

ЛЕКЦИЯ I © 2020 Нет Ит

JAVA OOP: Полиморфизъм

Теодор Костадинов



SOFTWARE
ACADEMY



Съдържание

1. Полиморфизъм
2. Overriding
3. Overloading



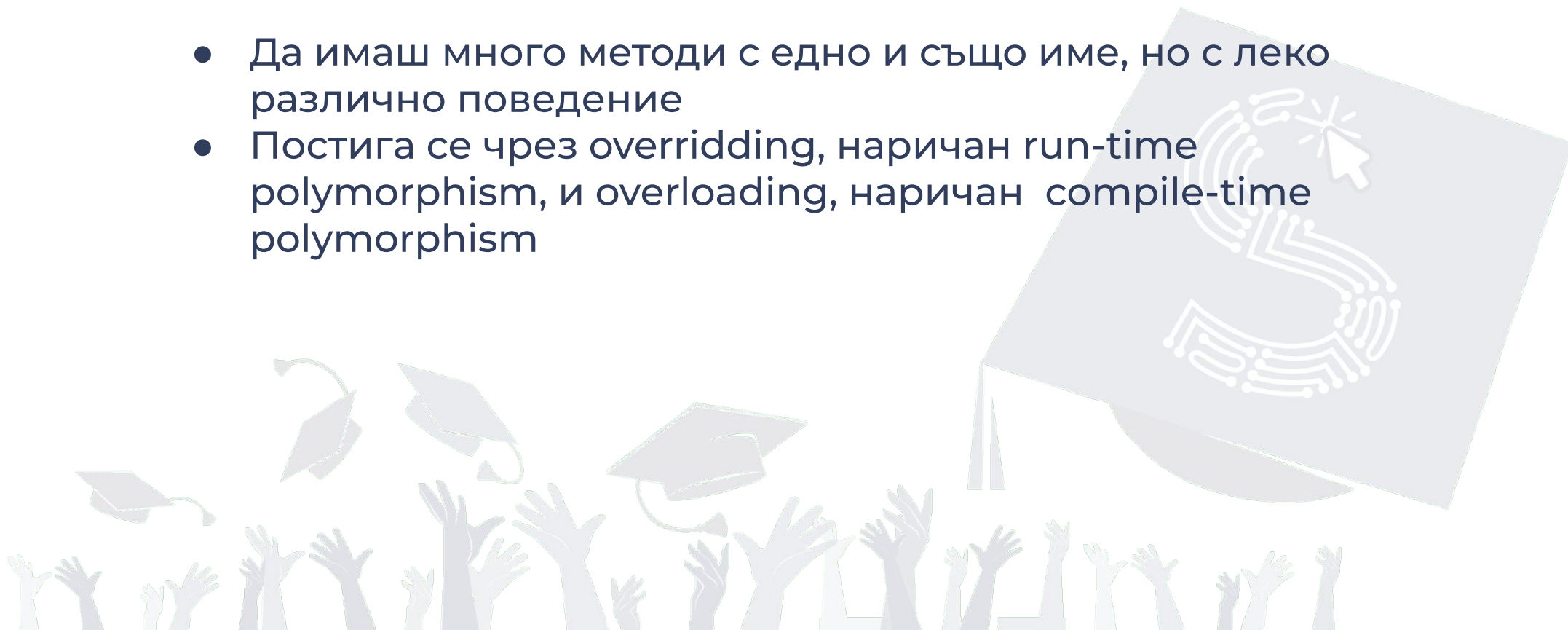
Принципи на ООП

- **Наследяване**
- **Полиморфизъм**
- Абстракция
- Капсулация



Полиморфизъм

- Едно име, много форми
- Да имаш много методи с едно и също име, но с леко различно поведение
- Постига се чрез overriding, наричан run-time polymorphism, и overloading, наричан compile-time polymorphism



