

Ministerul Educației și Cercetării al Republicii Moldova  
Colegiul Universității Tehnice a Moldovei  
Administrarea Aplicațiilor Web

# RAPORT

## Lucrare de laborator nr.4

Disciplina: Asistență pentru programarea orientată pe obiecte

Tema: Moștenire, Polimorfism și Abstractizare

A efectuat:

Tintiuc Cătălin

Chișinău 2025

## Varianta I

### Problema: Sistem de gestionare a parcării

#### Codul în format text

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include <sstream>

using namespace std;

class IVehicul {
public:
    virtual int calculeazaTimp() const = 0;
    virtual double calculeazaTarif() const = 0;
    virtual ~IVehicul() {}
};

class Vehicul : public IVehicul {
protected:
    string numarInmatriculare;
    int oraIntrare, oraIesire;
public:
    Vehicul(const string& nr, int intrare, int iesire) :
numarInmatriculare(nr), oraIntrare(intrare), oraIesire(iesire) {}
    int calculeazaTimp() const override {
        return oraIesire - oraIntrare;
    }
    virtual double calculeazaTarif() const override = 0;
    virtual void afiseazaDetalii() const {
        cout << "Număr: " << numarInmatriculare << ", Timp: " <<
calculeazaTimp() << " ore, Tarif: " << calculeazaTarif() << " lei\n";
    }
};

class Masina : public Vehicul {
public:
    Masina(const string& nr, int intrare, int iesire) : Vehicul(nr, intrare,
iesire) {}
    double calculeazaTarif() const override {
        return calculeazaTimp() * 5.0;
    }
};
```

```

    }
};

class Motocicleta : public Vehicul {
public:
    Motocicleta(const string& nr, int intrare, int iesire) : Vehicul(nr,
intrare, iesire) {}
    double calculeazaTarif() const override {
        return calculeazaTimp() * 3.0;
    }
};

class Camion : public Vehicul {
public:
    Camion(const string& nr, int intrare, int iesire) : Vehicul(nr, intrare,
iesire) {}
    double calculeazaTarif() const override {
        return calculeazaTimp() * 10.0;
    }
};

int main() {
    vector<Vehicul*> vehicule;
    ifstream fisier("parcare.txt");
    if (!fisier) {
        cerr << "Eroare la deschiderea fişierului!\n";
        return 1;
    }

    string linie, tip, numar;
    int oraIntrare, oraIesire;
    while (getline(fisier, linie)) {
        stringstream ss(linie);
        ss >> tip >> numar >> oraIntrare >> oraIesire;

        if (tip == "Masina")
            vehicule.push_back(new Masina(numar, oraIntrare, oraIesire));
        else if (tip == "Motocicleta")
            vehicule.push_back(new Motocicleta(numar, oraIntrare, oraIesire));
        else if (tip == "Camion")
            vehicule.push_back(new Camion(numar, oraIntrare, oraIesire));
    }
    fisier.close();

    for (const auto& vehicul : vehicule) {
        vehicul->afiseazaDetalii();
    }
}

```

```
}  
  
for (auto& vehicul : vehicule) {  
    delete vehicul;  
}  
return 0;  
}
```

Fișierul parcare.txt

```
Masina B123XYZ 8 12  
Motocicleta B987XYZ 9 11  
Camion B456XYZ 7 13
```

Rezultatul compilării

```
Număr: B123XYZ, Timp: 4 ore, Tarif: 20 lei  
Număr: B987XYZ, Timp: 2 ore, Tarif: 6 lei  
Număr: B456XYZ, Timp: 6 ore, Tarif: 60 lei
```