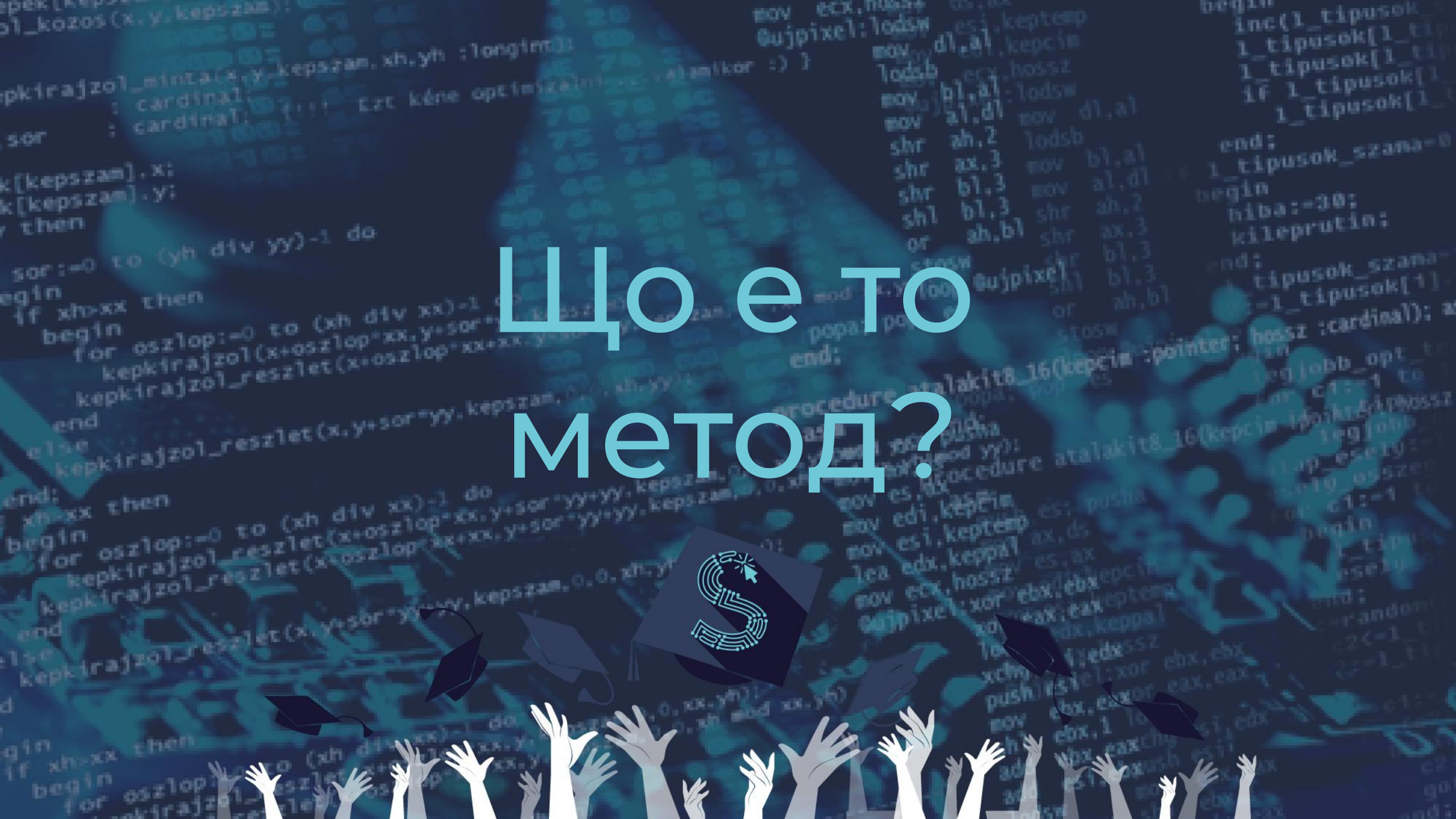
ЛЕКЦИЯ I

JAVA 101: Методи

Теодор Костадинов

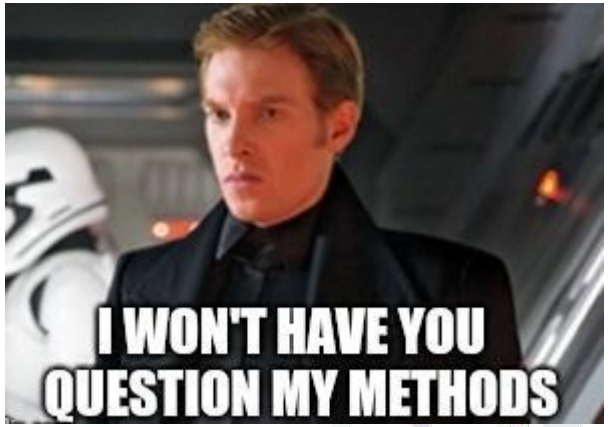
Съдържание

Що е то метод?



Методите са

- Методите са именувана последователност от действия. Те ни позволяват да разделим програмата на подпрограми, всяка една от които да решава само един отделен проблем/задача.
- В някои езици са известни като функции.
- Методите се създават само в рамките на даден клас и извън друг метод.
- Методите ни предоставят:
  - Логическо разделение на кода
  - Малки парчета, които вършат ЕДНО конкретно нещо
  - По този начин се оптимизира кода
  - Не се повтаря вече използвана логика
  - Ясно и лесно се структурира
  - Пишем по-малко

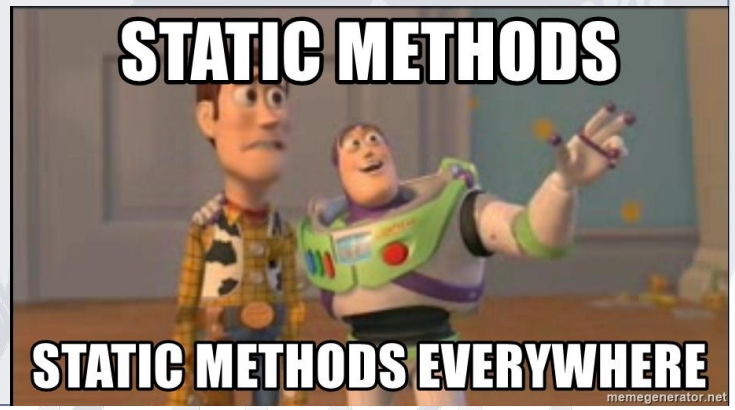


Пример

SOFTWARE ACADEMY
Educational Center Net It

```
class HelloWorld {
    public static void main( String []args ) {
        System.out.println("Hello World!");
    }
}

class Greet{
    public static void sayHello() {
      System.out.println("Hello");
    }
}
```



Методите са хубави защото

- Създаването на един метод предоставя възможността да именуваме последователност от действия, което от своя страна прави кода по-лесно четим и разбираем.
- Въвеждането на нови методи прави програмата по-малка (в повечето случаи), елиминирайки повтарящия се код.
- Една от най-популярните техники за решаването на даден проблем е разбиването му на по-малки подпроблеми. Методите позволяват да се съсредоточиш върху един подпроблем самостоятелно и после да интегрираш по-малките решения в цялостно такова.

Какво е main метод?

- Main методът е специален метод за всяка една Java програма.
- Java компилаторът не разбира нищо освен него. В момента, когато той се достигне, кодът на програмата ни се изпълнява.
- Ако той не съществува, кодът никога няма да се изпълни.

```
public static void sayHello(){
   System.out.println(“Hello”);
}
public static void main( String []args ){
   sayHello(); //actual execution
}
```