Overriding



Overriding

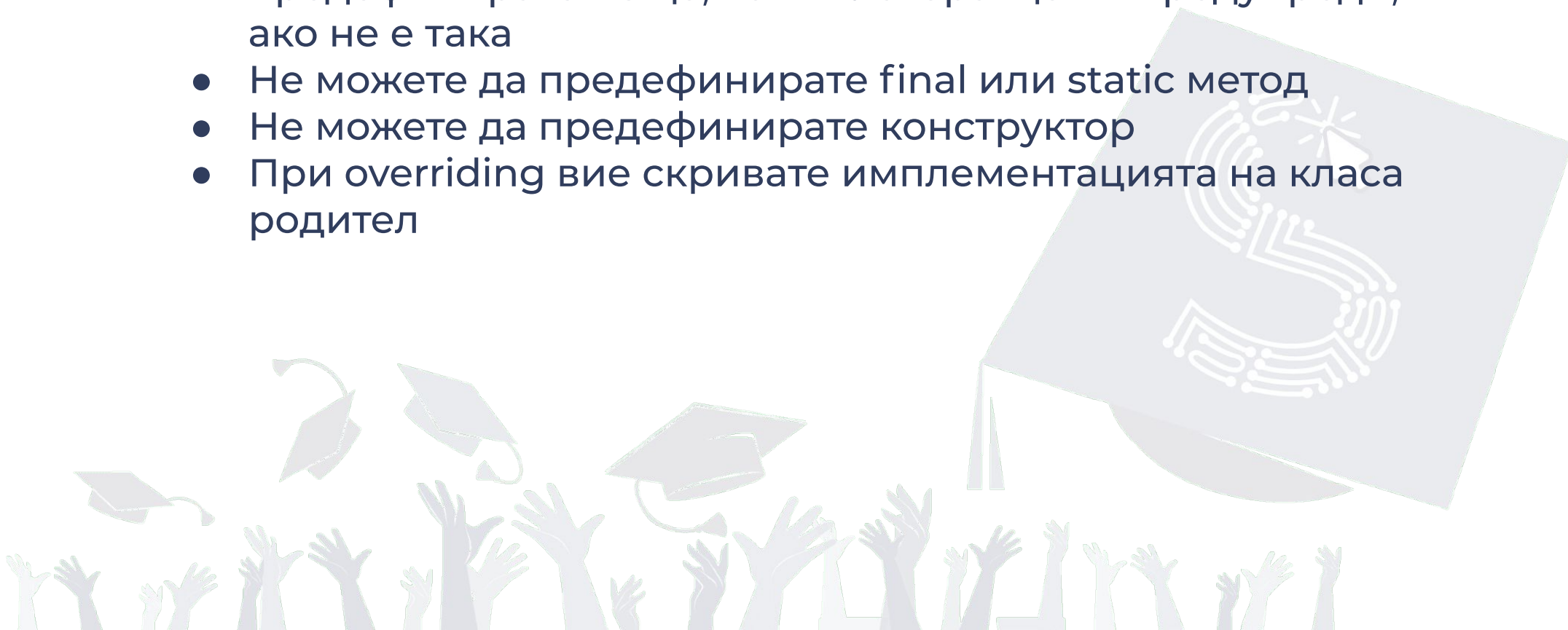
- Възможен е, когато един клас наследява друг.
- Когато класа-дете дефинира метод със същата сигнатура като класа-родител, но с различна имплементация, това се нарича overriding

```
class Human {  
    public void eat() {  
        sysout("eating loudly");  
    }  
}
```

```
class Lady extends Human {  
    @Override  
    public void eat() {  
        sysout("eating politely");  
    }  
}
```

Overriding

- @Override не е задължително, но така сте сигурни, че предефинирате нещо, компилатора ще ви предупреди, ако не е така
- Не можете да предефинирате final или static метод
- Не можете да предефинирате конструктор
- При overriding вие скривате имплементацията на класа родител



Hiding Fields

- Можете да дефинирате поле със същото име в класа-дете, това ще скрие това поле от класа-родител
- Може да дефинирате и статични полета и методи в класа-дете със същите имена, това ги крие от класа-родител
- Не е добра практика да се крият полета и методи.



