Ministerul Educației și Cercetării al Republicii Moldova  
Colegiul Universității Tehnice a Moldovei  
Specialitatea –Programare și analiza produselor de program

RAPORT

Lucrare de laborator nr.3

Disciplina: Asistență pentru programarea orientată pe obiecte

Tema: Moștenire

A efectuat: Lesan Denis PAPP-231  
  
  
 Chişinău 2025

Clasa de baza-Rezervari

import java.io.\***;**import java.util.Objects**;**import java.util.Vector**;**public class Rezervare {  
 static String *TipRezervare* = ""**;**   
 static String *NumeClient* = ""**;** static String *DataSiOra***;** static int *NumarulDePersoane***;** static int *DurataRez***;** public static void main(String[] args) {  
  
 try (BufferedReader br = new BufferedReader(new FileReader("foo.txt"))) {  
 String line**;** while ((line = br.readLine()) != null) {   
 if (!line.isEmpty()) {  
 String[] parts = line.split("\\s+")**;**   
 *TipRezervare* = parts[**0**]**;** *NumeClient* = parts[**1**]**;** *DataSiOra* = parts[**2**]**;** *NumarulDePersoane* = Integer.*parseInt*(parts[**3**])**;** *DurataRez* = Integer.*parseInt*(parts[**4**])**;** Vector<Object> dataVector = new Vector<>()**;** dataVector.add(*TipRezervare*)**;** dataVector.add(*NumeClient*)**;** dataVector.add(*DataSiOra*)**;** dataVector.add(*NumarulDePersoane*)**;** dataVector.add(*DurataRez*)**;** for (Object obj : dataVector){  
 System.*out*.print(" "+obj+" ")**;** }  
  
 if (Objects.*equals*(*TipRezervare***,** "RezervareZilnica")) {  
 RezervareZilnica rezZil = new RezervareZilnica()**;** rezZil.rezerv\_pers\_num()**;** } else if (Objects.*equals*(*TipRezervare***,** "RezervareEvenimentSpecial")) {  
 RezervareEvSpec rezEv = new RezervareEvSpec()**;** rezEv.rezerv\_taxa()**;** } else if (Objects.*equals*(*TipRezervare***,** "RezervareGrupMare")) {  
 RezervareGrupMari rezGr = new RezervareGrupMari()**;** rezGr.rezerv\_grup()**;** } else if (Objects.*equals*(*TipRezervare***,** "RezervareVip")) {  
 RezervareVip rezer\_vip = new RezervareVip()**;** rezer\_vip.rezerv\_vip()**;** }  
  
 }  
 }  
 } catch (IOException e) {  
 e.printStackTrace()**;** }  
  
 }  
  
}

**Folosirea bibliotecilor:**

import java.io.\***;**import java.util.Objects**;**import java.util.Vector**;**

**Java.io\*** - Importam tot din Java IO-Package

**Java.util.Objects** – Bibloteca data ne permita sa lucram cu obiectele de exemplu sa folosim **Objecs.eqauls();** pentru a compara sau a egala 2 obiecte sau mai multe obiecte

**Java.util.Vector** – Aceasta bibliotică ne permite sa lucrăm cu Vectorii[1].

https://www.reddit.com/r/learnjava/comments/j364m8/what\_does\_importjavaio\_mean/

public class Rezervare {  
 static String *TipRezervare* = ""**;**   
 static String *NumeClient* = ""**;** static String *DataSiOra***;** static int *NumarulDePersoane***;** static int *DurataRez***;**

**Formarea clasei Rezervare cu propriile sale atribute:**

**TipRezervare , NumecClient, DataSiora, NumarulDePersoane, DurataRez**

**Static** – Toate obiectele din aceasta clasa au acces la aceste variabile, De exemplu Obiectul RezervareVip are acces sa modifice atributele clase Rezervare

<https://www.geeksforgeeks.org/difference-between-static-and-non-static-variables-in-java/>

<https://stackoverflow.com/questions/95419/what-are-all-the-different-ways-to-create-an-object-in-java>

try (BufferedReader br = new BufferedReader(new FileReader("Rezervari.txt"))) {  
 String line**;** while ((line = br.readLine()) != null) {   
 if (!line.isEmpty()) {  
 String[] parts = line.split("\\s+")**;**   
 *TipRezervare* = parts[**0**]**;** *NumeClient* = parts[**1**]**;** *DataSiOra* = parts[**2**]**;** *NumarulDePersoane* = Integer.*parseInt*(parts[**3**])**;** *DurataRez* = Integer.*parseInt*(parts[**4**])**;**

Acest bloc de cod ne permite citirea și impartirea file **Rezervari.txt** in variabile de tip int si String.

