P1

#pragma once

class Data{

int zi;

int luna;

int an;

public:

Data(int z,int l,int a);

Data();

Data(const Data &a);

void set\_an(int a){an=a;};

void set\_zi(int z){zi=z;};

void set\_luna(int l){luna=l;};

int get\_zi(){return zi;};

int get\_luna(){return luna;}

int get\_an(){return an;}

Data &operator=(const Data &a);

~Data(){}

};

class Mamifer{

char\* tip;

Data nastere;

public:

virtual ~Mamifer(){delete tip;}

Mamifer();

Mamifer(Data a,char \*type);

virtual bool Mananca(){return true;};

virtual bool Merge\_la\_baie(){return true;}

virtual bool Hraneste\_animale(){return true;}

virtual void miauna()=0;

};

class Bunica:public Mamifer{

public:

Bunica():Mamifer(){};

Bunica(Data a,char \*type):Mamifer(a,type){}

~Bunica(){};

void miauna(){};

};

class PisicaDeCartier:public Mamifer{

public:

PisicaDeCartier():Mamifer(){}

~PisicaDeCartier(){}

PisicaDeCartier(Data a,char \*type):Mamifer(a,type){}

void miauna();

};

class PisicaSiameza:public Mamifer{

public:

PisicaSiameza():Mamifer(){}

~PisicaSiameza(){}

PisicaSiameza(Data a,char \*type):Mamifer(a,type){}

void miauna();

};

class PisicaEgipteana:public Mamifer{

public:

PisicaEgipteana():Mamifer(){}

~PisicaEgipteana(){}

PisicaEgipteana(Data a,char \*type):Mamifer(a,type){}

void miauna();

};

Fct.c

#include<iostream>

#include<string.h>

#include"head.h"

using namespace std;

Data::Data(){

an=0;

zi=0;

luna=0;

}

Data::Data(int z,int l,int a){

zi=z;

luna=l;

an=a;

}

//constr de copiere: aici nu e neaparata nevoie pt ca nu exista membri aloc dinamic

Data::Data(const Data &a){

zi=a.zi;

luna=a.luna;

an=a.an;

}

//supr op + aici nu e neaparata nevoie pt ca nu exista membri aloc dinamic

Data &Data::operator=(const Data &a){

zi=a.zi;

an=a.an;

luna=a.luna;

return \*this;

}

Mamifer::Mamifer(){

tip=new char[strlen("divine")+1];

strcpy(tip,"devine");

}

Mamifer::Mamifer(Data a,char \*type){

nastere=a;

tip=new char[strlen(type)+1];

strcpy(tip,type);

}

void PisicaSiameza::miauna(){

cout<<" fals"<<endl;

}

void PisicaDeCartier::miauna(){

cout<<" gangsta"<<endl;

}

void PisicaEgipteana::miauna(){

cout<<" egiptean"<<endl;

}

Main.c

#include<iostream>

#include<string.h>

#include"head.h"

#include<conio.h>

using namespace std;

int main()

{

Data d(22,7,1950);

Mamifer \*m[4];

m[0]=new Bunica(d,"devil");

m[1]=new PisicaSiameza();

m[2]=new PisicaDeCartier();

m[3]=new PisicaEgipteana();

for(int i=1;i<4;i++)

{

cout<<"pisica "<<i<<" miauna";

m[i]->miauna();

}

\_getch();

return 0;

}

P2

data segment

n dw 5

rezF dw 1

rezS dw 0

ends

stack segment

dw 128 dup(0)

ends

code segment

start:

mov ax,data

mov ds, ax

mov es, ax

mov cx, n

fact:

mov ax, rezF

mul cx

mov rezF, ax

loop fact

mov cx, n

sum:

mov ax, rezS

add ax, cx

mov rezS, ax

loop sum

; add your code here

mov ax, 4c00h

int 21h

ends

end start

P3

data segment

t db 1,2,3,4,5,6

len equ $-t

rezS db 0

ends

stack segment

dw 128 dup(0)

ends

code segment

start:

mov ax,data

mov ds, ax

mov es, ax

mov bx, offset t

xor si, si

mov cx, len

for:

xor ah,ah

mov al, rezS

add al, [bx][si]

mov rezS, al

inc si

loop for

; add your code here

mov ax, 4c00h

int 21h

ends

end start

P4

data segment

vect db 6,2,9,1,3,4,0,5

len equ $-vect

ends

stack segment

dw 128 dup(0)

ends

code segment

start:

mov ax,data

mov ds, ax

mov es, ax

mov bx, offset vect

mov dl, 1

while:

mov dl, 0

xor si, si

mov cx, len-1

for:

mov ah, [bx][si+1]

mov al, [bx][si]

cmp al, ah

jle next

xor al, ah

xor ah, al

xor al, ah

mov [bx][si+1], ah

mov [bx][si], al

mov dl, 1

next:

inc si

loop for

cmp dl, 1

je while

; add your code here

mov ax, 4c00h

int 21h

ends

end start

P5

data segment

vect db 6,2,9,1,3,4,0,5

len equ $-vect

ends

stack segment

dw 128 dup(0)

ends

code segment

start:

mov ax,data

mov ds, ax

mov es, ax

mov si, 1

mov cx, len-1

mov bx, offset vect

for:

mov di, si

while:

cmp di, 0

jle next

mov al, [bx][di-1]

mov ah, [bx][di]

cmp al, ah

jle next

xchg al, ah

mov [bx][di-1], al

mov [bx][di], ah

dec di

jmp while

next:

inc si

loop for

mov ax, 4c00h

int 21h

ends

end start

void insertionSort(int arr[], int length) {

int i, j, tmp;

for (i = 1; i < length; i++) {

j = i;

while (j > 0 && arr[j - 1] > arr[j]) {

tmp = arr[j];

arr[j] = arr[j - 1];

arr[j - 1] = tmp;

j--;

}

}

}

P6

#pragma once

template <class T>

struct Nod{

T data;

Nod \*next;

};

template <class T>

class Stack{

Nod<T> \*head;

public:

Stack(){

head = new Nod<T>;

head->data = 0;

head->next = 0;

}

Stack(T data){

head = new Nod<T>;

head->data = data;

head->next = 0;

}

void push(T data){

Nod<T> \*p = new Nod<T>;

p->data = data;

p->next = head;

head = p;

}

T top (){

return head->data;

}

void pop(){

if(!isEmpty()){

Nod<T> \*p = new Nod<T>;

p = head;

head = head->next;

delete p;

}

}

bool isEmpty(){

if(head->next == 0){

return true;

}else{

return false;

}

}

};

Main.c

#include "Header.h"

#include <iostream>

using namespace std;

int main(){

Stack<int> stack;

stack.push(1);

stack.push(0);

stack.push(2);

stack.push(3);

stack.pop();

if(!stack.isEmpty())

cout<<"2: "<<stack.top();

while(!stack.isEmpty()){

stack.pop();

}

cout<<"\n"<<stack.top();

return 0;

}

P7

#pragma once

#include <iostream>

using namespace std;

template<class T>

struct Nod{

Nod \*next;

T data;

};

template<class T>

class List{

Nod<T> \*head;

public:

List(){

head = 0;

}

List(T data){

head = new Nod<T>;

head -> data = data;

head ->next = head;

}

void insertElement(T data){

if(head == 0){

head = new Nod<T>;

head->data = data;

head->next = head;

}else{

Nod<T> \*aux = new Nod<T>;

aux->data = data;

aux->next = head->next;

head->next = aux;

head = aux;

}

}

void removeHead(){

Nod<T> aux = head;

head = head->next;

delete aux;

}

void print(){

Nod<T> \*aux = head->next;

cout<<endl;

while(aux != head){

cout<<aux->data<<" ";

aux = aux->next;

}

cout<<head->data;

}

void printHead(){

cout<<"\nHead is: "<<head->data;

}

void nextElem(){

head = head->next;

}

};

Main.c

#include "Header.h"

#include <iostream>

using namespace std;

int main(){

cout<<"Introduceti elemente( 0 = Stop ): ";

int val;

List<int> list;

cin>>val;

while (val){

list.insertElement(val);

cin>>val;

}

list.print();

return 0;

}

P8

#pragma once

#include <iostream>

using namespace std;

class Complex{

int re;

int im;

public:

Complex(){

re = im = 0;

}

Complex(int r, int i){

re = r;

im = i;

}

Complex &operator +=(const Complex &rhs){

this->re += rhs.re;

this->im += rhs.im;

return \*this;

}

Complex &operator -=(const Complex &rhs){

this->re -= rhs.re;

this->im -= rhs.im;

return \*this;