SOFTWARE ACADEMY
Educational Center Net It

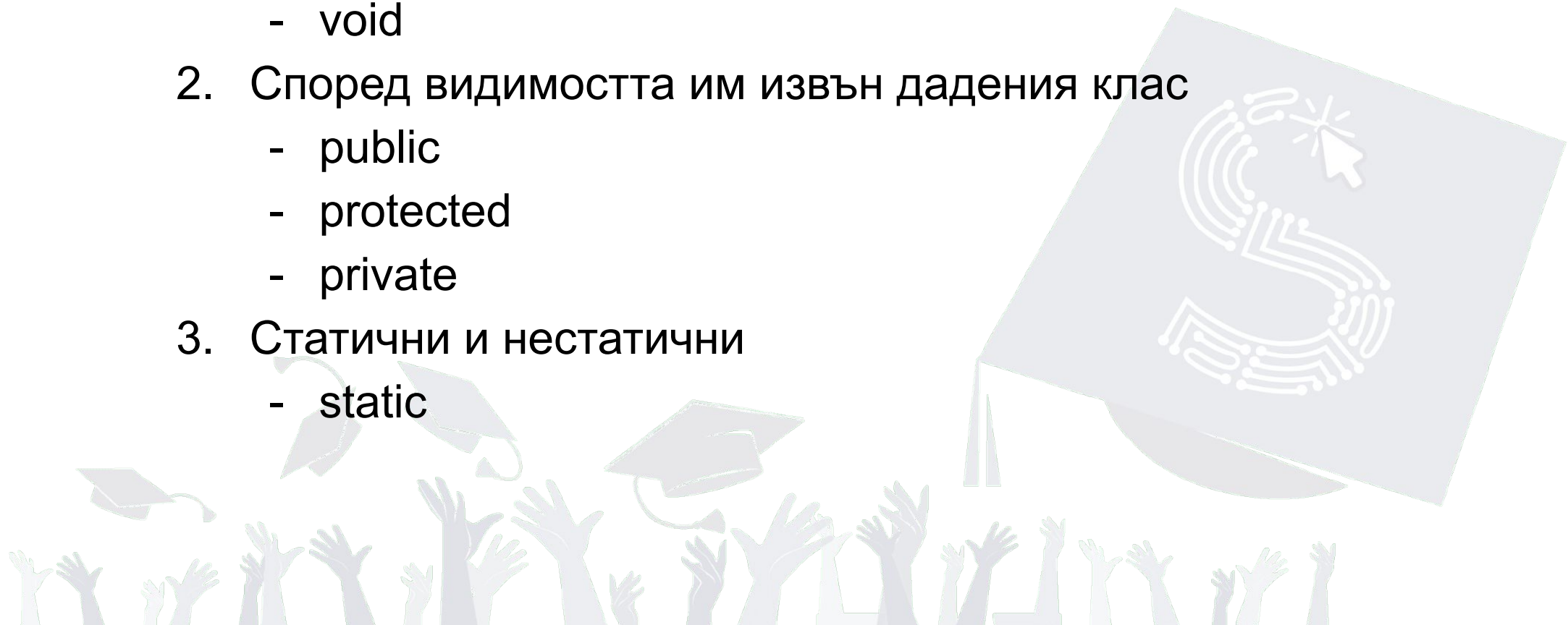
Напишете метод, който принтира в конзолата комплимент от вас към вас.

Ами сега?

Задача за упражнение

Видове методи

1. Според вида на върнатата стойност
   - int, boolean etc.
   - void
2. Според видимостта им извън дадения клас
   - public
   - protected
   - private
3. Статични и нестатични
   - static



Void Методи

Void метод е такъв, който просто извършва действие, прави дадена обработка.

```
public static void complimentMyself(){
   System.out.println(“I am a good programmer”);
}

public static void lockDoor(){
   System.out.println(“Door is now locked”);
}

public static void main( String []args ){
   complimentMyself();
}
```

Метод, който връща стойност

Досега разглежданите методи правеха нещо конкретно, но в повечето случаи ни трябва резултатът от тяхното изпълнение. Това става с ключовата дума return.

Връщането на стойност е различно от принтирането в конзолата.

```
public static int addOne(int number){
   return number + 1;
}

public static void main( String []args ){
   int a = addOne(1); //a = 2
}
```

SOFTWARE ACADEMY
Educational Center Net It

Напишете метод, който приема цяло число и връща същото число на втора степен

Ами сега?

Задача за упражнение

Как се използват методите?