**Try** – Ori ce cod ce se afla in try va fi supus verificari in caz ca va gasi erori

**Catch** – In catch putem prinde ori ce tip de eroare si putem scrie (de exemplu) ca sa ne printeze eroarea

**BufferedReader** – Permite Impartirea textului in caractere , in codul urmatori se citeste fiecare caracter , iar caracterele sunt introduse intro lista **parts. NumarulDePersoane** și **DurataRez** cu ajutorul lui **Integer.parseInt** sunt transformate in variabile de tip integer.

https://docs.oracle.com/javase/8/docs/api/java/io/BufferedReader.html

https://www.w3schools.com/java/java\_try\_catch.asp

Vector<Object> dataVector = new Vector<>()**;**dataVector.add(*TipRezervare*)**;**dataVector.add(*NumeClient*)**;**dataVector.add(*DataSiOra*)**;**dataVector.add(*NumarulDePersoane*)**;**dataVector.add(*DurataRez*)**;**for (Object obj : dataVector){  
 System.*out*.print(" "+obj+" ")**;**}

In Urmatorul cod variabilie se stocheaza in Vector de tip Object, adica variabilile pot fi de or ice tip: int, String, double, etc;

Si in cicluc for toate variabilile se afiseaza pe ecran.

<https://www.geeksforgeeks.org/java-util-vector-class-java/>

if (Objects.*equals*(*TipRezervare***,** "RezervareZilnica")) {  
 RezervareZilnica rezZil = new RezervareZilnica()**;** rezZil.rezerv\_pers\_num()**;** } else if (Objects.*equals*(*TipRezervare***,** "RezervareEvenimentSpecial")) {  
 RezervareEvSpec rezEv = new RezervareEvSpec()**;** rezEv.rezerv\_taxa()**;** } else if (Objects.*equals*(*TipRezervare***,** "RezervareGrupMare")) {  
 RezervareGrupMari rezGr = new RezervareGrupMari()**;** rezGr.rezerv\_grup()**;** } else if (Objects.*equals*(*TipRezervare***,** "RezervareVip")) {  
 RezervareVip rezer\_vip = new RezervareVip()**;** rezer\_vip.rezerv\_vip()**;** }  
  
 }  
 }  
 } catch (IOException e) {  
 e.printStackTrace()**;** }  
  
 }  
  
}

In instructiunea **if** noi egalam **TipRezervare** cu variantele posibile, in caz ca aceasta variabilă corespunde cu șirul de caractere dat , el va executa formarea unui obiect din clasele derivate din clasa de bază **Rezervare.**

**Clasa derivată RezervareZilnică**

class RezervareZilnica extends Rezervare {  
 static int *pret\_Pers* = **0;** public void rezerv\_pers\_num() {  
 if (*NumarulDePersoane* == **2**) {  
 *pret\_Pers* = **50;** } else if (*NumarulDePersoane* == **1**) {  
 *pret\_Pers* = **25;** } else if (*NumarulDePersoane* > **2**) {  
 *pret\_Pers* = **50** + (*NumarulDePersoane* - **2**) \* **20;** } else {  
 *pret\_Pers* = **0;** }  
 System.*out*.println("Pretul pentru acest numar de persoane este: " + *pret\_Pers* + " lei")**;** }  
}

Clasa rezervare zilnică ce este clasa derivata, conține o metodă **rezerv\_pers\_num** și o cariabila **pret\_Pers** care este egal la inceput cu 0.

In aceasta metodă noi calculam pretul unei rezervari zilnici în corespundere cu numarul de persoane , apoi pretul este afisat cu **System.out.println**

https://javarush.com/groups/posts/1950-metodih-ikh-parametrih-vzaimodeystvie-i-peregruzka

**Clasa derivată RezervareTip**

public class RezervareVip extends Rezervare{  
 static int *pret\_Pers* = **0;** public void rezerv\_vip(){  
 if(*NumarulDePersoane* == **5**){  
 *pret\_Pers* = **1000;** System.*out*.println(" Sala nr 1-Vip")**;** } else if (*NumarulDePersoane* > **5** && *NumarulDePersoane* < **10**) {  
 *pret\_Pers* = **2000;** System.*out*.println(" Sala nr 2-Vip")**;** }else if(*NumarulDePersoane* > **10**){  
 *pret\_Pers* = **5000;** System.*out*.println(" Sala nr 3-Vip")**;** }  
 System.*out*.println(" Pretul pentru Rezervare-Vip este: " + *pret\_Pers* + " lei")**;** }  
}