}

friend Complex operator+ (const Complex &a, const Complex &b){

int re = a.re + b.re;

int im = a.im + b.im;

return Complex(re,im);

}

friend ostream& operator<<(std::ostream& os, const Complex& obj){

os<<obj.re<<"+"<<obj.im<<"i";

return os;

}

};

template<class T>

struct Nod{

Nod \*next;

T data;

};

template<class T>

class List{

Nod<T> \*head;

public:

List(){

head = 0;

}

List(T data){

head = new Nod<T>;

head -> data = data;

head ->next = head;

}

void insertElement(T data){

if(head == 0){

head = new Nod<T>;

head->data = data;

head->next = head; //lista circulara

}else{

Nod<T> \*aux = new Nod<T>;

aux->data = data;

aux->next = head->next;

head->next = aux;

head = aux;

}

}

void removeHead(){

Nod<T> aux = head;

head = head->next;

delete aux;

}

void print(){

Nod<T> \*aux = head->next;

cout<<endl;

while(aux != head){

cout<<aux->data<<" ";

aux = aux->next;

}

cout<<head->data;

}

void printHead(){

cout<<"\nHead is: "<<head->data;

}

void nextElem(){

head = head->next;

}

};

Main.c

#include "Header.h"

#include <iostream>

using namespace std;

int main(){

cout<<"Introduceti elemente( 0 = Stop ): ";

int valR,valI;

List<Complex> list;

cin>>valR;

cin>> valI;

while (valR){

Complex val(valR,valI);

list.insertElement(val);

cin>>valR;

cin>>valI;

}

Complex a(1,1), b(2,2), c;

c = a+b;

c+=a;

cout<<"\n"<<c;

list.print();

return 0;

}

P9

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Animal{

string name;

public:

Animal(){

name = "";

}

Animal(string name){

this->name = name;

}

virtual ~Animal(){}

virtual void Speak(){}

string getName(){

return name;

}

};

class Dog : public Animal{

public:

Dog():Animal("Nameless Dog"){};

Dog(string name): Animal(name){}

~Dog(){}

void Speak(){

cout<<endl<<getName().c\_str()<<": Hello all, I bark!";

}

};

class Cat : public Animal{

public:

Cat():Animal("Nameless Cat"){};

Cat(string name): Animal(name){};

~Cat(){}

void Speak(){

cout<<endl<<getName().c\_str()<<": Hello all, I meow!";

}

};

class Horse : public Animal{

public:

Horse():Animal("Nameless Horse"){}

Horse(string name): Animal(name){}

~Horse(){}

void Speak(){

cout<<endl<<getName().c\_str()<<": Hello all, I don't know what to call my sounds!";

}

};

Main.c

#include "Header.h"

#include <iostream>

using namespace std;

int main(){

Animal \*yard[3];

Cat cat("Pussy");

Dog dog("Spike");

Horse horse("Retarded Horse");

yard[0] = &cat;

yard[1] = &dog;

yard[2] = &horse;

for(int i=0;i<3;i++){

yard[i]->Speak();

}

}

P10

package Stack;

import java.util.\*;

interface AbstractStack<T> {

public void push(T data);

public void pop();

public boolean isEmpty();

public T top();

}

public class MyStack<T> implements AbstractStack<T>{

ArrayList<T> head = new ArrayList<T>();

int index;

public MyStack(){

index = -1;

}

public MyStack(T value){

head.add(value);

index = 0;

}

public void push(T data){

index += 1;

head.add(index,data);

}

public void pop(){

head.remove(index);

index -= 1;

}

public boolean isEmpty(){

return head.isEmpty();

}

public T top(){

T aux = head.get(index);

pop();

return aux;

}

}

package Stack;

public class StackImplementation {

public static void main(String[] args){

MyStack<String> stack = new MyStack<String>();

stack.push("Sergiu");

stack.push("Marius");

stack.push("Dragos");

stack.pop();

System.out.println(stack.top());

System.out.println(stack.isEmpty());

}

}

P11

package CircularList;

public interface AbstractList<T>{

public void addElement(T element);

public void removeElement(int index);

public void printList(int index);

public boolean isEmpty();

public T getElement(int index);

}

package CircularList;

import java.util.\*;

public class CircularList<T> implements AbstractList<T> {

ArrayList<T> head = new ArrayList<T>();

int index;

public CircularList(){

index = -1;

}

public CircularList(T element){

head.add(element);

index = 0;

}

@Override

public void addElement(T element) {

index += 1;

head.add(element);

}

@Override

public void removeElement(int index) {

if(index > this.index){

index = index % this.index;

}

head.remove(index);

this.index -= 1;

}

@Override

public boolean isEmpty() {

return head.isEmpty();

}

@Override

public T getElement(int index) {

if(index > this.index){

index = index % this.index;

}

return head.get(index);

}

@Override

public void printList(int indx) {

for(int i = 0; i <= this.index; i++){

if(indx > this.index){

indx = indx % this.index;

}

System.out.print(head.get(indx)+" ");

indx += 1;

}

System.out.println();

}

}

package CircularList;

public class ImplementCircularList {

public static void main(String[] args){

CircularList<Integer> list = new CircularList<Integer>();

list.addElement(1);

list.addElement(2);

list.addElement(3);

list.addElement(4);

list.removeElement(0);

list.addElement(1);

list.printList(0);

list.printList(2);

list.printList(3);

list.printList(76);

}

}

P12

package Ceas;

public class Afisaj {

Afisaj(){

}

public void clipire(int nrApasari, Timp timp) {

if(nrApasari == 1){

System.out.println("Ore: "+timp.getOre());

}

else if(nrApasari == 2){

System.out.println("Minute: "+timp.getMinute());

}

}

}

package Ceas;

import java.util.Random;

public class Baterie {

int status;

Baterie(){

status = 100;

}

public int check() {

Random randomGenerator = new Random();

status -= randomGenerator.nextInt(10);

return status;

}

}

package Ceas;

public class Buton {

int nrApasari;

Buton(){

nrApasari = 0;

}

public int buton1() {

nrApasari += 1;

if(nrApasari == 3){

nrApasari = 0;

return 3;

}

return nrApasari;

}

public int buton2() {

return nrApasari;

}

}

package Ceas;

import java.util.\*;

public class CeasSimplu {

public static void main(String[] args){

Scanner scan = new Scanner(System.in);

Date data = new Date();

Baterie baterie = new Baterie();

Buton buton = new Buton();

Afisaj ecran = new Afisaj();

@SuppressWarnings("deprecation")

Timp timp = new Timp(data.getHours(),data.getMinutes());

int option;

do{

int sts = baterie.check();

System.out.println("Baterie incarcata "+sts+"%");

System.out.println("Este ora "+ timp.getOre()+":"+timp.getMinute());

if(sts <= 1){

System.out.println("Baterie consumata");

break;

}

System.out.println("1. Buton 1");

System.out.println("2. Buton 2");

System.out.println("3. Buton 1-2");

System.out.println("0. Exit");

System.out.print(">>");

option = scan.nextInt();

if(option == 1){

ecran.clipire(buton.buton1(),timp);

}

else if(option == 2){

timp.increment(buton.buton2());

}

else if(option == 3){

System.out.println("Modificari memorate!");

}

else if(option == 0){

System.out.println("La revedere!");

}

}while(option != 0);

scan.close();

}

}

package Ceas;

public class Timp {

int hours,minutes;

public Timp(int hours, int minutes) {

this.hours = hours;

this.minutes = minutes;

}

public void increment(int nrApasari) {

if(nrApasari == 1){

this.hours += 1;

if(hours >= 24){

hours = 0;

}

}

else if(nrApasari == 2){

this.minutes += 1;

if(this.minutes >= 60){

this.minutes = 0;

this.hours += 1;

if(hours >= 24){

hours = 0;

}

}

}

}

public int getOre() {

return this.hours;

}

public int getMinute() {

return this.minutes;

}

}

P13

package p13;

import java.util.Scanner;

public class Automat {

String []state={"00","01","10","11"};

boolean reset=true;

int index;

int value;

Automat(){

index=0;

}

public String get\_state(){

return state[index];

}

public void functionare(){

Scanner scan=new Scanner(System.in);

while(reset ){

System.out.print("value= ");

value=scan.nextInt();

if(value>1){

reset=false;

}

else{

state(value);

}

if(get\_state().equals("11")){

System.out.print("1 ");

}

}

scan.close();

System.out.println("Automatul si-a dat reset");

