

Arhitectura (Design Pattern) Model View Controller

În arhitectura Model View Controller (MVC) aplicația propriu-zisă (obiectul { Model}) este proiectată independent de contextul în care va fi utilizată. Prin urmare, obiectul model poate fi asociat ulterior cu interfețe utilizator (obiecte { View}) de cele mai diverse tipuri. Pentru a realize interacțiunea dintre elementele de interfață și aplicație sunt utilizate obiecte de control ({ Controller}).

Exemplu În această foarte simplificată ilustrare a elementelor de arhitectură MVC, aplicația este alcătuită dintr-un obiect `m` din clasa `Model`, asupra caruia se poate acționa pentru a mări cu o unitate atributul `m.x` (prin metoda `increment()`) sau pentru a inspecta valoarea acestui atribut (metoda `get_x()`).

Interfața pe care o asociem modelului este de tip `graphic` și este obiect `v` al clasei `View`. Elementul de interacțiune pe care îl avem în vedere în acest exemplu este obiectul buton `v.b` al clasei `Button`. La fiecare acționare a sa este incrementată valoarea atributului `m.x` iar noua valoare este afișată în câmpul `v.tf`.

Interceptarea și prelucrarea evenimentelor generate prin acționarea butonului `b` sunt realizate prin intermediul unui obiect de control `c` al clasei `Controller` (această clasă implementează metoda `void actionPerformed(ActionEvent)` a interfeței `ActionListener`).

Relația dintre clasele din acest exemplu este prezentată în Figura `MVCClase`. Colaborarea dintre obiectul model, obiectele de interfață și obiectul de control este ilustrată în Figura `MVCColaborare`. Prin mesajul `b.addActionListener(c)` trimis către obiectul buton `b` se face operația de { instalare } a evenimentelor. După instalare, acționarea butonului `b` este { notificată } prin mesajul `actionPerformed(...)` trimis obiectului de control. Ca răspuns, obiectul de control trimite mesaje către modelul `m` pentru modificarea și afișarea atributului `m.x`.

