

ЛЕКЦИЯ I © 2020 Нет Ит

JAVA OOP: Комуникация между обекти

Теодор Костадинов



SOFTWARE
ACADEMY

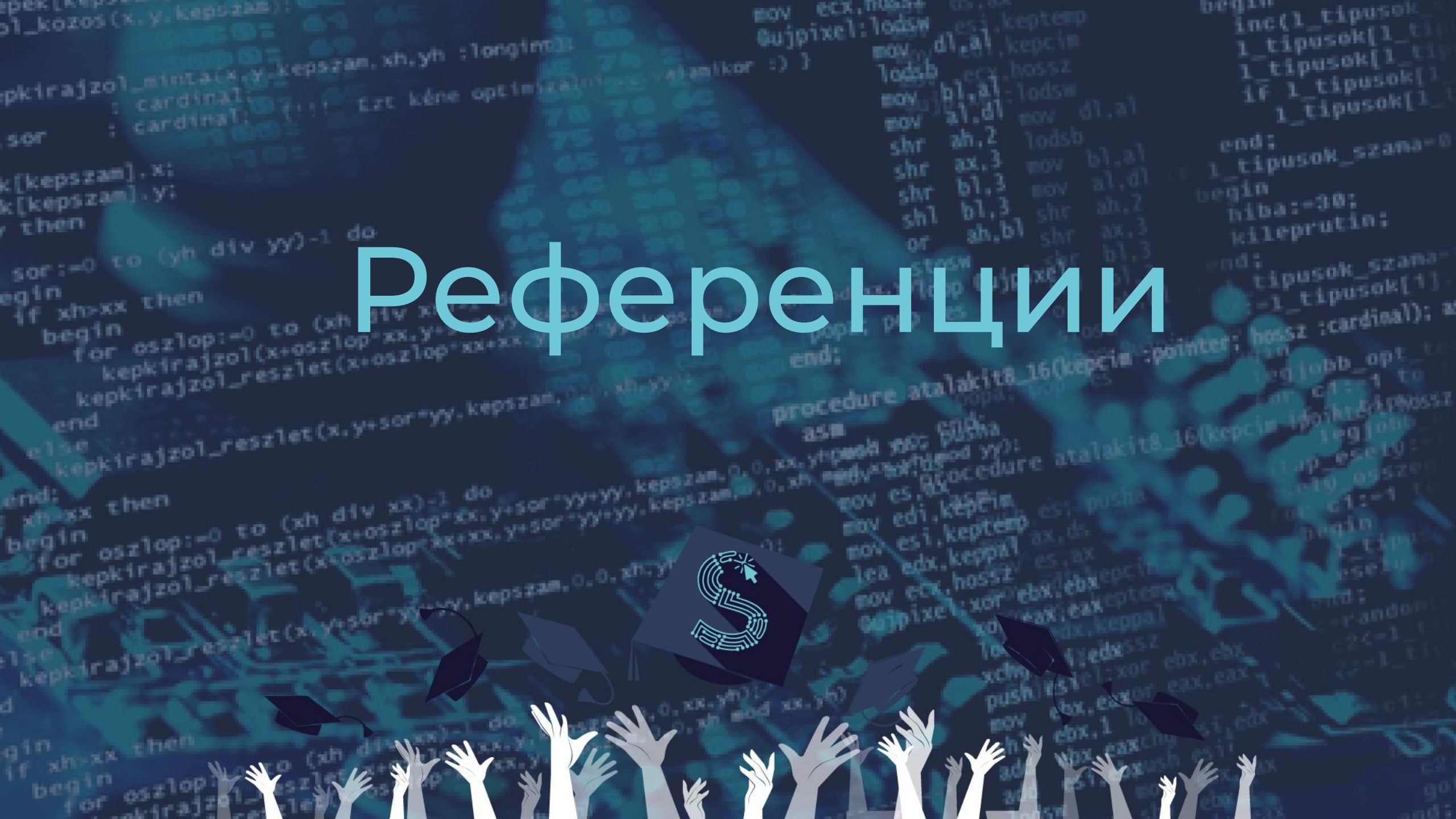


Съдържание

1. Референция
2. Композиция
3. Колбеци

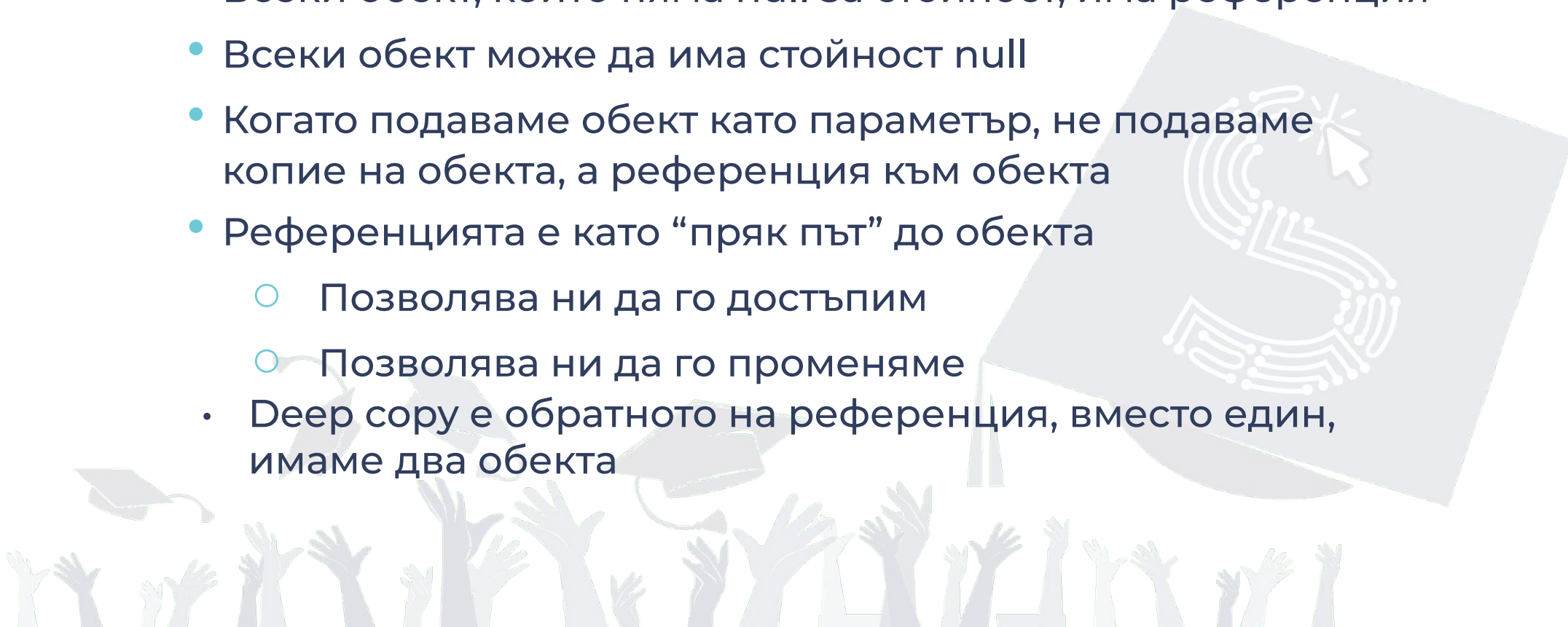


Референции

