**TEMA**: CLASE/METODE ABSTRACTE . INTERFETE

25.05.2021

* CLASE ABSTRACTE

Clasa abstract are restrictive pt a crea obiecte.Pentru a putea face asta,ele trebuie mostenite intr-o clasa non-abstracta

Abstracția datelor este procesul de ascundere a anumitor detalii și de afișare a utilizatorului numai a informațiilor esențiale.

Abstracția poate fi realizată fie cu clase abstracte, fie cu interfețe

Cuvântul cheie **abstract** este un modificator fără acces, utilizat pentru clase și metode:

**Clasa abstractă**: este o clasă restricționată care nu poate fi utilizată pentru a crea obiecte (pentru a accesa, trebuie să fie moștenită de la o altă clasă).

**Metoda abstractă**: poate fi utilizată doar într-o clasă abstractă și nu are un corp. Corpul este creat de subclasă (moștenită de la).

O clasă abstractă poate avea atât metode abstracte, cât și metode regulate:

abstract class Animal {

public abstract void animalSound();

public void sleep() {

System.out.println("Zzz");

}

}

Pentru a accesa clasa abstractă, aceasta trebuie moștenită de la o altă clasă. Să convertim clasa Animal pe care am folosit-o în capitolul Polimorfism într-o clasă abstractă:

// Abstract class

abstract class Animal {

// Abstract method (does not have a body)

public abstract void animalSound();

// Regular method

public void sleep() {

System.out.println("Zzz");

}

}

// Subclass (inherit from Animal)

class Pig extends Animal {

public void animalSound() {

// The body of animalSound() is provided here

System.out.println("The pig says: wee wee");

}

}

class Main {

public static void main(String[] args) {

Pig myPig = new Pig(); // Create a Pig object

myPig.animalSound();

myPig.sleep();

}

}

* INTERFETE

Un alt mod de a realiza abstractizarea în Java, este cu interfețele.

O interfață este o „clasă abstractă” complet utilizată pentru a grupa metode conexe cu corpuri goale:

// interface

interface Animal {

public void animalSound(); // interface method (does not have a body)

public void run(); // interface method (does not have a body)

}

Pentru a accesa metodele de interfață, interfața trebuie să fie „implementată” (cam ca moștenită) de o altă clasă cu **cuvântul cheie** **implements** (în loc de **extends**).

Putem implementa mai multe interfete…separandu-le prin virgula ( !!! la mostenire nu putem mosteni mai mult de 1 clasa)

interface FirstInterface {

public void myMethod(); // interface method

}

interface SecondInterface {

public void myOtherMethod(); // interface method

}

// IMPLEMENTAM MAI MULTE INTERFETE

class DemoClass implements FirstInterface, SecondInterface {

public void myMethod() {

System.out.println("Some text..");

}

public void myOtherMethod() {

System.out.println("Some other text...");

}

}

class Main {

public static void main(String[] args) {

DemoClass myObj = new DemoClass();

myObj.myMethod();

myObj.myOtherMethod();

}

}