`firstPosition("I like Java.", "Java") = 7`



## Ами сега?

Задача за упражнение

Продължение:

1. Подайте `null` като стойност на първия параметър и вижте какъв е резултатът.
2. Нека методът да връща `-5`, ако някоя от стойностите на подадените низове е `null`.
3. Нека методът да връща `-3`, ако първият низ е празен (има дължина 0).



Ами сега?

Задача за упражнение

# replace

Методът `replace` приема три параметъра - текстът върху който работи, думата която ще заменя и думата с която ще бъде заменена.





# substring

---

Методът `substring` реже един стринг по подаден индекс и създава нов, по-къс стринг.



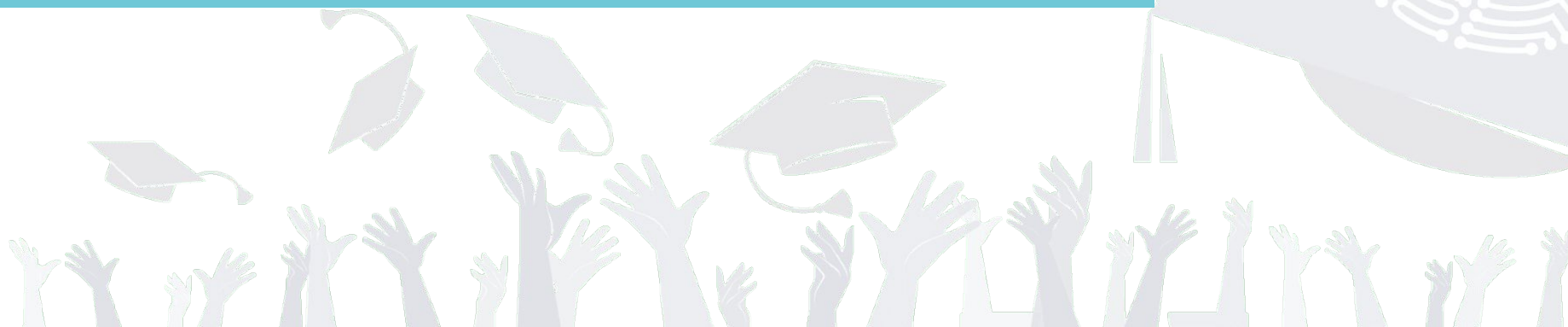
ВЪПРОСИ?



# Резюме



- С текстове може да се прави всичко
- Текстовете заемат много памет и трябва да се внимава как се обработват.

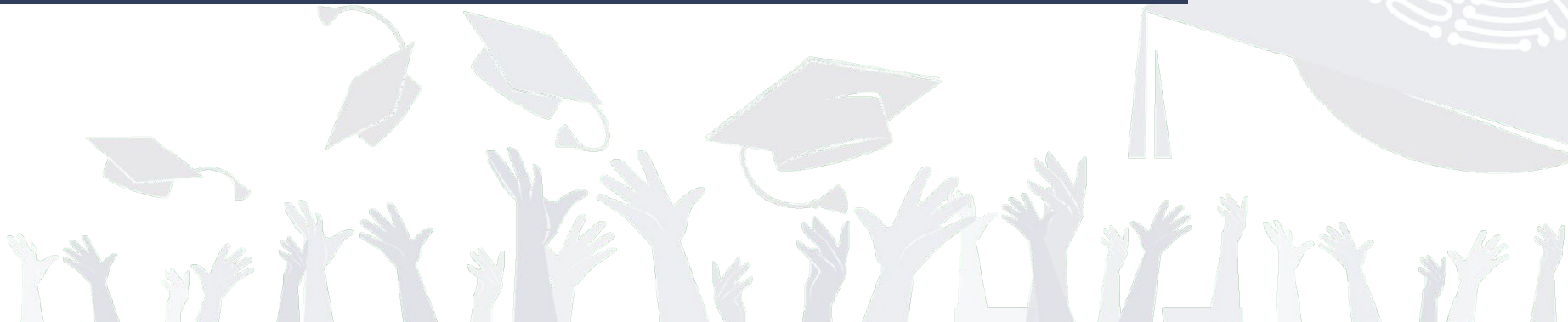




# Ресурси



- [Docs](#)
- [GitHub Repo with Demos](#)
- 



# Задачи за упражняване



# Задача 1

Даден е масив, който съдържа адрес за изпращане на писмо. На всяка от позициите има съответно: [име на получател, държава, град, адрес, пощенски код]. Напишете метод, който приема като параметър такъв масив и връща като резултат низ, който съдържа всеки елемент от адреса на отделен ред.