In clasa **RezervareVip** este clasa derivata cu o variabila si o metoda , este asemanator cu clasa **Rezervare Zilnica** insă sunt puse alte conditi la metodă

https://javarush.com/groups/posts/1950-metodih-ikh-parametrih-vzaimodeystvie-i-peregruzka

**Clasa derivată RezervareEvSpec**

public class RezervareEvSpec extends Rezervare{  
 static int *pret\_Pers* = **200;** static int *taxa\_ora* = **50;** public void rezerv\_taxa(){  
 System.*out*.println(" Taxa este de: " + (**200** + ( *DurataRez* \* **50**))+" lei")**;** }  
}

In clasa RezervareEvSpec ce moștenește clasa de bază Rezervare ce are 2 variabile **Pret\_Pers = 200** , și **taxa\_ora = 50** , cu aceste variabile lucreaza metoda **rezerv\_taxa** ce afiseaza taxa de plata a Rezervării date.

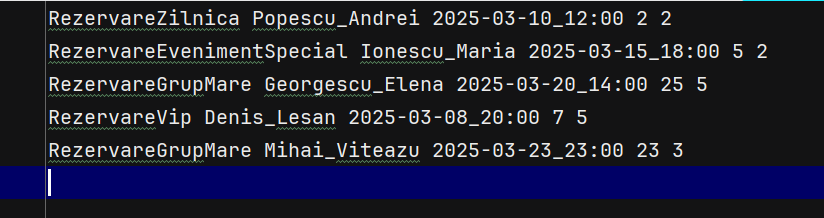
https://javarush.com/groups/posts/1950-metodih-ikh-parametrih-vzaimodeystvie-i-peregruzka

public class RezervareGrupMari extends Rezervare{  
 static int *Pret\_pers* = **0;** public void rezerv\_grup(){  
 if(*NumarulDePersoane* == **20**){  
 *Pret\_pers* = **500;** }else if(*NumarulDePersoane* > **20**){  
 *Pret\_pers* = **500** + (*NumarulDePersoane* - **20**) \* **10;** }else{  
 System.*out*.println(" Rezervarea nu se face - Grupul de persoane este mic!")**;** }  
 System.*out*.println(" Pretul pentru grup mare este: " + *Pret\_pers* + " lei")**;** }  
}

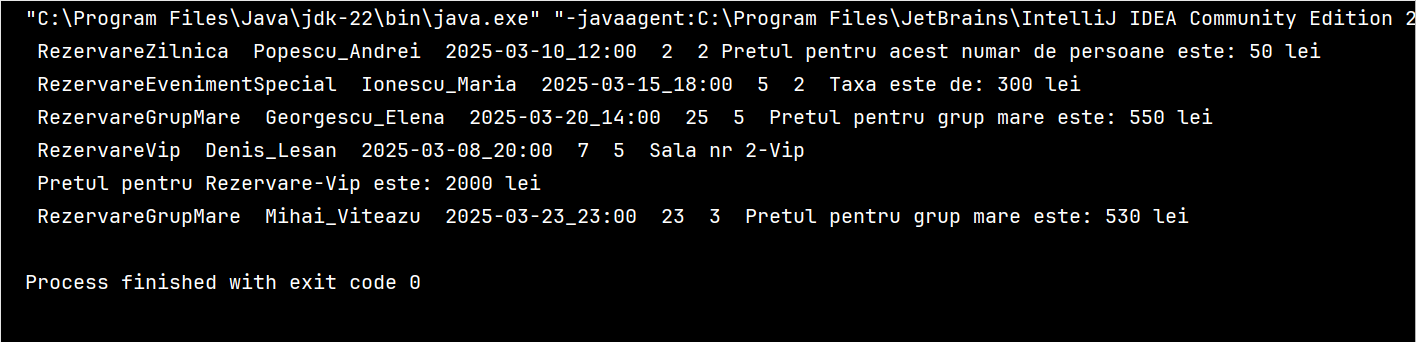
Clasa **RezervareGrupMari** este asemanator cu celelante clase derivate , tot așa moștenește clasa de bază dar diferenta este ca metoda sa **rezerv\_grup** are propriile condiți de afișare de exemplu dacă grupul este mic atunci se va afișa pe ecran propoziția data

**Rezultatul executari:**

Fișierul:



Execuția:



**Concluzie:**

În concluzie, codul oferă o bază funcțională pentru gestionarea rezervărilor, dar poate fi optimizat prin utilizarea instanțelor de clasă de exemplu **extends** pentru a evita modificarea globală a datelor și prin îmbunătățirea gestionării a codului fara al redacta radical.

Bibliografie

[1] Bibliotecile in Java, Accesat: 10/03/25, link-ul: <https://www.reddit.com/r/learnjava/comments/j364m8/what_does_importjavaio_mean/>