Увод

в обектно ориентираното програмиране

ООП – Принципи

Съдържание

- Капсулация
- Абстракция
- Наследяване
- Полиморфизъм
- Интерфейси

ООП

ООП

- 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.

Overriding

Когато един клас наследява друг:

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

```
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

Overloading

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

```
class Human {
   public void eat() {
   sysout(“eating louduly”);
   }
   public void eat(int n) {
   sysout(“eating louduly for ” + n + “ hours”);
   }
}
```

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

Polymorphism

Polymorphism

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

Задача - шахматна фигура

- 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)

Задача - офицер

- Наследете клас PlayingPiece
- Предефинирайте метод moveIsLegal (int newX, int newY)

Задача - пешка

- Наследете клас PlayingPiece
- Предефинирайте метод moveIsLegal (int newX, int newY)
- Ако пешката не е местена от първоначално зададената си позиция тя има правото да се премести с две напред, иначе се мести с едно напред

Капсулация

Капсулация

- Защита на данните и имплементацията
- Скриването на имплементацията на данните чрез ограничаване на достъпа до мутаторите
- Можем да правим промени на обекта без да се тревожим, че ще счупим другия код, който извиква методите от класа за информация

Интерфейси

Интерфейси

Дефинират списък от операции, методи, без да дефинират самите тях

Нещо като обещание, че един клас ще има дадени методи

Дефинират абстрактни типове данни

```
class Dog implements IBarkable {
   public void bark() {
   sysout(“bau”);
   }
}
```

```
interface IBarkable{
void bark();
}
```

Абстракция

Абстракция

- Създаването на класове, обекти и типове по техните интерфейси и функционалност вместо по имплементационните им детайли
- Възможността да взаимодействаш не само с конкретен клас, а с всички класове правещи дадено нещо

Задача

- Създайте клас Куче с няколко полета и методи
- Имплементирайте интерфейса Comparable за класа Куче

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 имплеметирани 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 обек е от типа друг обект)

Пример:
Клас човек;
Класа ученик е човек;

Един клас има

- Полета: променливи, които определят настоящия статус на обекта
- Статични полета: променливи, които са общи за всички обекти от класа и по-скоро определят статуса на класа като цяло, не на отделните обекти от този клас.
- Методи: изпълним код, позволяващ ни да променяме състоянието на обекта или да достъпваме данни от него
- Статични методи: изпълним код, отнасящ се за класа като цяло, не трябва да използва в себе си променливи, които не са статични
- Вложени класове и интерфейси

Статичност

Статичност

Дадено поле или метод се асоциират с класа, а не с обект от него

Съществува само едно копие от тази променлива и то се използва от всички обекти от класа.

Тези атрибути се извикват с името на класа

Инициализират се при първото използване на този клас

Задача

● Нека при създаването на обект куче от класа Куче - на всяко новосъздадено куче да се задава уникален пореден номер

Задача

Създайте йерархия Dog, Frog, Cat, Kitten, Tomcat и дефинирайте съответните конструктори и методи за всеки клас. Кучетата, жабите и котките са Animals. Всички животни могат да издават звуци (уточнени чрез интерфейса ISound). Kittens и tomcats са котки. Всички животни имат години, име и пол. Kittens могат да бъдат само женски, а tomcats – мъжки. Всяко животно издава специфичен звук. *Създайте масиви от различни видове животни и пресметнете средната възраст на всеки вид животно използвайки статичен метод.

Домашно

Задача

Видеотека:

Създайте система за вземане на видеокасети от видеотека

Да може да се създава акаунт на потребителите, които взимат касети

Да се знае кой потребител кои касети, кога е взел, кога трябва да ги върне

Да може да се вземе/върне касета

Да може да се извежда списък с всички невърнати касети и акаунтите, които са ги взели

Задача

- Създайте класове за всички шахматни фигури и имплементирайте проверката за правилен ход на всички

- Добавете в класа PlayingPiece поле цвят

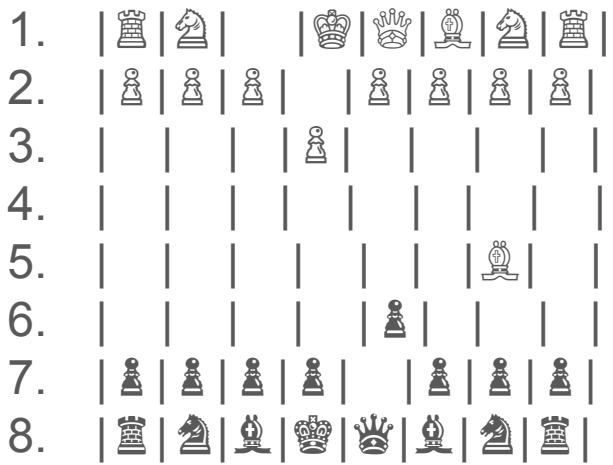
- Създайте клас дъска, който при създаването си да инициализира двумерен масив 8x8 и в него да слага фигурите

- Класа дъска да има метод play(int color, int fromX, int fromY, int toX, int toY) - да не разрешава да се мести ако два пъти по ред се играе един и същ цвят

- Класа дъска да има проверка дали на дъската има шах или шахмат, тя да се извиква след всеки ход

- След всеки ход дъската да се чертае на конзолата

Пример



Тук са кодовете на символите

http://www.utf8-chartable.de/unicode-utf8-table.pl? start=9728&number=128

Задача

Прочетете това:

- http://www.introprogramming.info/wp-content/uploads/2014/04/Introduction-to-Programing-with-Java-Book-v2014.pdf

И това:

- http://www.introprogramming.info/intro-java-book/readonline/glava14-definirane-na-klasove/