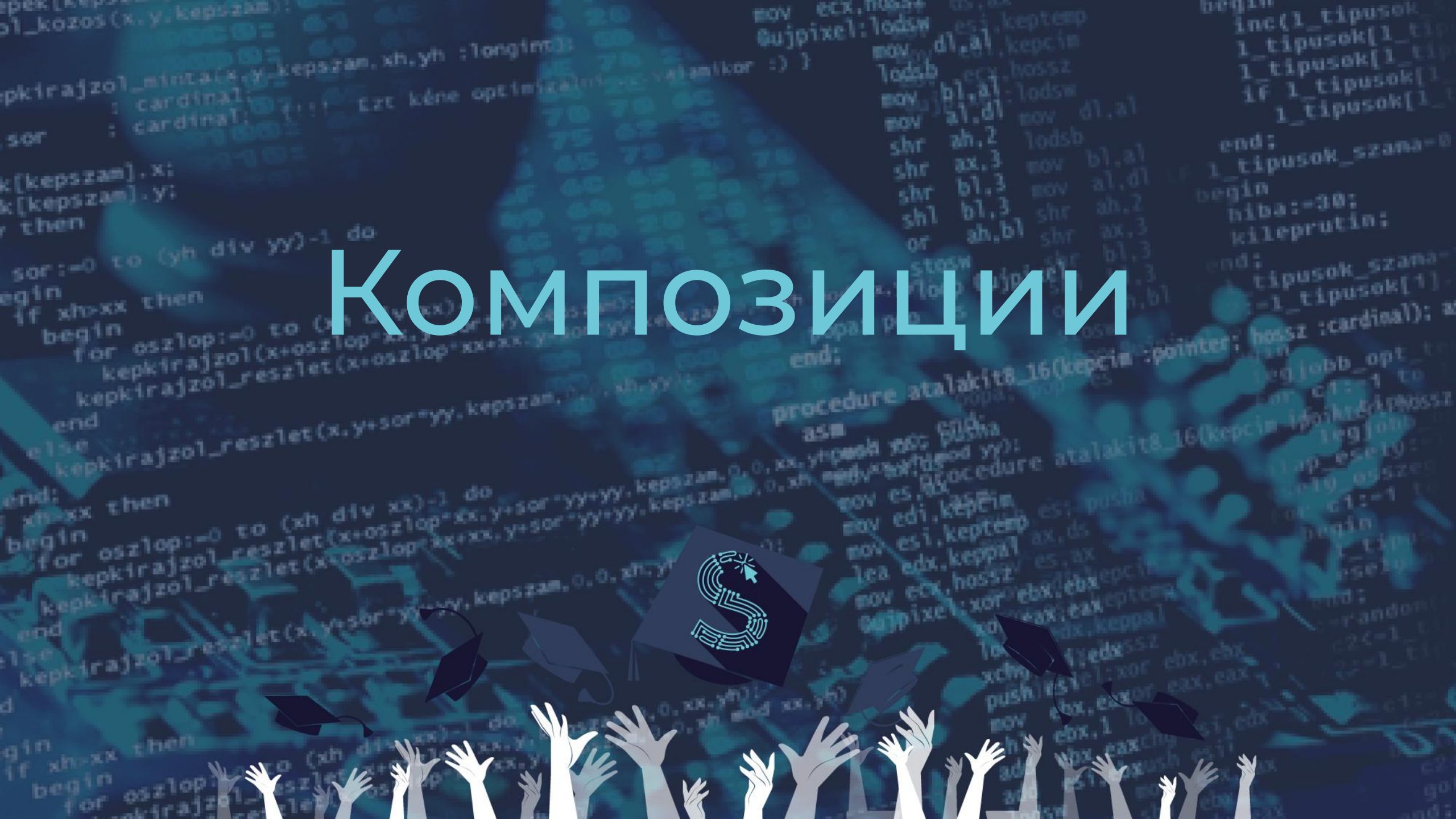


Референция vs Deep Copy

- Референция е адресът в паметта, където се запазват данните (променливи и методи) на един обект
- Всеки обект, който няма null за стойност, има референция
- Всеки обект може да има стойност null
- Когато подаваме обект като параметър, не подаваме копие на обекта, а референция към обекта
- Референцията е като “пряк път” до обекта
 - Позволява ни да го достъпим
 - Позволява ни да го променяме
- Деер сору е обратното на референция, вместо един, имаме два обекта



