

```

package com.itfactory;

import com.itfactory.UnitateBancara.BancaPentruLocuinte;
import com.itfactory.UnitateBancara.BancaPentruNevoiPersonale;
import com.itfactory.UnitateBancara.UnitateBancara;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        //1. Se cere sa se declare un obiect de tipul UnitateBancara
        (Atentie doar declarati in prima faza)

        UnitateBancara bancaPentruLocuinte = new BancaPentruLocuinte(1500,30)
    {
        @Override
        public double calculCredit(double Credit) {
            return super.calculCredit(Credit);
        }
    };
    //2.
    System.out.println("Introduceti datele necesare pentru analiza: "
+
        "\n Nume Prenume: " );
    Scanner scanner = new Scanner(System.in);
    String NumePrenume = scanner.nextLine();
    System.out.println("\n Varsta: " );
    Scanner scanner2 = new Scanner(System.in);
    int Varsta = Integer.parseInt(scanner2.nextLine());
    System.out.println("\n CNP:" );
    Scanner scanner3 = new Scanner(System.in);
    long CNP = Long.parseLong(scanner3.nextLine());
    System.out.println( "\n Salariu Net: ");
    Scanner scanner4 = new Scanner(System.in);
    double SalariuNet = Double.parseDouble(scanner.nextLine());

    if ( Varsta <=40) {
        if (Varsta < 18) {
            System.out.println("Persoana nu este eligibila pentru un
credit");
        } else {
            UnitateBancara bancaPentruLocuinte1 = new
BancaPentruLocuinte(SalariuNet,65-Varsta) {
                @Override
                public double calculCredit(double Credit) {
                    return super.calculCredit(Credit);
                }
            };
            System.out.println("Se poate acorda suma de: ");

```

```

System.out.println(bancaPentruLocuinte1.calculCredit(SalariuNet));
    System.out.println("cu dobanda de: ");

System.out.println(bancaPentruLocuinte1.calculDobanda(SalariuNet));

    }
    } else{
        UnitateBancara bancaPentruNevoiPersonale1 = new
        BancaPentruNevoiPersonale(SalariuNet, 70-Varsta) {
            @Override
            public double calculDobanda(double Dobanda) {
                return super.calculDobanda(Dobanda);
            }
        };

        System.out.println("Se poate acorda suma de: ");

System.out.println(bancaPentruNevoiPersonale1.calculCredit(SalariuNet));
    System.out.println("cu dobanda de: ");

System.out.println(bancaPentruNevoiPersonale1.calculDobanda(SalariuNet));

    }
    }
}

```