}

public void state(int x){

switch(index){

case 0:{if(x==0)index=2;else if(x==1)index=1;}break;

case 1:{if(x==0)index=2;else if(x==1)index=3;}break;

case 2:{if(x==0)index=3;else if(x==1)index=1;}break;

case 3:{if(x==0)index=2;else if(x==1)index=1;}break;

default:break;

}

}

}

package p13;

public class MyClass {

public static void main(String[] args) {

Automat a=new Automat();

a.functionare();

}

}

P14

package p14;

public class Counter extends Thread {

private CounterEvent ce;

int count;

public Counter( String name, int count ){

super(name);

this.count=count;

}

public Counter(CounterEvent b,int count){

ce=b;

this.count=count;

}

public Counter(){

count=0;

}

public void run(){

for(int i=0;i<count;i++){

try{

sleep(1500);

}catch(InterruptedException e){}

ce.put(i);

System.out.println("Numaratorul are valoarea:\t"+i);

}

}

}

class Receptor extends Thread{

private CounterEvent ce;

int count;

public Receptor(CounterEvent b,int count){

ce=b;

this.count=count;

}

public void run(){

int value=0;

for(int i=0;i<count;i++){

value=ce.getCount();

System.out.println("Receptorul a primt valoarea:\t"+value);

}

}

}

package p14;

import java.util.Scanner;

public class CounterControl {

public static void main(String[] args) {

int nr;

Scanner scan=new Scanner(System.in);

System.out.print("Valoarea maxima=");

nr=scan.nextInt();

scan.close();

CounterEvent b=new CounterEvent();

Counter c=new Counter(b,nr);

Receptor r=new Receptor(b,nr);

c.start();

r.start();

}

}

package p14;

public class CounterEvent {

private int count;

boolean available=false;

public CounterEvent ( int count ){

this.count=count;

}

public CounterEvent(){

count=-1;

}

public synchronized int getCount(){

while(!available){

try{

wait();

}

catch(InterruptedException e){e.printStackTrace();

}

}

available=false;

notifyAll();

return count;

}

public synchronized void put(int number){

while(available){

try{

wait();

}catch(InterruptedException e){

e.printStackTrace();

}

}

this.count=number;

available=true;

notifyAll();

}

}

P15

package p15;

import java.awt.\*;

import java.awt.event.\*;

@SuppressWarnings("serial")

public class Butoane extends Frame implements ActionListener{

Button []b;

Button exit;

TextField text;

Butoane(String titlu){

super(titlu);

this.addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e){

dispose();

System.exit(0);

}

});

this.setLayout(new FlowLayout());

this.setSize(200, 200);

text=new TextField();

add(text);

b=new Button[3];

b[0]=new Button("1");

b[1]=new Button("2");

b[2]=new Button("3");

exit=new Button("EXIT");

for(int i=0;i<3;i++){

add(b[i]);

b[i].addActionListener(this);

}

add(exit);

exit.addActionListener(this);

}

public void actionPerformed(ActionEvent e) {

String command=e.getActionCommand();

if(command.equals("EXIT")){

System.exit(0);

}

else{

text.setText(command);

text.setBackground(Color.yellow);

}

}

}

package p15;

public class Myclass {

@SuppressWarnings("deprecation")

public static void main(String []args){

Butoane F=new Butoane("Varianta 15");

F.show();

}

}

P16

package p16;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

@SuppressWarnings("serial")

public class Butoane extends JFrame implements ActionListener{

private JButton []b;

private JTextField txt;

private JButton exit;

Butoane(String nume){

super(nume);

this.addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e){

dispose();

System.exit(0);

}

});

getContentPane().setLayout(new FlowLayout());

setSize(200, 200);

b=new JButton[3];

for(int i=0;i<3;i++){

b[i]=new JButton(""+(i+1));

add(b[i]);

b[i].addActionListener(this);

}

exit=new JButton("EXIT");

add(exit);

exit.addActionListener(this);

txt=new JTextField();

add(txt);

}

public void actionPerformed(ActionEvent e) {

String command=e.getActionCommand();

if(command.equals("EXIT")){

System.exit(0);

}

else{

txt.setText(command);

txt.setSize(25, 25);

txt.setBackground(Color.ORANGE);

}

}

}

package p16;

public class Myclass {

@SuppressWarnings("deprecation")

public static void main(String[] args) {

Butoane b=new Butoane("Myclass");

b.show();

}

}

P17

package p17;

import javax.swing.\*;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.event.\*;

@SuppressWarnings("serial")

public class MyMouse extends JFrame implements ActionListener {

JTextField txt;

private static int x,y;

MyMouse(String nume){

super(nume);

getContentPane().setLayout(new FlowLayout());

setSize(400,400);

this.addMouseListener(new MouseAdapter(){

public void mouseClicked(MouseEvent e) {

x=e.getX();

y=e.getY();

txt.setSelectionStart(10);

txt.setSize(90, 20);

txt.setText("X= "+x+" "+"Y= "+y);

txt.setBackground(Color.orange);

txt.setLocation(0,0);

}

});

this.addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e){

dispose();

System.exit(0);

}

});

txt=new JTextField();

add(txt);

}

public void actionPerformed(ActionEvent e) {

}

}

package p17;

public class Myclass {

@SuppressWarnings("deprecation")

public static void main(String []args){

MyMouse m=new MyMouse("Varianta 17");

m.show();

}

}

P18

package p18;

import java.awt.\*;

import java.awt.event.\*;

@SuppressWarnings("serial")

public class MyMouse extends Frame implements ActionListener {

TextField txt;

private static int x,y;

MyMouse(String nume){

super(nume);

this.setLayout(new FlowLayout());

setSize(400,400);

this.addMouseListener(new MouseAdapter(){

public void mouseClicked(MouseEvent e) {

x=e.getX();

y=e.getY();

txt.setSelectionStart(10);

txt.setSize(90, 20);

txt.setText("X= "+x+" "+"Y= "+y);

txt.setBackground(Color.cyan);

txt.setLocation(6,30);

}

});

this.addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e){

dispose();

System.exit(0);

}

});

txt=new TextField();

add(txt);

}

public void actionPerformed(ActionEvent e) {

}

}

package p18;

public class Myclass {

@SuppressWarnings("deprecation")

public static void main(String []args){

MyMouse m=new MyMouse("Varianta 17");

m.show();

}

}

P19

package p19;

import java.awt.\*;

@SuppressWarnings("serial")

public class draw extends Canvas{

boolean change;

private static int nr\_erori=0;

private int lungime;

private char []cuvant;

private char []rez;

private boolean [] vizibil;

draw(char cuv[]){

lungime=cuv.length;

cuvant=new char[lungime];

rez=new char[lungime];

vizibil=new boolean[lungime];

for(int i=0;i<lungime;i++)

{

rez[i]='\*';

cuvant[i]=cuv[i];

vizibil[i]=false;

}

}

void Game(char litera){

change=true;

for(int i=0;i<lungime;i++){

if(cuvant[i]==litera){

vizibil[i]=true;

rez[i]=cuvant[i];

change=false;

repaint();

}

}

if(change==true)

{ nr\_erori++;

repaint();

}

}

boolean Get\_vizibil(){

for(int i=0;i<lungime;i++){

if(vizibil[i]==false)

return false;

}

return true;

}

int Get\_erori(){

return nr\_erori;

}

public void paint(Graphics g){

g.setColor(Color.BLACK);

g.fillRect(40, 260, 80, 30);

g.fillRect(60, 40, 40, 220);

g.fillRect(100,40,130,40);

g.fillRect(190, 80, 16, 40);

if(nr\_erori==1)

g.drawOval(172, 120, 51, 51);

g.setFont(new Font("Courier",Font.PLAIN,15));

String s1=new String(rez);

g.drawString("Cuvant "+s1, 70, 400);

g.drawString("Incercari gresite "+nr\_erori, 70, 430);

if(nr\_erori==2){

g.drawOval(172, 120, 51, 51);

g.fillRect(195, 170, 3, 70);

}

if(nr\_erori==3){

g.drawOval(172, 120, 51, 51);

g.fillRect(195, 170, 3, 70);

g.drawLine(195, 170, 155, 190);

}

if(nr\_erori==4){

g.drawOval(172, 120, 51, 51);

g.fillRect(195, 170, 3, 70);

g.drawLine(195, 170, 155, 190);

g.drawLine(198, 170, 235, 190);

}

if(nr\_erori==5){

g.drawOval(172, 120, 51, 51);

g.fillRect(195, 170, 3, 70);

g.drawLine(195, 170, 155, 190);

g.drawLine(198, 170, 235, 190);

g.drawLine(195, 240, 155, 270);