Композиции



Композиция

- Композицията е механизъм, който ни позволява да моделираме обект създаден от други обекти
- Създава се чрез “има” (“has-a”) връзка между тях
- Помага за преизползването на код
- Съставните обекти (тези, които се съхраняват в един обект) са унищожени, когато главния обект бъде унищожен

```
class Engine { ... }  
class Chassis { ... }  
class Wheel { ... }
```

```
class Car {  
    private Engine engine;  
    private Chassis chassis;  
    private Wheel[] wheels;  
}
```

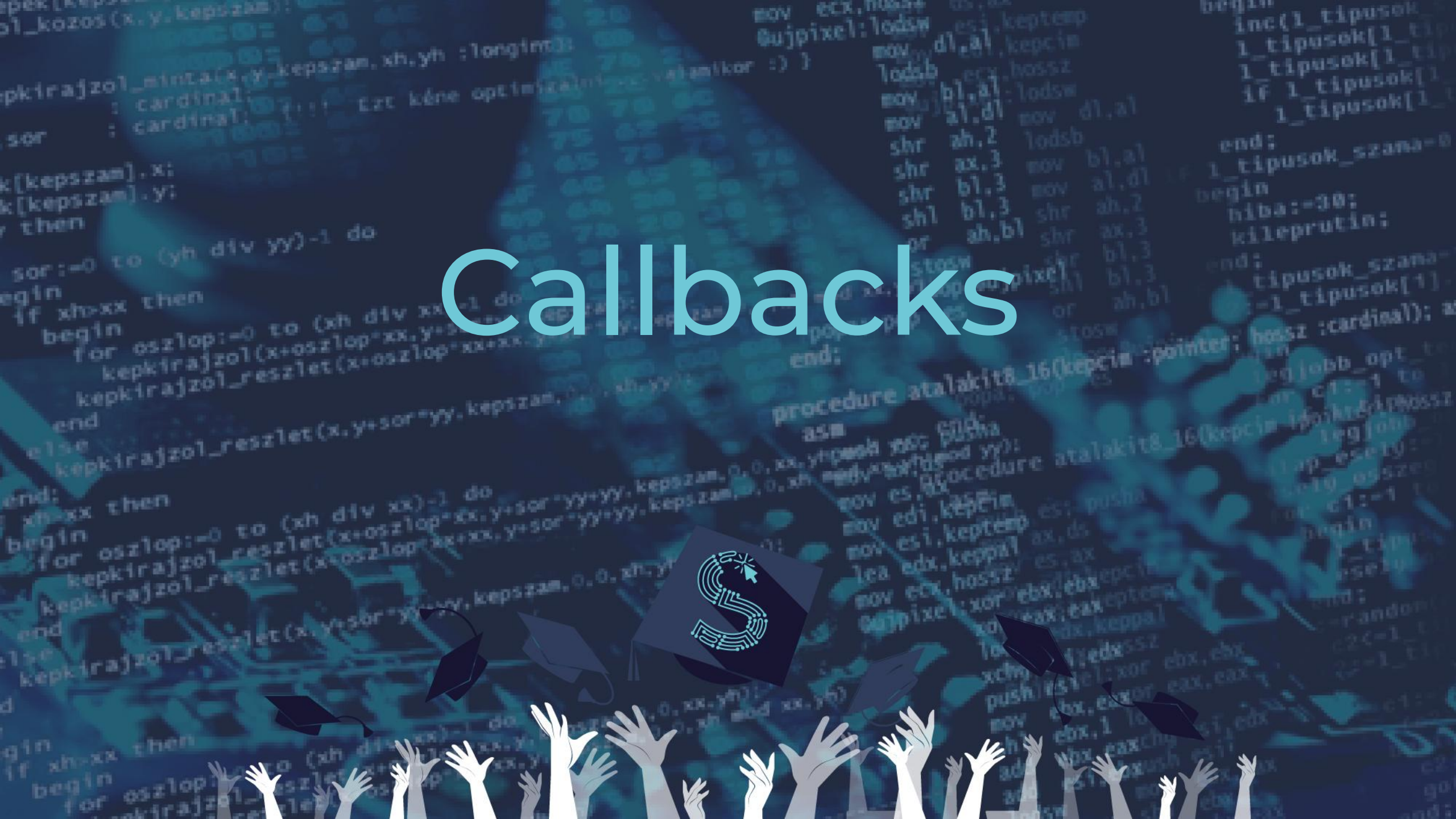
Композиция с абстракция

- Абстракцията вдига качеството на кода и прави КОМПОЗИЦИЯТА ПО-ГЪВКАВА

```
interface Engine { ... }  
interface Chassis { ... }  
interface Wheel { ... }  
class 4Cylinder implements Engine  
class Sedan implements Chassis  
class Wheel16 implements Wheel
```

```
class Car {  
    private Engine engine;  
    private Chassis chassis;  
    private Wheel[] wheels;  
}
```


Callbacks



Callbacks

- Парче код, което подаваме на друг код, което да бъде изпълнено в бъдещето
- “Другият” код изпълнява това парче код
- Това се случва с помощта на Интерфейси

