

FACULTATEA CALCULATOARE, INFORMATICA SI
MICROELECTRONICA

UNIVERSITATEA TEHNICA A MOLDOVEI

ANALIZA PROIECTAREA SI PROGRAMAREA
ORIENTATA PE OBIECTE

LUCRAREA DE LABORATOR#1

Principiile POO

Autor:

Dumitrita GARABA

lector universitar:

Mihail PECARI

Lucrarea de laborator #1

0.1 Scopul lucrării de laborator

Studierea principiilor Programării Orientate pe Obiecte.

0.2 Obiective

- Alegerea limbajului POO
 - Crearea unei aplicații
 - Implementarea încapsulării
 - Implementarea mostenirii
 - Implementarea abstractizării
 - Implementarea polimorfismului

0.3 Efectuarea lucrării de laborator

0.3.1 Analiza lucrării de laborator

În această lucrare de laborator am elaborat o aplicație în limbajul Java, IDE-ul folosit fiind IntelliJ. Aplicația creată reprezintă o reprezentare a informației despre o companie, precum și calculul salariului lucrătorilor și profitul firmei.

Scopul lucrării de laborator a fost implementarea celor 4 principii POO: încapsularea, mostenirea, abstractizarea și polimorfismul. Încapsularea este proprietatea obiectelor de a-și ascunde o parte din date și metode. Din exteriorul obiectului sunt accesibile ("vizibile") numai datele și metodele publice. Acest principiu este vizibil în clasa Lucrator.

```
public class Lucrator extends Organizatie
private String NumeLucrator;
private String FunctieLucrator;
private double SalariuBrut;
private double Scutiri;
public String getNumeLucrator()
return NumeLucrator;
public String getFunctieLucrator()
return FunctieLucrator;
public void setNumeLucrator(String numeLucrator)
NumeLucrator = numeLucrator;
public void setFunctieLucrator(String functieLucrator)
FunctieLucrator = functieLucrator;
public double getScutiri()
return Scutiri;
public double getSalariuBrut()
return SalariuBrut;
public void setScutiri(double scutiri)
```

```

Scutiri = scutiri;
public void setSalariuBrut(double salariuBrut)
SalariuBrut = salariuBrut;

```

Mostenirea este proprietatea unei clase de a contine toate atributele (variabilele) si metodele superclasei sale. In consecinta, trecerea de la clasa la subclasa se face prin adaugarea de atribute si/sau de metode. Un exemplu de mostenire in aplicatia elaborata este intre clasa Organizatie si clasa Lucrator.

```

public class Lucrator extends Organizatie

```

Polimorfismul permite ca aceeasi operatie sa se realizeze in mod diferit in clase diferite sau in aceeasi clasa. Exemplu avem in clasa Lucrator.

```

public void SalariuNet(double SB)
double SN;
double Vi;
double Iv;
Vi = SB-648-486;
Iv = 0.07*(31140/2)+0.18*(Vi-31140/2);
SN = SB-(648+486+Iv);
System.out.println("Salariul net:" + SN);
public void SalariuNet(double SB, double S)
double SN;
double Vi;
double Iv;
Vi = SB-648-486-S;
Iv = 0.07*(31140/2)+0.18*(Vi-31140/2);
SN = SB-(648+486+Iv);
System.out.println("Salariul net:" + SN);

```

Abstractizarea este un proces de ascundere a detaliilor de implimentare si reprezentarea utilizatorului doar a functionalului. In aplicatie clasa Profit este o clasa abstracta.

```

public abstract class Profit
double CheltuieliLunare;
double VenitLunar;
public abstract void profit ();

```

0.3.2 Imagini

```

*****:

Detaliile despre organizatie:

Nume organizatie:Senaco Sistem
Fondator: Parfentieva Oxana
Anul fondarii:2010

*****:

Lucratorii organizatiei:

Nume si prenume:Bou Corina
Functia:Manager
Salariul net:10026.72
Nume si prenume:Albu Ina
Functia:Contabil
Salariul net:9638.82

*****:

Profitul organizatiei:

Profitul lunar este 150000.0
Profitul primei luni este 149500.0

```

Figure 1: Aplicatia

Concluzii

In aceasta lucrare de laborator mi-am dezvoltat abilitatile practice in elaborarea unei aplicatii in limbajul Java. IDE-ul folosit a fost IntelliJ.

Am studiat si implimentat cele patru principii ale Programarii orientate pe obiecte: incapsulare, mostenire, abstractizare si polimorfism. Incapsularea este proprietatea obiectelor de a-si ascunde o parte din date si metode. Din exteriorul obiectului sunt accesibile ("vizibile") numai datele si metodele publice. Mostenirea este proprietatea unei clase de a contine toate atributele (variabilele) si metodele superclasei sale. In consecinta, trecerea de la clasa la subclasa se face prin adaugarea de atribute si/sau de metode. Polimorfismul permite ca aceeaasi operatie sa se realizeze in mod diferit in clase diferite sau in aceeaasi clasa. Abstractizarea este un proces de ascundere a detaliilor de implimentare si reprezentarea utilizatorului doar a functionalului.

Programarea orientata pe obiecte este o tehnologie moderna in domeniul programarii calculatoarelor, care a rezultat din necesitatea realizarii de aplicatii din ce in ce mai complexe. Programarea clasica si structurata avea urmatoarele dezavantaje: control greoi al programelor de dimensiuni mari, dificultati cu privire la reutilizarea codurilor de programe si inflexibilitatea adaptarii si extinderii unor module de program deja realizate.

Bibliography

- [1] <https://www.codecademy.com/learn/learn-java>
- [2] <http://www.learnjavaonline.org/>
- [3] <https://www.javatpoint.com/java-oops-concepts>