Ministerul Educației și Cercetării al Republicii Moldova

Colegiul Universității Tehnice a Moldovei

**Administrarea aplicatiilor web**

RAPORT

Lucrare de laborator nr.3

**Disciplina:** **Sisteme de gestiune a bazelor de date**

**Tema:** **Sistem de gestioanare a rezervarilor pentru restaurante**

A efectuat: Lipcanu Laurentiu AAW-231

Chişinău 2025

**Varianta 3**

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.**

#include <iostream>

#include <string>

#include <fstream>

#include <vector>

using namespace std;

class Rezervare {

protected:

string numeClient;

string dataOra;

int nrPersoane;

public:

void setNumeClient(string nume) {

this->numeClient = nume;

}

void setDataOra(string data) {

this->dataOra = data;

}

void setNrPersoane(int nr) {

this->nrPersoane = nr;

}

virtual int calculeazaDurata() {

return 2; // Durata standard a unei mese este 2 ore

}

virtual int calculeazaPret() {

return 0;

}

virtual string getTip() { return "Rezervare"; }

void afisareInfo() {

cout << "Tip: " << getTip() << endl;

cout << "Client: " << numeClient << endl;

cout << "Data si ora: " << dataOra << endl;

cout << "Numar persoane: " << nrPersoane << endl;

cout << "Durata: " << calculeazaDurata() << " ore" << endl;

cout << "Pret total: " << calculeazaPret() << " lei" << endl;

cout << endl;

}

};

class RezervareZilnica : public Rezervare {

public:

int calculeazaPret() override {

return nrPersoane <= 2 ? 50 : 50 + (nrPersoane - 2) \* 20;

}

string getTip() override { return "Rezervare Zilnica"; }

};

class RezervareEvenimentSpecial : public Rezervare {

private:

int oreSuplimentare;

public:

void setOreSuplimentare(int ore) {

this->oreSuplimentare = ore;

}

int calculeazaDurata() override {

return 4 + oreSuplimentare;

}

int calculeazaPret() override {

return 200 + oreSuplimentare \* 50;

}

string getTip() override { return "Rezervare Eveniment Special"; }

};

class RezervareGrupMare : public Rezervare {

public:

int calculeazaPret() override {

return nrPersoane <= 20 ? 500 : 500 + (nrPersoane - 20) \* 10;

}

string getTip() override { return "Rezervare Grup Mare"; }

};

class RezervareVIP : public Rezervare {

public:

int calculeazaPret() override {

return nrPersoane <= 5 ? 1000 : 1000 + (nrPersoane - 5) \* 50;

}

string getTip() override { return "Rezervare VIP"; }

};

int main() {

string path = "rezervari.txt";

ifstream fin(path);

if (!fin.is\_open()) {

cout << "Eroare la deschiderea fisierului " << path << " !" << endl;

return 1;

}

vector<Rezervare\*> rezervari;

string tipRezervare, numeClient, dataOra;

int nrPersoane, oreSuplimentare;

while (fin >> tipRezervare >> numeClient >> dataOra >> nrPersoane) {

Rezervare\* rezervare = nullptr;

if (tipRezervare == "RezervareZilnica") {

rezervare = new RezervareZilnica();

} else if (tipRezervare == "RezervareEvenimentSpecial") {

fin >> oreSuplimentare;

rezervare = new RezervareEvenimentSpecial();

dynamic\_cast<RezervareEvenimentSpecial\*>(rezervare)->setOreSuplimentare(oreSuplimentare);

} else if (tipRezervare == "RezervareGrupMare") {

rezervare = new RezervareGrupMare();

} else if (tipRezervare == "RezervareVIP") {

rezervare = new RezervareVIP();

}

if (rezervare) {

rezervare->setNumeClient(numeClient);

rezervare->setDataOra(dataOra);

rezervare->setNrPersoane(nrPersoane);

rezervari.push\_back(rezervare);

}

}

for (auto r : rezervari) {

r->afisareInfo();

delete r;

}

fin.close();

return 0;

}

**Fisier rezervari.txt**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

**Rezultatul compilarii**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**