Overloading



Overloading

- Този термин се използва, когато съществуват два метода с едно и също име в един клас. Компилятора при създаване на вашата програма избира кой да използва спрямо вида и броя на параметрите.

```
class Human {  
    public void eat() {  
        sysout("eating loudly");  
    }  
  
    public void eat(int n) {  
        sysout("eating loudly for " + n + " hours");  
    }  
}
```

```
class Test {  
    ..main..  
    Human jeff = new Human();  
    jeff.eat();  
    jeff.eat(6);  
}
```

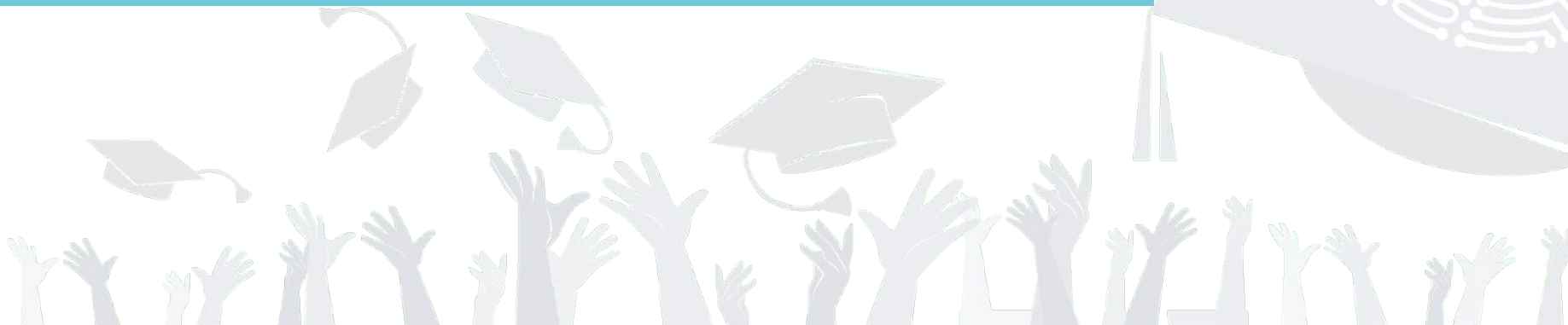

ВЪПРОСИ?



Резюме



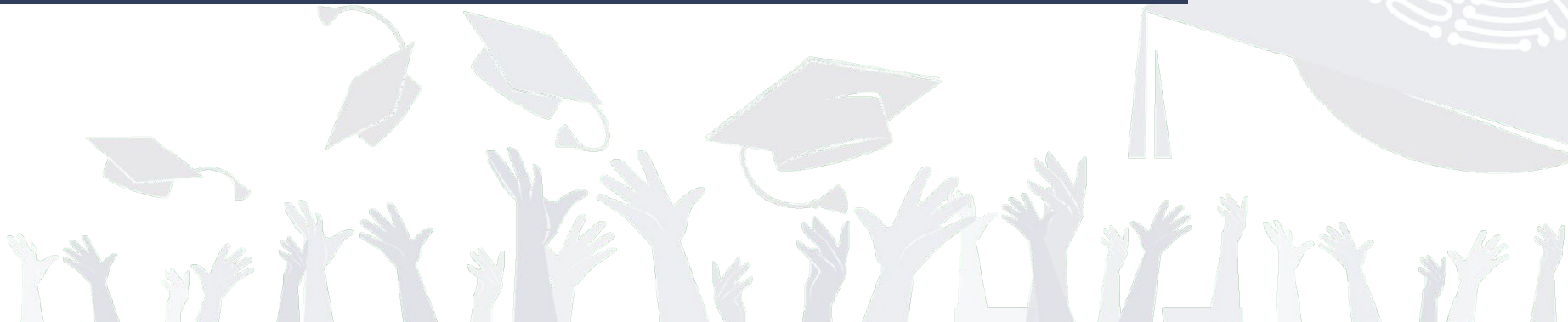
- Полиморфизмът ни казва, че можем да извикваме един и същ метод на различни класове и той може да има различна имплементация.
- Трябва да внимаваме да не скрием имплементация, която не искаме.