SOFTWARE ACADEMY
Educational Center Net It

1. Деклариране на метода

```
public static void bark()
```

2. Имплементация (логиката на нашия метод)

```
System.out.println("woof, woof");
```

3. Извикване с цел решаване на даден проблем

```
bark();
```

Почивка

до 19:50

Деклариране

Състои се от следните три **задължителни** неща:

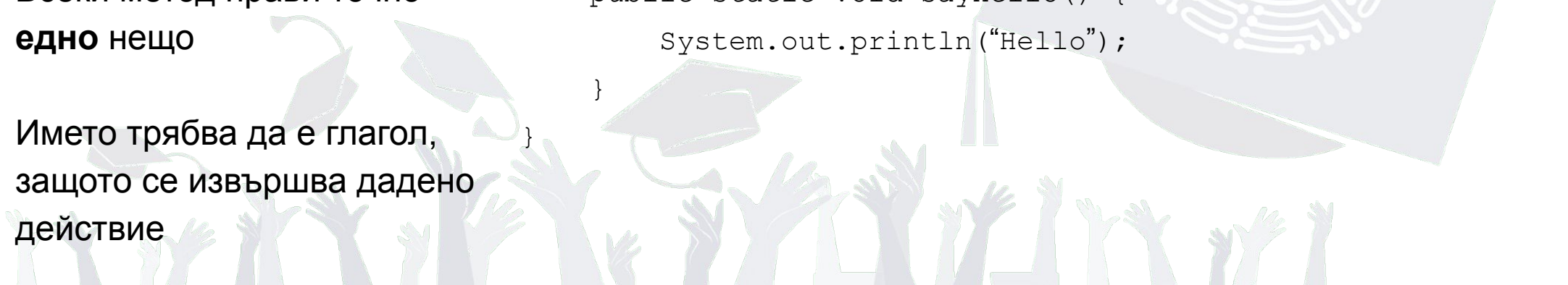
<тип> <име> (<параметър1>, <параметър2>, …, <параметърN>) { }

1. <тип> е типът на върнатата стойност, независимо дали е от примитивен тип (int, boolean, double, char), референтен (String, масиви, custom типове) или не връща нищо (void)
2. <име> е името на метода
3. <параметър1>, <параметър2>, …, <параметърN> са определени променливи, върху които трябва да извършим дадена последователност от действия, за да решим задачата. Могат да бъдат една, няколко или николко

Деклариране

1. Методите се именуват по така наречената camelCase конвенция
2. Имената трябва да са описателни и да говорят сами за себе си
3. Всеки метод прави точно **едно** нещо
4. Името трябва да е глагол, защото се извършва дадено действие

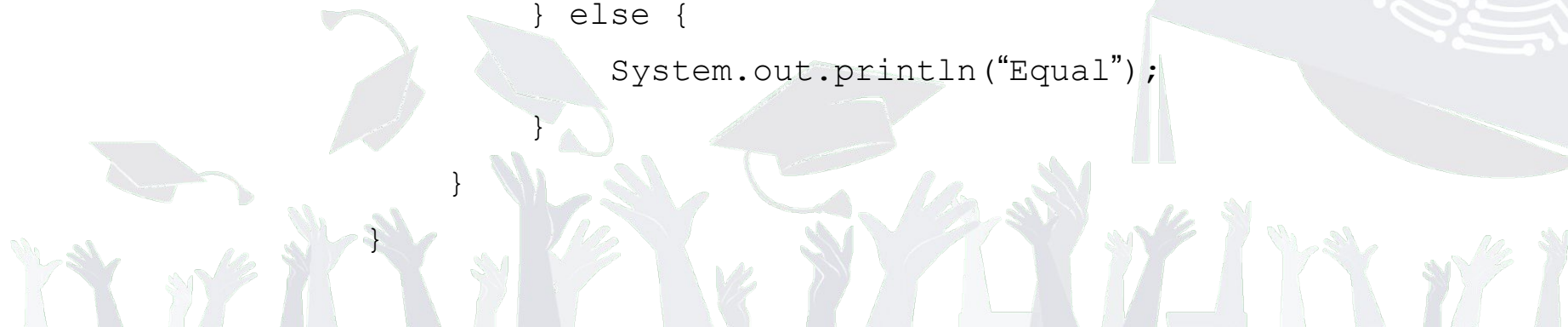
```
class ExampleClass {
    public static void printBigger(int a, int b) {
        if(a > b) {
            System.out.println(a);
        } else {
            System.out.println(b);
        }
    }
    public static void sayHello() {
        System.out.println("Hello");
    }
}
```



Имплементиране

Пространството между къдравите скоби се нарича “**тяло**” на метода и там се описва логиката, която той извършва:

```
class ExampleClass {
    public static void findMax(int a,  int b) {
        if (a > b) {
            System.out.println(a);
        } else if (b > a) {
            System.out.println(b);
        } else {
            System.out.println(“Equal”);
        }
    }
}
```



SOFTWARE ACADEMY
Educational Center Net It

Напишете метод, който по две дадени положителни числа проверява дали последната им цифра е една и съща

Ами сега?

