Lab 1

-am facut ceva chestii de baza, ne-am acomodat cu java, m-am acomodat cu sintaxa, interfata si cu workload ul, pentru inceput e foarte asemanator cu cpp ;

In loc de # include avem import

Clasa principala e calsa care contine metoda main si e punctul de intrare al aplicatiei

Public static void main;

Lab 2:

-In java nu aem pointeri ,

Ne-am jucat cu functiile din aceeasi clasa, main, am folosit if, while, do while , for ,chestii de genu

Lab 3:

Aici deja am intrat in oop si descoperim clasele din java

In Src pe acelasi nivel cu clasa main am facut un pachet „Models” unde am incarcat clasa sablon Animal, dupa din aceasta clasa am derivat clasele pisica si caine

La inceput punem ” package+ „numa pachet” ” in toate fisierele cu clase

Dar in main import models.Animal (import pachet + numele clasei)

Public abstract class Animal; abstract keyword ->clasa speciala sablon

Public class Dog extends Animal : extends keyword ,- ne indica mostenirea

@override – indicam suprascrierea

Lab 4:

Interfata-

Se comporta ca un fisier header din cpp, e locul unde punemdoar semnatura functiilor dintro clasa;

O interfață în Java este o structură care definește un set de metode pe care o clasă care implementează interfața trebuie să le conțină. Interfețele sunt utilizate pentru a defini comportamente contractuale și pentru a facilita polimorfismul și decuplarea în cod.

In src pe acelasi nivel cu main aem iar pachetele:

Interfaces -> cu interfata

Package interfaces

Public interface iMasina

Public default void acelereaza() { }

Models -> cu clasa propruzsa ce implementeaza interfata : keyword implements

Package model;

Import Interfaces.iMasina;

Public class Masina implements iMasina{ }

In main

import Interfaces.iMasina;  
import Models.Masina;

iMasina masina = new Masina();

Lab5:

Lab6:

Lab7: teste

Cum testam fiecare functie din clasele noastre intr un proiect? Separat

Putem sa da run la aplicatie din main , sau sa dam run din test,

In clasa test avem mai mule functii , pentru a testa fiecare componenta

@Test  
void testGetName()

{  
 Haine h = new Haine("bluza","kasmir",15);  
 *assertEquals*("bluza",h.getName());  
}

Shortcur, click dreapta pe numele clasei , create test pentru clasa dorita

Nu mai folosim inteli J folosim grindlq, gen librarie, whatever

Json transfer de date mult mai rapida

Lab 4 interfata, clea de mijloc edem ce facem

Interfat video(forntend)

La ce se refera inteftata in cod(interfata backend)

Default

10 prezente

o aplicatie java si un word argumentam lab pe care le-am facut

1 punct din oficiu

1p ruleaza aplicatia-> lab 1 + word scrisoare de dragoste 4 pg

lab 5 interface

2 grafice uml arhitectura aplicatie visual paradigm darau.io schema baze de date

salvare date json txt plus serializare ???

interafata grafica

teste

**package org.example;  
  
import com.fasterxml.jackson.databind.ObjectMapper;  
import java.io.File;  
import java.io.IOException;  
import java.util.List;  
  
public class JsonUtil {  
  
// Funcția pentru a salva datele într-un fișier JSON  
public static void saveToJson(List<Student> studenti) {  
ObjectMapper mapper = new ObjectMapper();  
try {  
// Salvează lista într-un fișier JSON  
mapper.writeValue(new File("studenti.json"), studenti);  
System.out.println("Datele au fost salvate în fisierul studenti.json");  
} catch (IOException e) {  
e.printStackTrace();  
}  
}  
  
// Funcția pentru a încărca datele dintr-un fișier JSON  
public static List<Student> loadFromJson() {  
ObjectMapper mapper = new ObjectMapper();  
File file = new File("studenti.json");  
  
// Verificăm dacă fișierul există și nu este gol  
if (file.exists() && file.length() > 0) {  
try {  
// Citește datele din fișierul JSON și le deserializați într-o listă de obiecte Student  
List<Student> studenti = mapper.readValue(file, mapper.getTypeFactory().constructCollectionType(List.class, Student.class));  
return studenti;  
} catch (IOException e) {  
e.printStackTrace();  
}  
} else {  
System.out.println("Fișierul JSON este gol sau nu există.");  
}  
return null; // Dacă fișierul nu există sau este gol, returnăm null  
}  
}**

**in build grandle**

**import com.fasterxml.jackson.databind.ObjectMapper;**

**dependinte**

**dependencies parsare json**

**uml**

**si jason**

**laboratorul 8 – verificam testele , yeeeeeee**