```
interface Callback {  
    void onResult(int result);  
}  
  
public class Calculator {  
    public void factorial(  
        int factorial,  
        Callback callback) {  
        ...  
        callback.onResult(result);  
    }  
}
```

```
class Program {  
    int factorial = 6;  
    public void doSomething() {  
        Calculator calc = new Calculator();  
        calc.factorial(factorial, new Callback() {  
            @Override  
            public void onResult(int result) {  
                println("Factorial of " + factorial + " is: "  
                    + result);  
            }  
        })  
    }  
}
```

Почивка

ДО ЖЖ:ЖЖ



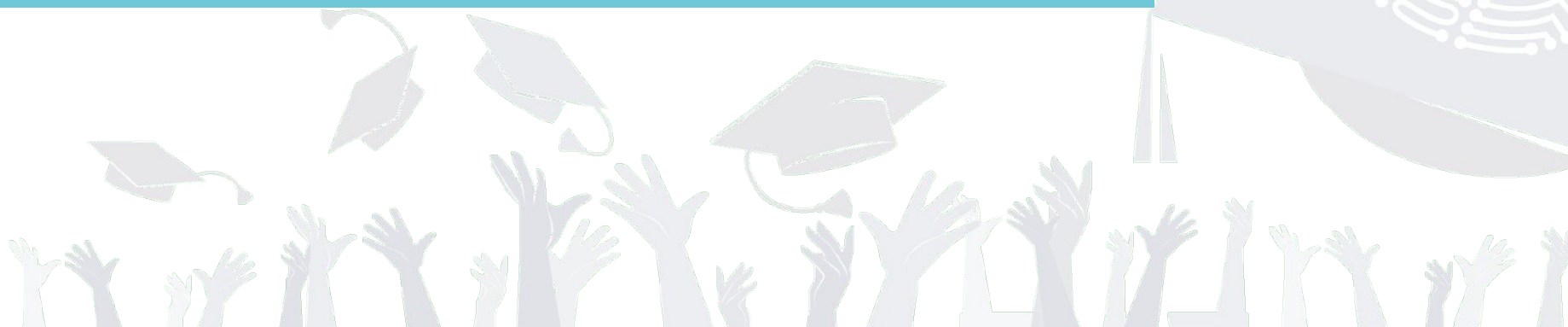
ВЪПРОСИ?



Резюме



- Поука 1
- Поука 2



Задачи за упражняване



Задача 1

Детектор за плагиатство

Условието е тук:

<https://github.com/fmi/java-course/blob/ab5f6a55fce87696aca7f4edc55d30b60e92b71a/homeworks/02-authors-ship-detection/README.md>

Не е задължително да спазвате класовете и интерфейсите дефинирани в условието.



Ами сега?

Задача за упражнение

© 2020 Нет Ит

БЛАГОДАРЯ ЗА ВНИМАНИЕТО!



SOFTWARE
ACADEMY