Задача за упражнение

ВЪПРОСИ?

Параметри vs аргументи

- Параметър е променлива в декларацията на метода:

```
public static void doSmth(int myParameter) { //код }
```

- Аргументи са данните, които подаваме на метода при неговото извикване:

```
doSmth(0);
```

```
<modifier> static <returnType> <nameOfMethod> (<parameterList>){
    //method body
}
```

```
public static void methodWithParams(int param){
    printHelloFromMyFirstMethod()
    System.out.println("This method\'s parameter is: " + param);
}
```

```
public static int addOne(int number){
    return number + 1;
}
```

Напишете метод, който по две дадени числа проверява дали същите на втора степен имат една и съща последна цифра

Hint: Използвайте вече написаните преди малко методи

Scope

Отнася най-вече до видимостта на променливите. В зависимост от това къде са декларирани, те могат или не могат да бъдат достъпвани от конкретно място в нашата програма. Нека разгледаме следния пример:

```
class ExampleClass {
    public static void swap(int a,  int b){
        int c = a; //visible only here
        a = b;
        b = c;
        System.out.println(a + " " + b);
    }
    public static void main( String []args ){
        swap();
        System.out.println(c); //not visible
    }
}
```

Методи с едно и също име (Overloading)

- Използване на един идентификатор за дефиниране на множество методи, които се различават по своите входни параметри.

- Използва се, когато няколко метода правят почти едно и също нещо, но с различен брой параметри

```
public static boolean areEqual(int a, int b){
    if(a == b){
        return true;
    }
    return false;
}

public static boolean areEqual(int a, int b, int c){
    if(a == b){
        if(b == c){
            return true;
        }
    }
    return false;
}
```

Ами сега? Как можем да рефакторираме кода?

Задача за упражнение

SOFTWARE ACADEMY
Educational Center Net It

Напишете два метода с едно и също име, които връщат кое от съответно две или три цели числа е най-близо до 100. И трите подадени числа приемаме, че ще бъдат различно далече от 100.

Ами сега?

Задача за упражнение

ВЪПРОСИ?