}

if(nr\_erori==6){

g.drawOval(172, 120, 51, 51);

g.fillRect(195, 170, 3, 70);

g.drawLine(195, 170, 155, 190);

g.drawLine(198, 170, 235, 190);

g.drawLine(195, 240, 155, 270);

g.drawLine(198, 240, 235, 270);

}

}

}

package p19;

import java.awt.BorderLayout;

import java.awt.Frame;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import java.util.Scanner;

@SuppressWarnings("serial")

public class Spanzuratoare extends Frame {

draw joc;

char [][] cuvinte\_joc;

Spanzuratoare(int optiune){

setSize(500,500);

cuvinte\_joc =new char[10][];

this.setVisible(true);

char []cuv1={'b','i','l','i','a','r','d'};

cuvinte\_joc[1]=cuv1;

char []cuv2={'e','x','t','r','a','t','e','r','e','s','t','r','u'};

cuvinte\_joc[2]=cuv2;

char []cuv3={'t','e','l','e','n','c','i','c','l','o','p','e','d','i','e'};

cuvinte\_joc[3]=cuv3;

char []cuv4={'l','i','n','g','v','i','s','t','i','c'};

cuvinte\_joc[4]=cuv4;

char []cuv5={'a','s','t','r','o','n','a','u','t'};

cuvinte\_joc[5]=cuv5;

char []cuv6={'j','a','v','a','s','c','r','i','p','t'};

cuvinte\_joc[6]=cuv6;

char []cuv7={'a','m','s','t','e','r','d','a','m',};

cuvinte\_joc[7]=cuv7;

char []cuv8={'g','e','r','m','a','n','i','a'};

cuvinte\_joc[8]=cuv8;

char []cuv9={'p','a','r','a','d','i','g','m'};

cuvinte\_joc[9]=cuv9;

char []cuv0={'p','o','l','i','m','o','r','f','i','s','m'};

cuvinte\_joc[0]=cuv0;

joc=new draw(cuvinte\_joc[optiune]);

add(joc,BorderLayout.CENTER);

this.addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e){

dispose();

System.exit(0);

}

});

JOC();

}

void JOC(){

Scanner reader = new Scanner(System.in);

while(joc.Get\_erori()<6 && !joc.Get\_vizibil()){

System.out.print("Litera= ");

char c = reader.next().trim().charAt(0);

joc.Game(c);

}

reader.close();

if(joc.Get\_vizibil())

System.out.println("FELICITARI AI CASTIGAT");

else

System.out.println(" AI PIERDUT");

}

}

package p19;

import java.util.Scanner;

public class MyClass {

@SuppressWarnings("deprecation")

public static void main(String []args){

System.out.println("Introduceti optiune joc\nAveti la dispozitie 10 jocuri\nSelectati o valoare de la 0 la 9");

System.out.println("optiune=");

int opt;

Scanner scan=new Scanner(System.in);

opt=scan.nextInt();

Spanzuratoare a=new Spanzuratoare(opt);

scan.close();

a.show();

}

}

P20

package p20;

import java.awt.\*;

import javax.swing.\*;

@SuppressWarnings("serial")

public class draw extends JPanel{

boolean change;

private static int nr\_erori=0;

private int lungime;

private char []cuvant;

private char []rez;

private boolean [] vizibil;

draw(char cuv[]){

lungime=cuv.length;

cuvant=new char[lungime];

rez=new char[lungime];

vizibil=new boolean[lungime];

for(int i=0;i<lungime;i++)

{

rez[i]='\*';

cuvant[i]=cuv[i];

vizibil[i]=false;

}

}

void Game(char litera){

change=true;

for(int i=0;i<lungime;i++){

if(cuvant[i]==litera){

vizibil[i]=true;

rez[i]=cuvant[i];

change=false;

removeAll();

revalidate();

repaint();

}

}

if(change==true)

{ nr\_erori++;

removeAll();

revalidate();

repaint();

}

}

boolean Get\_vizibil(){

for(int i=0;i<lungime;i++){

if(vizibil[i]==false)

return false;

}

return true;

}

int Get\_erori(){

return nr\_erori;

}

public void paint(Graphics g){

super.paint(g);

g.setColor(Color.BLUE);

g.fillRect(40, 260, 80, 30);

g.fillRect(60, 40, 40, 220);

g.fillRect(100,40,130,40);

g.fillRect(190, 80, 16, 40);

if(nr\_erori==1)

g.drawOval(172, 120, 51, 51);

g.setFont(new Font("Courier",Font.PLAIN,15));

String s1=new String(rez);

g.drawString("Cuvant "+s1, 70, 400);

g.drawString("Incercari gresite "+nr\_erori, 70, 430);

if(nr\_erori==2){

g.drawOval(172, 120, 51, 51);

g.fillRect(195, 170, 3, 70);

}

if(nr\_erori==3){

g.drawOval(172, 120, 51, 51);

g.fillRect(195, 170, 3, 70);

g.drawLine(195, 170, 155, 190);

}

if(nr\_erori==4){

g.drawOval(172, 120, 51, 51);

g.fillRect(195, 170, 3, 70);

g.drawLine(195, 170, 155, 190);

g.drawLine(198, 170, 235, 190);

}

if(nr\_erori==5){

g.drawOval(172, 120, 51, 51);

g.fillRect(195, 170, 3, 70);

g.drawLine(195, 170, 155, 190);

g.drawLine(198, 170, 235, 190);

g.drawLine(195, 240, 155, 270);

}

if(nr\_erori==6){

g.drawOval(172, 120, 51, 51);

g.fillRect(195, 170, 3, 70);

g.drawLine(195, 170, 155, 190);

g.drawLine(198, 170, 235, 190);

g.drawLine(195, 240, 155, 270);

g.drawLine(198, 240, 235, 270);

}

}

}

package p20;

import java.util.Scanner;

public class MyClass {

@SuppressWarnings("deprecation")

public static void main(String []args){

System.out.println("Introduceti optiune joc\nAveti la dispozitie 10 jocuri\nSelectati o valoare de la 0 la 9");

System.out.println("optiune=");

int opt;

Scanner scan=new Scanner(System.in);

opt=scan.nextInt();

Spanzuratoare a=new Spanzuratoare(opt);

scan.close();

a.show();

}

}

package p20;

import java.awt.BorderLayout;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import java.util.Scanner;

import javax.swing.JFrame;

@SuppressWarnings("serial")

public class Spanzuratoare extends JFrame {

draw joc;

char [][] cuvinte\_joc;

Spanzuratoare(int optiune){

setSize(500,500);

cuvinte\_joc =new char[10][];

this.setVisible(true);

char []cuv1={'b','i','l','i','a','r','d'};

cuvinte\_joc[1]=cuv1;

char []cuv2={'e','x','t','r','a','t','e','r','e','s','t','r','u'};

cuvinte\_joc[2]=cuv2;

char []cuv3={'t','e','l','e','n','c','i','c','l','o','p','e','d','i','e'};

cuvinte\_joc[3]=cuv3;

char []cuv4={'l','i','n','g','v','i','s','t','i','c'};

cuvinte\_joc[4]=cuv4;

char []cuv5={'a','s','t','r','o','n','a','u','t'};

cuvinte\_joc[5]=cuv5;

char []cuv6={'j','a','v','a','s','c','r','i','p','t'};

cuvinte\_joc[6]=cuv6;

char []cuv7={'a','m','s','t','e','r','d','a','m',};

cuvinte\_joc[7]=cuv7;

char []cuv8={'g','e','r','m','a','n','i','a'};

cuvinte\_joc[8]=cuv8;

char []cuv9={'p','a','r','a','d','i','g','m'};

cuvinte\_joc[9]=cuv9;

char []cuv0={'p','o','l','i','m','o','r','f','i','s','m'};

cuvinte\_joc[0]=cuv0;

joc=new draw(cuvinte\_joc[optiune]);

add(joc,BorderLayout.CENTER);

this.addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e){

dispose();

System.exit(0);

}

});

JOC();

}

void JOC(){

Scanner reader = new Scanner(System.in);

while(joc.Get\_erori()<6 && !joc.Get\_vizibil()){

System.out.print("Litera= ");

char c = reader.next().trim().charAt(0);

joc.Game(c);

}

reader.close();

if(joc.Get\_vizibil())

System.out.println("FELICITARI AI CASTIGAT");

else

System.out.println(" AI PIERDUT");

}

}

P21

package p21;

import java.awt.\*;

import java.awt.event.\*;

import java.io.File;

import java.io.FilenameFilter;

@SuppressWarnings("serial")

public class Notepad extends Frame implements ActionListener{

int ok=1;