Примерен резултат:  
Иван Иванов

България  
София  
Ул. Незабравка 12  
1000



Ами сега?

Задача за упражнение

# Задача 2

Напишете метод, който приема за параметър даден текст и връща масив, съдържащ думите на този текст.  
Напр: `wordify("Java is cool") -> ["Java", "is", "cool"]`



Ами сега?

Задача за упражнение





Това домашно влиза в крайната ви оценка!

# Домашно

Качвайте домашното си в ГитХъб и  
слагайте линка тук:

<https://forms.gle/AcvCptCbSDizr2Ay6>



# Задача 1



При въведен текст от потребителя, кажете дали започва и свършва по огледален начин. Например

Акара - да, започва и свършва с а

Ба123аб - да, започва с ба и свършва с аб

Изведете какъв е повтарящият се участък. Т.е. Изведете “а” или “ба” от горният пример.



# Задача 2



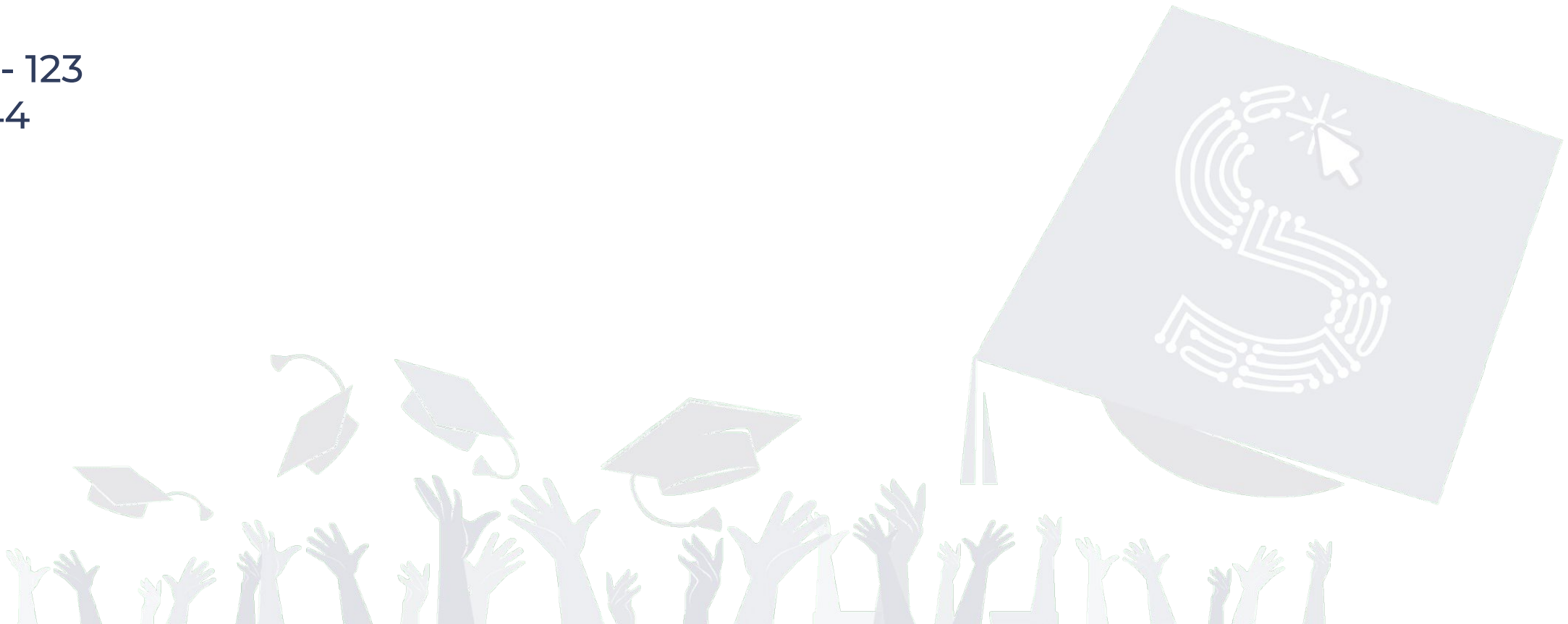
При подаден стринг, върнете сумата на числата (не цифрите!) в този текст.