Резюме

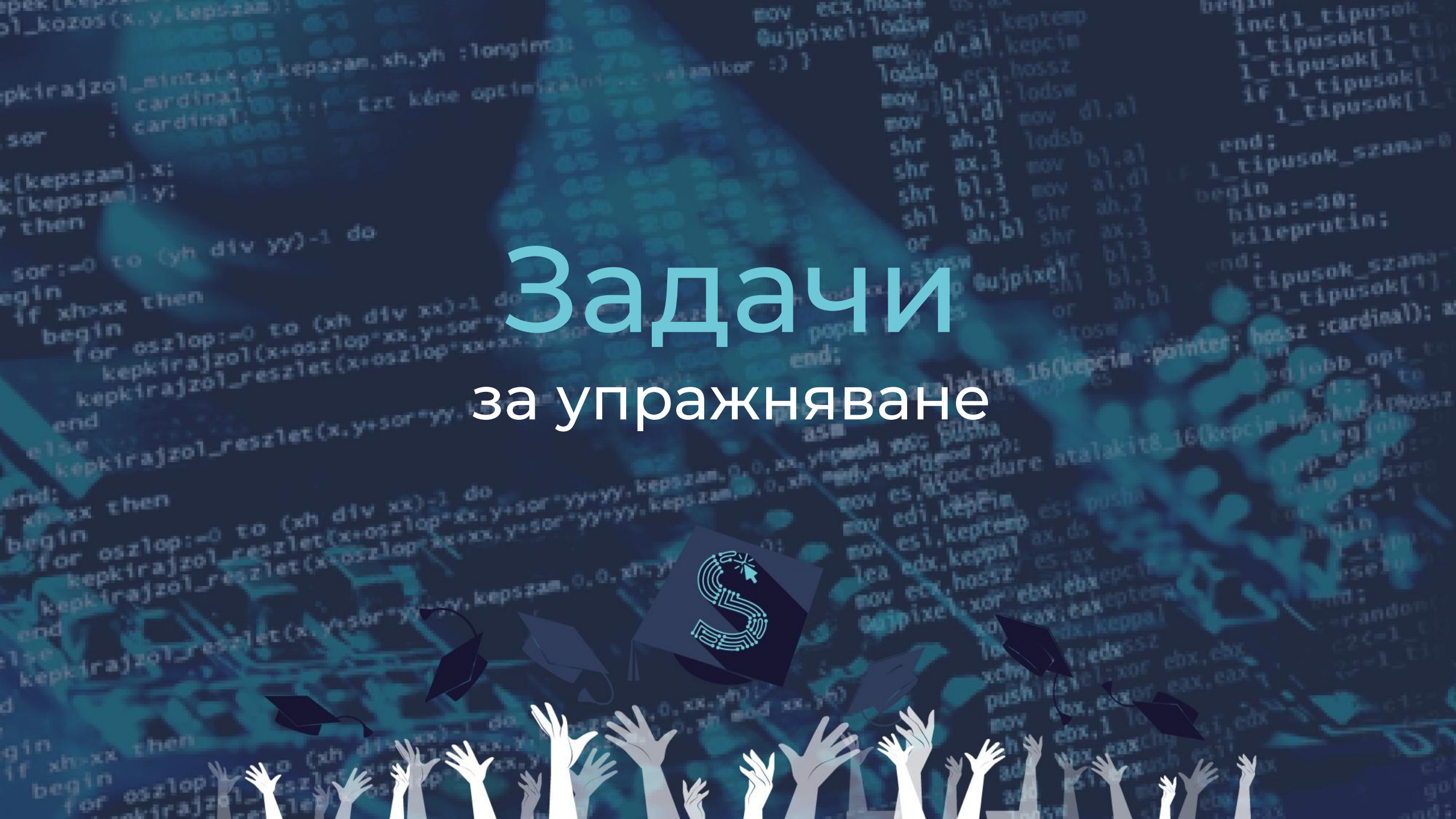
- Всеки метод прави само едно нещо
- Именуването става с помощта на практиката camelCase
- Името на всеки метод започва с малка буква и е глагол
- Името обозначава какво точно прави метода, макар понякога името да е дълго
- Не е добра практика за се използва and и or при именуването, защото това е показател, че методът прави повече от едно нещо

Ресурси

- https://codingbat.com/java/Logic-1
- Docs

Задачи

за упражняване



Задача 1 и 2

SOFTWARE ACADEMY
Educational Center Net It

1. Да се изведе на конзолата по-голямото от две числа.
Бонус: Ако разликата между двете числа е повече от 10, тя също да се изведе на конзолата (друг метод)

2. Да се напише метод, който да проверява дали едно число е четно.
Бонус: да проверява дали числото се дели на 5, 7 и 13 (друг метод)

А как може да се рефакторира кода с цел по-малко писане?

Ами сега?

Задача за упражнение

Задача 3 и 4

SOFTWARE ACADEMY
Educational Center Net It

3. Да се напишат два метода, като единият конвертира температура от Целзий към Фаренхайт, а другият- обратното, по формулите:

F = C * 1.8 + 32
C = (F - 32) / 1.8

4. Да се напише метод, който проверява дали две числа са равни.

Напишете метод, който проверява дали три числа са равни.

А как може да се рефакторира кода с цел по-малко писане?

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

SOFTWARE ACADEMY
Educational Center Net It

1. Да се напише метод, който приема число и определя дали то е положително (ще считаме 0 за положително число)

Задача 2

SOFTWARE ACADEMY
Educational Center Net It

2. Да се напише метод, който приема цяло число между 1 и 7 и принтира съответния ден от седмицата

Задача 3

SOFTWARE ACADEMY
Educational Center Net It

3. Да се напише метод, който приема три числа и определя дали първото се дели на второто и третото

Задача 4

SOFTWARE ACADEMY
Educational Center Net It

4. Да се напише метод, който проверява дали една година е високосна или не

Задача 5

SOFTWARE ACADEMY
Educational Center Net It

5. Да се напише метод, който проверява дали един символ е гласна буква
Бонус: Да не се отчита дали е главна или малка буква

БЛАГОДАРЯ
ЗА ВНИМАНИЕТО!