MenuItem f1,f2,f3,f4,f5,f6,f7,for1,for2,e1,e2,e3,e4,e5,h1,h2,m1;

MenuBar bara\_meniu;

Menu edit,fisier,help,format,view;

FileDialog load,save;

TextArea text;

Notepad(String a){

super(a);

this.addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e){

dispose();

System.exit(0);

}

});

bara\_meniu=new MenuBar();

fisier=new Menu("File");

f1=new MenuItem("New");

fisier.add(f1);

f2=new MenuItem("Open");

fisier.add(f2);

f3=new MenuItem("Save");

fisier.add(f3);

f4=new MenuItem("Save as");

fisier.add(f4);

fisier.addSeparator();

f5=new MenuItem("Page Setup");

fisier.add(f5);

f6=new MenuItem("Print");

fisier.add(f6);

fisier.addSeparator();

f7=new MenuItem("Exit");

fisier.add(f7);

f1.addActionListener(this);

f2.addActionListener(this);

f3.addActionListener(this);

f4.addActionListener(this);

f5.addActionListener(this);

f6.addActionListener(this);

f7.addActionListener(this);

edit=new Menu("Edit");

e1=new MenuItem("Undo");

edit.add(e1);

edit.addSeparator();

e2=new MenuItem("Cut");

edit.add(e2);

e3=new MenuItem("Copy");

edit.add(e3);

e4=new MenuItem("Paste");

edit.add(e4);

e5=new MenuItem("Delete");

edit.add(e5);

format=new Menu("Format");

for1=new MenuItem("Word Wrap");

format.add(for1);

for2=new MenuItem("Font");

format.add(for2);

view=new Menu("View");

m1=new MenuItem("Status Bar");

view.add(m1);

help=new Menu("Help");

h1=new MenuItem("View Help");

help.add(h1);

help.addSeparator();

h2=new MenuItem("About Help");

help.add(h2);

bara\_meniu.add(fisier);

bara\_meniu.add(help);

bara\_meniu.add(view);

bara\_meniu.add(format);

bara\_meniu.add(edit);

setMenuBar(bara\_meniu);

text=new TextArea();//textul

add(text);

setSize(400,400);

}

@SuppressWarnings("deprecation")

public void actionPerformed(ActionEvent e) {

MenuItem mi=(MenuItem)e.getSource();

if(f7==mi){

System.exit(0);

}

if (mi==f2){

load=new FileDialog(this,"Alege fisierul",FileDialog.LOAD);

load.setDirectory(".");

load.setFilenameFilter(new FilenameFilter(){

public boolean accept(File dir, String nume) {

return (nume.endsWith(".txt"));

}

});

load.show();

}

if(mi==f4){

save=new FileDialog(this,"Salvare fisier",FileDialog.SAVE);

save.setDirectory(".");

save.setFilenameFilter(new FilenameFilter(){

@Override

public boolean accept(File dir, String name) {

// TODO Auto-generated method stub

return (name.endsWith(".txt"));

}

});

save.show();

}

}

}

package p21;

public class MyClass {

@SuppressWarnings("deprecation")

public static void main(String []args){

Notepad a=new Notepad("Notepad");

a.show();

}

}

P22

package p22;

import javax.swing.\*;

import java.awt.BorderLayout;

import java.awt.event.\*;

import java.io.File;

import javax.swing.JFrame;

@SuppressWarnings("serial")

public class Notepad extends JFrame implements ActionListener{

int ok=1;

JMenuItem f1,f2,f3,f4,f5,f6,f7,for1,for2,e1,e2,e3,e4,e5,h1,h2,m1;

JMenuBar bara\_meniu;

JMenu edit,fisier,help,format,view;

JFileChooser load,save;

JTextArea text;

Notepad(String a){

super(a);

setLayout(new BorderLayout());

this.addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e){

dispose();

System.exit(0);

}

});

bara\_meniu=new JMenuBar();

fisier=new JMenu("File");

f1=new JMenuItem("New");

fisier.add(f1);

f2=new JMenuItem("Open");

fisier.add(f2);

f3=new JMenuItem("Save");

fisier.add(f3);

f4=new JMenuItem("Save as");

fisier.add(f4);

fisier.addSeparator();

f5=new JMenuItem("Page Setup");

fisier.add(f5);

f6=new JMenuItem("Print");

fisier.add(f6);

fisier.addSeparator();

f7=new JMenuItem("Exit");

fisier.add(f7);

f1.addActionListener(this);

f2.addActionListener(this);

f3.addActionListener(this);

f4.addActionListener(this);

f5.addActionListener(this);

f6.addActionListener(this);

f7.addActionListener(this);

edit=new JMenu("Edit");

e1=new JMenuItem("Undo");

edit.add(e1);

edit.addSeparator();

e2=new JMenuItem("Cut");

edit.add(e2);

e3=new JMenuItem("Copy");

edit.add(e3);

e4=new JMenuItem("Paste");

edit.add(e4);

e5=new JMenuItem("Delete");

edit.add(e5);

format=new JMenu("Format");

for1=new JMenuItem("Word Wrap");

format.add(for1);

for2=new JMenuItem("Font");

format.add(for2);

view=new JMenu("View");

m1=new JMenuItem("Status Bar");

view.add(m1);

help=new JMenu("Help");

h1=new JMenuItem("View Help");

help.add(h1);

help.addSeparator();

h2=new JMenuItem("About Help");

help.add(h2);

bara\_meniu.add(fisier);

bara\_meniu.add(help);

bara\_meniu.add(view);

bara\_meniu.add(format);

bara\_meniu.add(edit);

bara\_meniu.setSize(20, 20);

add(bara\_meniu,BorderLayout.NORTH);

text=new JTextArea();//textul

add(text);

setSize(400,400);

}

@SuppressWarnings("deprecation")

public void actionPerformed(ActionEvent e) {

JMenuItem mi=(JMenuItem)e.getSource();

if(f7==mi){

System.exit(0);

}

if (mi==f2){

load=new JFileChooser();

load.setCurrentDirectory(new File(System.getProperty("user.home")));

int result=load.showOpenDialog(this);

if (result == JFileChooser.APPROVE\_OPTION) {

File selectedFile = load.getSelectedFile();

System.out.println("Selected file: " + selectedFile.getAbsolutePath());

}

load.show();

}

}

}

package p22;

import p22.Notepad;

public class MyClass {

@SuppressWarnings("deprecation")

public static void main(String []args){

Notepad a=new Notepad("Notepad");

a.show();

}

}

P23

package p23;

import java.awt.\*;

@SuppressWarnings("serial")

public class draw extends Canvas{

String name;

draw(String a){

name=a;

}

public void paint(Graphics g){

g.setColor(Color.blue);

if(name.equals("patrat")){

g.fillRect(20, 20, 100, 100);

}

if(name.equals("cerc")){

g.setColor(Color.red);

g.fillOval(300, 20, 120,120);

}

if(name.equals("triunghi")){

g.setColor(Color.green);

int []x=new int[3];

int []y=new int[3];

x[0]=70;x[1]=20;x[2]=120;

y[0]=300;y[1]=420;y[2]=420;

g.fillPolygon(x,y, 3);

}

}

}

package p23;

import java.awt.\*;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

@SuppressWarnings("serial")

public class Fabrica\_de\_Nave extends Frame implements Factory{

draw a;

Frame f;

public Fabrica\_de\_Nave(){

f=new Frame();

f.setSize(500,500);

f.addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e){

dispose();

System.exit(0);

}

});

}

public Nava make(String nume){

a=new draw(nume);

if(nume.equals("patrat")){

return new Patrat(f,a);

}

else if(nume.equals("cerc")){

return new Cerc(f,a);

}

else if(nume.equals("triunghi")){

return new Triunghi(f,a);

}

else{

System.out.println("NAVA INEXISTENTA");

return null;

}

}

}

interface Factory{

public Nava make(String nume);

}

interface Nava{

public void afisare();

}

class Patrat implements Nava{

draw a;

Frame f;

public Patrat(Frame fr,draw ar){

f=fr;

a=ar;

}

@SuppressWarnings("deprecation")

public void afisare() {

f.add(a);

f.show();

}

}

class Triunghi implements Nava{

draw a;

Frame f;

public Triunghi(Frame fr,draw ar){

f=fr;

a=ar;

}

@SuppressWarnings("deprecation")

public void afisare() {

f.add(a);

f.show();

}

}

class Cerc implements Nava{

draw a;

Frame f;

public Cerc(Frame fr,draw ar){

f=fr;

a=ar;

