



Увод

в обектно ориентираното програмиране



ООП – Упражнения

Съдържание

- Преговор
- Задачи

Какво е ООП?

Какво е то?

- Основна единица на обектно ориентираното програмиране е класа.
- ООП започва от идентифицирането на класовете и преминава в имплементиране на методите в тях.
- Класът дефинира променливите и методите, които обектите ще имат.
- Обект е представител на клас. Всеки обект е от даден клас, който му дефинира свойствата и възможностите.

Защо го използваме?

- Добро структуриране на кода
- Намалява сложността на кода
- Позволява преизползване на кода
- Може да се постигне абстрактност

Основни принципи

Основни принципи

- Капсулация: знае се КАКВО може да прави компонента, а не как
- Наследяване: един обект със същите свойства като друг може да го наследи
- Полиморфизъм: позволява унифицирано извършване на действия над различни обекти
- Абстракция: Създаването на класове, обекти и типове по техните интерфейси и функционалност вместо по имплементационните им детайли

Overriding

Когато един клас наследява друг:

Наследника може да промени поведението на някои от методите на базовия клас. Той ги предефинира.

@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

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.

Клас

Клас

- Класът е шаблон от който създаваме обект
- Обекта е конкретен елемент от даден клас. Нарича се още инстанция на класа

Какво има в един клас

Какво има в един клас

- Полета:
 - String name;
 - int age;
- Методи:
 - void study(){...}
 - void doHomework(){...}
- Конструктор
- Други неща

Конструктор

- Конструктора е метод, който се извиква автоматично при създаването на обект от класа и само тогава
- Използва се за да се зададат първоначални стойности на полетата и за да се направят първоначални настройки на класа
- Един клас може да има много конструктори

Нива на достъп

Нива на достъп

Modifier	Class	Package	Subclass	World
public	✓	✓	✓	✓
protected	✓	✓	✓	✗
no modifier	✓	✓	✗	✗
private	✓	✗	✗	✗

Polymorphism

Polymorphism

- one name, many forms
- Да имаш много методи с едно и също име, но с еко различно поведение
- Постига се чрез `overriding`, наричан `run-time polymorphism`, и `overloading`, наричан `compile-time polymorphism`

Наследяване

Наследяване

- Една от най-силните черти на наследството е възможността за extend-ване на компоненти без да се знае нищо за начина, по който са имплементирани в базовия клас
- Обектите могат да бъдат свързани помежду си чрез връзка от типа “има”, „използва“ и „е“ . Именно „е“ връзката е начина на наследяване на един обект от друг. (Когато можем да кажем, че един обект е от типа друг обект)



Задача 1

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).

Задача 2

Инициализирайте масив от 10 студента и ги сортирайте по оценка в нарастващ ред. Използвайте Java интерфейса **java.lang.Comparable**.

Задача 3 (1/4)

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.

Задача 3 (2/4)

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 3 (3/4)

- a) Напишете класове за: библиотека, издания, книга, вестник. Нека тези класове имат връзки помежду си, атрибути и методи, както е описано в условието.
- b) 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.

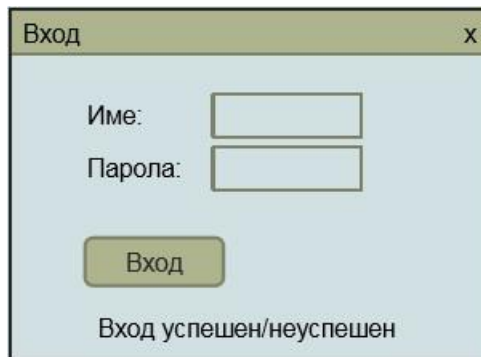
Задача 3 (4/4)

с) 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 b), тествайте метода от нея.

Домашно

Задача

Направете форма за вход на потребител със Swing. Нека формата да има следния вид:



The image shows a Java Swing window titled "Вход" (Login) with a close button "x" in the top right corner. The window has a light blue background and contains the following elements:

- A label "Име:" followed by a text input field.
- A label "Парола:" followed by a text input field.
- A button labeled "Вход" (Login).
- A status label "Вход успешен/неуспешен" (Login successful/unsuccessful) at the bottom.

Задача

При натискане на бутона “Вход”:

- Ако въведените име и парола са равни съответно “admin” и „1234“, да изписва в текстовото поле „Вход успешен“.
- Ако са различни, да изписва „Вход неуспешен“.

(По-важно е да направите да работи както е указано, отколкото елементите да са подредени както на картинката.)

Задача

Имаме превозни средства автомобили и мотори, които имат:

- година на производство
- марка и модел
- изминати километри
- движи се

Да се напише програма, която

- въвежда превозни средства от разл. тип (автомобил, мотор)
- извежда списък с марка, модел и изминати километри на всички превозни средства

*Задача

По избор прочетете това:

- <http://www.introprogramming.info/intro-java-book/read-online/glava16-lineini-strukturi-ot-danni/>