Пример:

Abc123xyz - 123

Aa11b33 - 44

7 11 - 18



# Задача 3



При подаден стринг, пребройте думите завършващи на у и на z.

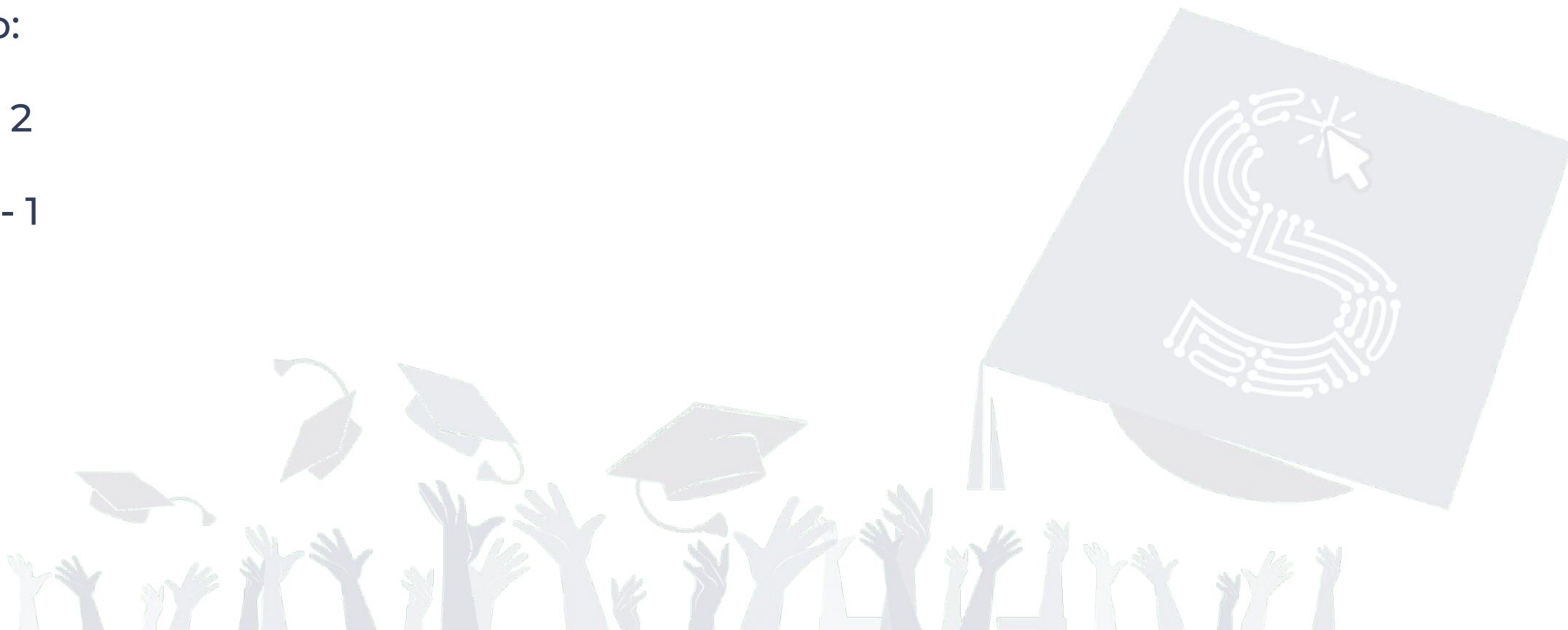
Например:

fez day - 2

day fyuyz - 2

Dad fez - 1

Yellow fez - 1





© 2020 Нет Ит

# БЛАГОДАРЯ ЗА ВНИМАНИЕТО!



SOFTWARE  
ACADEMY