}

@SuppressWarnings("deprecation")

public void afisare() {

f.add(a);

f.show();

}

}

package p23;

public class MyClass {

public static void main(String []args){

Fabrica\_de\_Nave a=new Fabrica\_de\_Nave();

Nava []b=new Nava[3];

b[0]=a.make("patrat");

b[1]=a.make("triunghi");

b[2]=a.make("cerc");

b[0].afisare();

}

}

P24

package p24;

import java.awt.Color;

import java.awt.Graphics;

import javax.swing.JPanel;

@SuppressWarnings("serial")

public class draw extends JPanel{

String name;

draw(String a){

name=a;

}

public void paint(Graphics g){

g.setColor(Color.blue);

if(name.equals("patrat")){

g.fillRect(20, 20, 100, 100);

}

if(name.equals("cerc")){

g.setColor(Color.red);

g.fillOval(300, 20, 120,120);

}

if(name.equals("triunghi")){

g.setColor(Color.green);

int []x=new int[3];

int []y=new int[3];

x[0]=70;x[1]=20;x[2]=120;

y[0]=300;y[1]=420;y[2]=420;

g.fillPolygon(x,y, 3);

}

}

}

package p24;

import java.awt.\*;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

@SuppressWarnings("serial")

public class Fabrica\_de\_Nave extends Frame implements Factory{

draw a;

Frame f;

public Fabrica\_de\_Nave(){

f=new Frame();

f.setSize(500,500);

f.addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e){

dispose();

System.exit(0);

}

});

}

public Nava make(String nume){

a=new draw(nume);

if(nume.equals("patrat")){

return new Patrat(f,a);

}

else if(nume.equals("cerc")){

return new Cerc(f,a);

}

else if(nume.equals("triunghi")){

return new Triunghi(f,a);

}

else{

System.out.println("NAVA INEXISTENTA");

return null;

}

}

}

interface Factory{

public Nava make(String nume);

}

interface Nava{

public void afisare();

}

class Patrat implements Nava{

draw a;

Frame f;

public Patrat(Frame fr,draw ar){

f=fr;

a=ar;

}

@SuppressWarnings("deprecation")

public void afisare() {

f.add(a);

f.show();

}

}

class Triunghi implements Nava{

draw a;

Frame f;

public Triunghi(Frame fr,draw ar){

f=fr;

a=ar;

}

@SuppressWarnings("deprecation")

public void afisare() {

f.add(a);

f.show();

}

}

class Cerc implements Nava{

draw a;

Frame f;

public Cerc(Frame fr,draw ar){

f=fr;

a=ar;

}

@SuppressWarnings("deprecation")

public void afisare() {

f.add(a);

f.show();

}

}

package p24;

import p24.Fabrica\_de\_Nave;

import p24.Nava;

public class MyClass {

public static void main(String []args){

Fabrica\_de\_Nave a=new Fabrica\_de\_Nave();

Nava []b=new Nava[3];

b[0]=a.make("patrat");

b[1]=a.make("triunghi");

b[2]=a.make("cerc");

b[0].afisare();

}

}

P25

package p25;

import java.awt.\*;

@SuppressWarnings("serial")

public class draw extends Canvas{

String name;

draw(String nume){

name=nume;

}

public void paint(Graphics g){

g.setColor(Color.blue);

g.fillRect(20, 20, 120, 120);

if(name.equals("mitraliera")){

g.setColor(Color.black);

g.fillRect(170, 80, 100, 10);

}

if(name.equals("blast")){

g.setColor(Color.red);

g.fillOval(210, 40, 80, 80);

g.setColor(Color.yellow);

g.drawOval(210, 40, 81, 81);

g.drawOval(210, 40, 82, 82);

g.drawOval(210, 40, 83, 83);

g.drawOval(210, 40, 84, 84);

g.drawOval(210, 40, 85, 85);

g.drawOval(210, 40, 86, 86);

g.drawOval(210, 40, 87, 87);

}

if(name.equals("rachete")){

g.setColor(Color.green);

g.fillRect(170, 65, 100, 10);

g.fillRect(170, 80, 100, 10);

g.fillRect(170, 95, 100, 10);

}

}

}

package p25;

import java.awt.Frame;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import p25.draw;

@SuppressWarnings("serial")

public class Fabrica\_de\_foc extends Frame implements Factory2{

draw a;

Frame f;

public Fabrica\_de\_foc(){

f=new Frame();

f.setSize(400,400);

f.addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e){

dispose();

System.exit(0);

}

});

}

public Fire make(String nume){

a=new draw(nume);

if(nume.equals("mitraliera")){

return new Mitraliera(f,a);

}

else if(nume.equals("blast")){

return new Blast(f,a);

}

else if(nume.equals("rachete")){

return new Racheta(f,a);

}

else{

System.out.println("NAVA INEXISTENTA");

return null;

}

}

}

interface Factory2{

Fire make(String nume);

}

interface Fire{

public void afisare();

}

class Mitraliera implements Fire{

draw a;

Frame f;

public Mitraliera(Frame fr,draw ar){

f=fr;

a=ar;

}

@SuppressWarnings("deprecation")

public void afisare() {

f.add(a);

f.show();

}

}

class Racheta implements Fire{

draw a;

Frame f;

public Racheta(Frame fr,draw ar){

f=fr;

a=ar;

}

@SuppressWarnings("deprecation")

public void afisare() {

f.add(a);

f.show();

}

}

class Blast implements Fire{

draw a;

Frame f;

public Blast(Frame fr,draw ar){

f=fr;

a=ar;

}

@SuppressWarnings("deprecation")

public void afisare() {

f.add(a);

f.show();

}

}

package p25;

import p25.Fabrica\_de\_foc;

public class MyClass {

public static void main(String []args){

Fabrica\_de\_foc a=new Fabrica\_de\_foc();

Fire []b=new Fire[3];

b[0]=a.make("mitraliera");

b[1]=a.make("rachete");

b[2]=a.make("blast");

b[0].afisare();

}

}

P26

package p26;

public class Adapter\_mancare implements Mananca{

ManancaOrice m;

public Adapter\_mancare(String tip){

if(tip.equalsIgnoreCase("bunica")){

m=new Bunica();

}

else if(tip.equalsIgnoreCase("pisica")){

m=new Pisica();

}

}

public void mananca(String tip, String mancare) {

if(tip.equalsIgnoreCase("bunica")){

m.manancaBunica(mancare);

}

else if(tip.equalsIgnoreCase("pisica")){

m.manancaPisica(mancare);

}

}

}

package p26;

public class Bunica implements ManancaOrice{

@Override

public void manancaPisica(String mancare) {

// TODO Auto-generated method stub

}

@Override

public void manancaBunica(String mancare) {

System.out.println("Bunica mananca "+mancare);

}

}

package p26;

public class Mamifer implements Mananca{

Adapter\_mancare am;

Mamifer(){};

@Override

public void mananca(String tip, String mancare) {

if(tip.equalsIgnoreCase("porc")){

System.out.println("Porcul mananca "+mancare);

}

else if(tip.equalsIgnoreCase("bunica")||tip.equalsIgnoreCase("pisica")){

am=new Adapter\_mancare(tip);

am.mananca(tip, mancare);

}

else {

System.out.println("Nu avem informatii despre ce mananca "+tip);

}

}

}

package p26;

public interface Mananca {

public void mananca(String tip,String mancare);

}

interface ManancaOrice{

public void manancaPisica(String mancare);

public void manancaBunica(String mancare);

}

package p26;

public class MyAdapter {

public static void main(String []args){

Mamifer m=new Mamifer();

m.mananca("bunica", "legume");

m.mananca("porc", "alune");

m.mananca("broasca", "mure");

m.mananca("pisica", "soareci");

m.mananca("bunica","spanac");

}

}

package p26;

public class Pisica implements ManancaOrice{

@Override

public void manancaPisica(String mancare) {

System.out.println("Pisica mananca "+mancare);

}

@Override

public void manancaBunica(String mancare) {

// TODO Auto-generated method stub

}

}

P30

package p30;

import java.util.Scanner;

public class FCFS

{

public static void main(String[] args)

{

Thread id[]=new Thread[20];

int btime[]=new int[20];

int wtime[]=new int[20];

int total=0;

float avg;

Scanner sn = new Scanner(System.in);

System.out.print("\nEnter the number of Threads: ");

int n = sn.nextInt();

for (int i=0;i<n;i++) {

System.out.println();

id[i]=new Thread("Thread"+i);

System.out.print("Enter the burst time of threads "+(i+1)+" : ");

btime[i]=sn.nextInt();