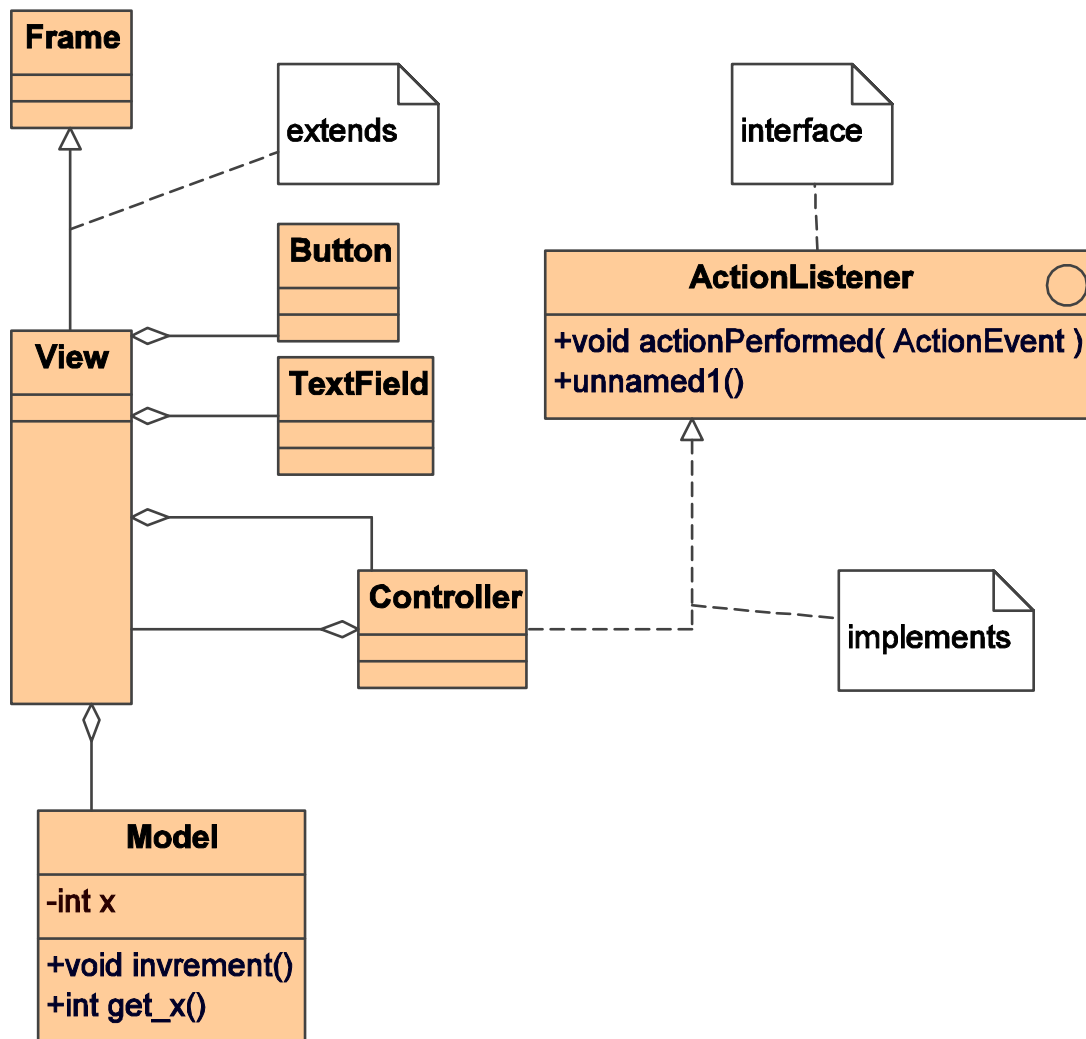


Figura MVCClase. Relația dintre clase

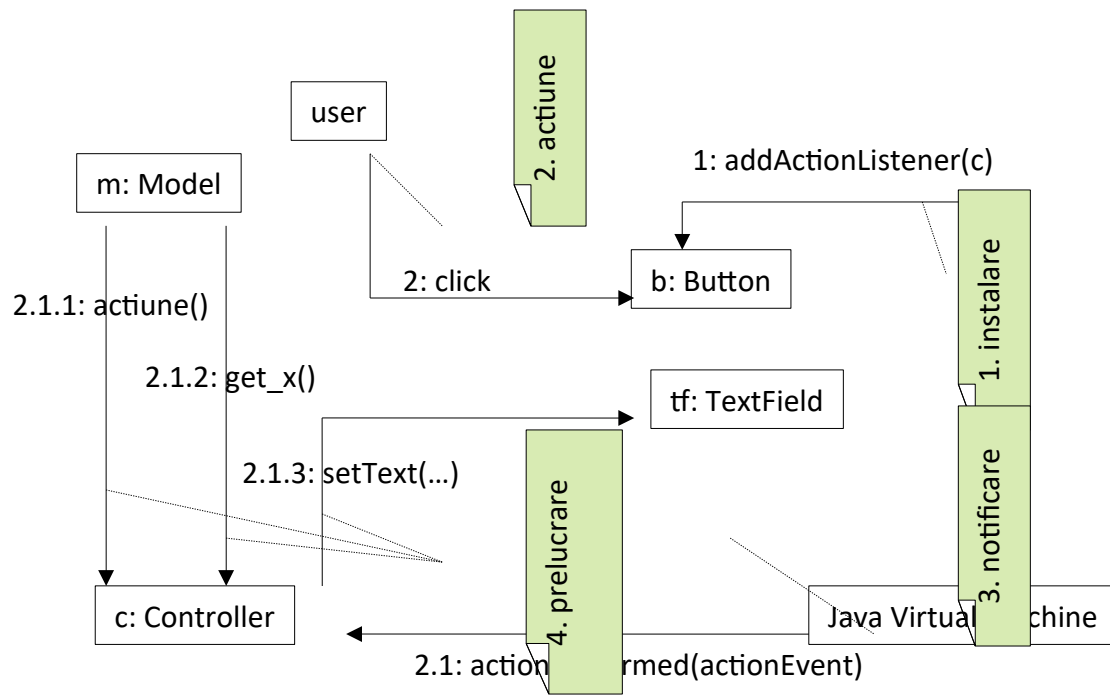


Figura MVCColaborare. Diagrama de colaborare între obiectele MVC

```
// Model View Controller

import java.awt.*;
import java.awt.event.*;
class Model{
    private int x=0;
    public Model(){};
    public void increment(){x++;}
    public int get_x(){return x;}
}
public class View extends Frame{
    private Button b;
    protected Model m;
    private Controller c;
    protected TextField tf;
    public static void main(String args[]){
        Frame v= new View();
    }
    public View(){
        setTitle("Exemplu Model-View-Controller");

        b= new Button("Actiune");
        add("North",b);

        m=new Model();

        c=new Controller(this);
        b.addActionListener(c);

        tf=new TextField(10);
        add("Center",tf);

        setSize(100,250);
    }
}
```

```
        setVisible(true);
    }
}

class Controller implements ActionListener{
    private View vw;
    public Controller(View v){
        vw=v;
    }
    public void actionPerformed(ActionEvent e){
        vw.m.increment();
        vw.tf.setText(String.valueOf(vw.m.get_x()));
    }
}
```