Ресурси



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



Задачи за упражняване



Задача 1

Дефинирайте клас `PlayingPiece`

Да има полета `x` & `y`

Да има поле `isAlive`

Да има метод `move(int newX, int newY)`

Да има метод `moveIsLegal(int newX, int newY)`

За офицер:

Наследете клас `PlayingPiece`

Предефинирайте метод `moveIsLegal (int newX, int newY)`



Ами сега?

Задача за упражнение

Задача 2

Дефинирайте клас Human със свойства "собствено име" и "фамилно име". Дефинирайте клас Student, наследяващ Human, който има свойство "оценка". Дефинирайте клас Worker, наследяващ Human, със свойства "надница" и "изработени часове".

Имплементирайте и метод "изчисли надница за 1 час", който смята колко получава работникът за 1 час работа, на базата на надницата и изработените часове. Напишете съответните конструктори и методи за достъп до полетата (свойства).



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



Напишете класове, които представят работа с библиотека. Библиотеката има име, адрес и работно време (текстови полета).

Библиотеката има и списък с издания, които могат да бъдат наемани.

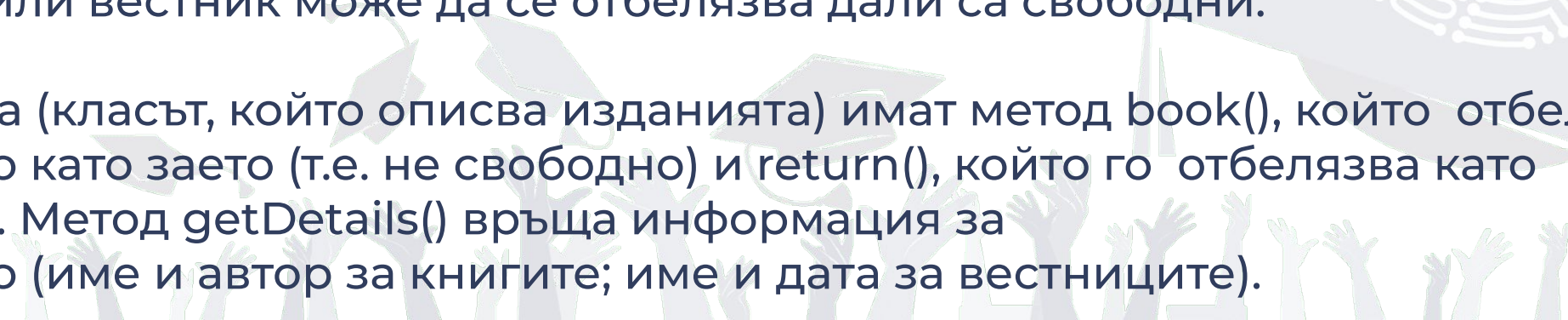
Изданията може да бъдат книги или вестници.

Книгите имат име, автор, година на издаване.

Вестниците имат име и дата на издаване.

За книга или вестник може да се отбелязва дали са свободни.

Изданията (класът, който описва изданията) имат метод `book()`, който отбелязва изданието като заето (т.е. не свободно) и `return()`, който го отбелязва като свободно. Метод `getDetails()` връща информация за изданието (име и автор за книгите; име и дата за вестниците).



Задача 1



1. Напишете класове за: библиотека, издания, книга, вестник. Нека тези класове имат връзки помежду си, атрибути и методи, както е описано в условието.
2. Напишете метод на класа Библиотека, който да приема като параметър обект издание. В метода се обхожда списъка с издания на библиотеката и ако описанието на подаденото издание съвпада с някое от списъка, да отбелязва това в списъка като заето.
3. Направете клас TestLibrary с main метод, в който създайте няколко книги и вестници. Създайте и една библиотека и ѝ добавете създадените издания. Изведете в конзолата списък с описанията на всички издания от създадената библиотека. Ако сте направили подточка 2), тествайте метода от нея.



© 2020 Нет Ит

БЛАГОДАРЯ ЗА ВНИМАНИЕТО!



SOFTWARE
ACADEMY