}

wtime[0]=0;

for (int i=1;i<n;i++) {

wtime[i]=wtime[i-1]+btime[i-1];

total=total+wtime[i];

}

avg=(float)total/n;

System.out.println("\nProcess\_ID\tBurst\_time\tWait\_time");

for(int i=0;i<n;i++)

{

System.out.println(id[i].getName()+"\t\t"+btime[i]+"\t\t"+wtime[i]);

}

System.out.println("\nTotal wait time: "+total+"\nAverage wait time: "+avg);

sn.close();

}

}

P31

package p31;

import java.util.\*;

public class Buffer {

LinkedList<String> Fifo;

Buffer(){

Fifo = new LinkedList<String>();

Fifo.add("First mesage");

}

void put(String msg){

Fifo.add(msg);

}

String get(){

return Fifo.getLast();

}

void incrementIndex(){

Fifo.removeFirst();

}

}

package p31;

import java.util.\*;

public class MainClass {

public static void main(String[] args){

ThreadUsers trd1, trd2, trd3;

Buffer buff = new Buffer();

trd1 = new ThreadUsers(buff);

trd2 = new ThreadUsers(buff);

trd3 = new ThreadUsers(buff);

trd1.start();

trd2.start();

trd3.start();

while(true){

if(trd1.getSeen() && trd2.getSeen() && trd3.getSeen()){

buff.incrementIndex();

trd1.resetSeen();

trd2.resetSeen();

trd3.resetSeen();

}

System.out.println("Introduceti mesaj");

Scanner scan = new Scanner(System.in);

String s = scan.nextLine();

buff.put(s);

}

}

}

package p31;

import java.util.Scanner;

public class ThreadUsers extends Thread{

boolean seen;

String msg;

Buffer buf;

ThreadUsers(Buffer buffer){

seen = false;

buf = buffer;

msg = new String();

}

public void run(){

while(true){

if(seen == false){

msg = buf.get();

System.out.println("Am receptionat mesajul: "+msg);

seen = true;

}

try {

sleep(1000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

public boolean getSeen(){

return seen;

}

void resetSeen(){

seen = false;

}

}

P32

package p32;

public class MyClass {

public static void main(String []args){

SumThread a=new SumThread(8);

MulThread b=new MulThread(8);

SquareThread c=new SquareThread(8);

a.start();

b.start();

c.start();

System.out.println("Suma= "+a.getSum());

System.out.println("Factorial= "+b.get\_Fact());

System.out.println("Patrat= "+c.getSquare());

System.out.println("Se observa ca la rulari repetate ale programului\n apar rezultate diferite,de aici rezultand hazardul Thread`urilor");

}

}

package p32;

public class SumThread extends Thread{

int nr;

int sum = 0;

public SumThread(int n){

this.nr = n;

}

public void run(){

for(int i=0 ; i<=nr ; i++){

sum += i;

}

}

public int getSum(){

return sum;

}

}

class MulThread extends Thread{

int nr;

int mul = 1;

public MulThread(int n){

this.nr = n;

}

public void run(){

for(int i=1 ; i<=nr ; i++){

mul \*= i;

}

}

public int get\_Fact(){

return mul;

}

}

class SquareThread extends Thread{

int nr;

int square;

public SquareThread(int n){

this.nr = n;

}

public void run(){

square = nr\*nr;

}

public int getSquare(){

return square;

}

}

P33

package p33;

public class tred implements Runnable{

private int strt,stp;

private int suma;

Thread runner;

tred(String name){

strt=0;

stp=10;

suma=0;

runner=new Thread(this,name);

System.out.println("Threadul "+name+" a pornit");

runner.start();

};

tred(int x,int y,String name){

strt=x;

stp=y;

suma=0;

runner=new Thread(this,name);

System.out.println("Threadul "+name+" a pornit");

runner.start();

}

public int get\_suma(){

return suma;

}

public void run() {

// TODO Auto-generated method stub

for(int i=strt;i<=stp;i++){

suma=suma+i;

}

}

}

package p33;

import java.util.Scanner;

public class MyClass {

@SuppressWarnings("static-access")

public static void main(String []args){

int n,total=0;

System.out.print("Introdu numar: ");

Scanner scan=new Scanner(System.in);

n=scan.nextInt();

tred a=new tred(1,n/4,"Thread1");

tred b=new tred(n/4+1,n/2,"Thread2");

tred c=new tred(n/2+1,3\*n/4,"Thread3");

tred d=new tred(3\*n/4+1,n,"Thread4");

scan.close();

try {

//delay for one second

Thread.currentThread().sleep(4000);

} catch (InterruptedException e) {

}

total=total+d.get\_suma()+a.get\_suma()+b.get\_suma()+c.get\_suma();

System.out.println("Suma este "+total);

}

}

P34

package p34;

public class MyThread extends Thread {

int n;

int s=0;

public MyThread(int a){

n=a;

}

public void run(){

for(int i=0;i<=n;i++){

s=s+i;

}

}

public int get\_S(){

return s;

}

}

package p34;

import java.util.\*;

public class MyClass {

public static void main(String []args){

LinkedList<Integer> n=new LinkedList<Integer>();

n.add(4);

n.add(6);

n.add(10);

n.add(9);

MyThread a,b,c,d;

a=new MyThread(n.getFirst());

n.remove();

b=new MyThread(n.getFirst());

n.remove();

c=new MyThread(n.getFirst());

n.remove();

d=new MyThread(n.getFirst());

a.start();

b.start();

c.start();

d.start();

try{

Thread.sleep(2000);

}catch(InterruptedException e){

e.printStackTrace();

}

System.out.println("a="+a.get\_S());

System.out.println("b="+b.get\_S());

System.out.println("c="+c.get\_S());

System.out.println("d="+d.get\_S());

}

}

P35

package p35;

import java.io.\*;

public class Multiply extends Thread{

DataOutputStream out;

private int alfa;

int []v;

Multiply(int a,int []b,DataOutputStream o){

alfa=a;

v=b;

out=o;

}

int Get\_lenght(){

return v.length;

}

public void run(){

for(int i=0;i<v.length;i++){

v[i]=v[i]\*alfa;

try{

out.writeInt(v[i]);

}catch(IOException e){

e.printStackTrace();

}

try {

sleep(1000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

}

class Sort extends Thread{

DataOutputStream out;

DataInputStream in;

int nr\_elem\_din\_vector;

int []v;//pun elementele din buffer intr-un vector,sortez vectorul apoi il bag iar in buffer

public Sort(DataInputStream i,DataOutputStream o,int n){

in=i;

out=o;

nr\_elem\_din\_vector=n;

v=new int[n];

}

public void run(){

try {

sleep(2000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

for(int i=0;i<v.length;i++){

try{

v[i]=in.readInt();

}catch(IOException e){

e.printStackTrace();

}

}

boolean ok=true;

while(ok){

ok=false;

for(int i=0;i<v.length-1;i++){

if(v[i]>v[i+1]){

ok=true;

int aux=v[i];

v[i]=v[i+1];

v[i+1]=aux;

}

}

}

for(int i=0;i<v.length;i++){

try{

out.writeInt(v[i]);

}catch(IOException e){

e.printStackTrace();

}

}

}

}

class Afisare extends Thread{

int nr;

int []value;

DataInputStream in;

public Afisare(int n,DataInputStream i){

in=i;

nr=n;

value=new int[nr];

}

public void run(){

try {

sleep(4000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

for(int i=0;i<nr;i++){

try {

value[i]=in.readInt();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

for(int i=0;i<nr-1;i+=2)

System.out.print("("+value[i]+","+value[i+1]+") ");

}

}

package p35;

import java.io.\*;

public class MyClass {

public static void main(String []args)throws IOException{

PipedOutputStream Pout=new PipedOutputStream();

PipedInputStream Pin=new PipedInputStream(Pout);

DataOutputStream out =new DataOutputStream(Pout);

DataInputStream in =new DataInputStream(Pin);

int[]v=new int [4];

for(int i=0;i<4;i++){

v[i]=4-i;

}

Multiply m=new Multiply(2, v, out);

Sort s=new Sort(in, out, 4);

Afisare a=new Afisare(4, in);

m.start();

s.start();

a.start();

}

}

P39

from logic import \*

a = eval(raw\_input('A= '))

b = eval(raw\_input('B= '))

c = eval(raw\_input('C= '))

not1 = Not("N1")

not2 = Not("N2")

and1 = And("A1")

and2 = And("A2")

and3 = And("A3")

and4 = And("A4")

and5 = And("A5")

not1.B.connect(and2.B)

not2.B.connect(and2.A)

not2.B.connect(and3.B)

not1.B.connect(and4.A)

and4.C.connect(and5.A)

or1 = Or("Or1")

or2 = Or("Or2")

or3 = Or("Or3")

and1.C.connect(or1.A)

and2.C.connect(or1.B)

and3.C.connect(or2.A)

and5.C.connect(or2.B)

or1.C.connect(or3.A)

or2.C.connect(or3.B)

or3.C.monitor = 1

not1.A.set(a)

not2.A.set(b)

and1.A.set(b)

and1.B.set(c)

and3.A.set(a)

and4.B.set(b)

and5.B.set(c)

P40

from logic import \*

alpha = eval(raw\_input("A = "))

betta = eval(raw\_input("B = "))

gamma = eval(raw\_input("C = "))

xor2 = Xor("XorGate, Pin")

xor1 = Xor("Xor1")

and1 = And("And1")

and2 = And("And2")

or1 = Or("OrGate, Pin")

xor1.C.connect(xor2.A)

xor1.C.connect(and2.A)

and2.C.connect(or1.A)

and1.C.connect(or1.B)

xor2.C.monitor = 1

or1.C.monitor = 1

xor1.A.set(alpha)

xor1.B.set(betta)

and1.A.set(alpha)

and1.B.set(betta)

xor2.B.set(gamma)

and2.B.set(gamma)

P41

from logic import \*

A = eval(raw\_input("A = "))

B = eval(raw\_input("B = "))

C = eval(raw\_input("C = "))

and1 = And("A1")

and2 = And("A2")

and3 = And("A3")

not1 = Not("N1")

not2 = Not("N2")

or1 = Or("O1")

xor1 = Xor("X1")

and1.C.connect(and3.A)

or1.C.connect(not1.A)

not1.B.connect(not2.A)

not1.B.connect(and2.A)

and2.C.connect(xor1.B)

not2.B.connect(and3.B)

and3.C.connect(xor1.A)

xor1.C.monitor = 1

and1.A.set(A)

and1.B.set(B)

or1.A.set(A)

or1.B.set(C)

and2.B.set(B)

P42

from logic import \*

def main():

#definire celor 5 stari

state1 = "0 cents deposited"

state2 = "5 cents deposited"

state3 = "10 cents deposited"

state4 = "15 cents deposited"

state5 = "20 cents or more deposited"

#succesiunea starilor in functie de nickel sau dime

nickel = [state1, state2, state3, state4, state5, state2]

dime1 = [state1, state3, state5, state3]

dime2 = [state2, state4, state5, state3, state5]

# state3 si state5 pe care le-am declarat in dime2 sunt redundante

# la felul in care am implementat(pentru ca, atunci cand ajung intr-o

# stare care se gaseste in dime2 si nu in dime1, la urmatoarea iteratie

# din bucla while se verifica mai intai in dime1

#fisierul din care citesc starea initiala

fisier = open("fisier.txt", "r")

stareaInitiala = fisier.read()

currentState = stareaInitiala

print "Starea curenta: ", currentState # afisarea starii curente

while True:

nr = input("0-nickel; 1-dime; 2-iesire din program\n")

if nr == 2:

break

if nr == 0: # in nickel sunt toate starile(dupa schema)

aux = nickel.index(currentState)

currentState = nickel[aux+1] # trec la starea urmatoare

if nr == 1:

if currentState in dime1:

aux = dime1.index(currentState)

currentState = dime1[aux+1]

else: # dupa schema, daca nu e in dime1, cu siguranta e in dime2

aux = dime2.index(currentState)

currentState = dime2[aux+1]

print "Starea curenta: ", currentState # afisarea starii curente

main()

P43

from Crypto.Hash import MD5

FileName = raw\_input("Enter file name: ")

readFile = open(FileName,'r')

writeFileName = FileName.partition(".")

writeFileName = writeFileName[0]+".txt"

writeFile = open(writeFileName,'w+')

for line in readFile:

hashLine = MD5.new(line).hexdigest()

writeFile.write(hashLine)

P44

# TCP Client Code

host="127.0.0.1" # Seteaza variabila host

port=4446 # Seteaza variabila port 4446

from socket import \* # Importa socket module

s=socket(AF\_INET, SOCK\_STREAM) # Creaza socket-ul

s.connect((host,port)) # Se conecteaza la server

while 1:

msg=s.recv(1024)

print "Mesajul primit : " + msg

data=raw\_input("Introduceti mesajul : ")

if not data: break

s.send(data)

s.close() # Inchide socket-ul

# TCP Server Code

host="127.0.0.1" # Seteaza variabila host

port=4446 # Seteaza variabila port 4446

from socket import \* # Importa socket module

s=socket(AF\_INET, SOCK\_STREAM)

s.bind((host,port))

s.listen(1)

print "Se asteapta conexiunea.. "

q,addr=s.accept()

while 1:

data=raw\_input("Introduceti mesajul : ")

q.send(data)

msg=q.recv(1024)

if not msg: break

print "Mesajul primit : " + msg

s.close()

P45

%reprezentare dreptunghi

dreptunghi(point(A,B),point(C,B),point(A,D),point(C,D)).

%rep cerc

cerc(point(\_,\_),\_).

P46

big(bear).

big(elephant).

small(cat).

brown(bear).

black(cat).

gray(elephant).

dark(Z):-black(Z).

dark(Z):-brown(Z).

%big(x),dark(x):

%big(bear):-dark(bear)

% dark(bear):-black(bear) False

% :-brown(bear) True

% dark(x),big(x):

% dark(bear):-black(bear) False

% :-brown(bear) True

% Se pare ca big(x),dark(x) are o operatie in plus

P47

%http://inf.ucv.ro/~rstoean/courses/pnp/c2.pdf

%https://sites.google.com/site/prologsite/prolog-problems/1/solutions-1

initial(00).

final(11).

trans(00,1,01).

trans(01,1,11).

trans(00,0,10).

trans(01,0,10).

trans(10,1,01).

trans(11,1,01).

trans(11,10,0).

accepts(State,[]):-final(State).

accepts(State,[X|Rest]):-trans(State,X,State1),accepts(State1,Rest).

%stabilim starea initiala si cea finala

%punem elementele de tranzitie dintre stari

%

% prima clauza accepts:situl gol [] este acceptat din Starea State daca

% State este o stare finala,11 in cazul nostru

% a doua clauza accepts:un sir nevid este acceptat din starea State daca

% primul element din lista determina trecerea din State in State1 iar

% restul elementelor sunt acceptate din starea State1.

%

% ex de rulare: accept(11,[]). (Va da true deoarece 11 e stare finala

% accepts(00,[X1,X2,X3]).(va da X1=x2=X3=1..vezi figuara

P48

%afisarea elementelor unei liste una cate una:conditia initiala de stop

% lista vida []:tab(3)=3 taburi:Prim=head:apel recursiv:afis([1,2,3]).

afis([]).

afis([Prim|Rest]):-write(Prim),tab(3),afis(Rest).

%ultimul element dintr-o lista

% clauza de stop ca restul listei sa fie 0,daca nu e 0 atunci eliminam

% capul acesteia

my\_last(X,[X|[]]).

my\_last(X,[\_|Rest]):-my\_last(X,Rest).

%ex rulare: my\_last(X,[1,2,3,4,5]). (va da X=5)

P49

%eliminarea duplicatelor

%daca am un element deja existent elimin restul

compress([],[]).

compress([X],[X]).

compress([X,X|Xs],Zs) :- compress([X|Xs],Zs).

compress([X,Y|Xs],[X|Zs]) :- X \= Y, compress([Y|Xs],Zs).

% clauza 3:in prima faza daca am 2 elemente consec egale apel recursiv

%clauza 4:daca am 2 elemente consecutive diferite in lista

% Ys:X si Y si iar in lista Zs il am pe X,atunci il trec pe Y in Zs ex

% de r:compress([a,a,a,a,b,c,c,a,a,d,e,e,e,e],X). (va

% afisa:X=[a,b,c,a,d,e]

P50

drop(L1,N,L2) :- drop(L1,N,L2,N).

drop([],\_,[],\_).

drop([\_|Xs],N,Ys,1) :- drop(Xs,N,Ys,N).

drop([X|Xs],N,[X|Ys],K) :- K > 1, K1 is K - 1, drop(Xs,N,Ys,K1).

% drop(L1,N,L2):L2 se obtine din L1 prin eliminarea elementelor de pe poz

% n

% drop(L1,N,L2,K):L2 se obtine din L1 prin adaugarea primelor k-1

% elemente din L1