

1.Обяснете понятието полиморфизъм.

2.Напишете фрагмент на програмния език Java, съдържащ блок **try- catch**. Ако стойността на въведените години в променливата **age** са число по- малко от **18**, се хвърля изключение **Exception**. Изключението трябва да е инстанция на класа **Exception** и при хващането му трябва да се отпечата: **java.lang.ArithmeticException: Достъпа е забранен под 18 години**.

3.

```
public class TestException{

    public static void throwit(){

        System.out.print("изключение");

        throw new IndexOutOfBoundsException();

    }

    public static void main(String[] args){

        try{

            System.out.print("привет");

            throwit();

        }

        catch(Exception e){

            System.out.print("хванато ");

        }

        finally{

            System.out.print("накрая изключение");

        }

        System.out.println("и след изключение");

    }

}
```

а)привет изключение хванато

б)грешка при компилация

в)привет изключение IndexOutOfBoundsException накрая изключение и след изключение

г) привет изключение хванато накрая икключение и след изключение

4.Как се нарича класа от JavaFX, които представя целия прозорец на приложението.

a)javafx.scene.Scene

b)javafx.stage.Stage

c)javafx.scene.layout.StackPane

5. Към готовия код даден по- долу за създаване на списък със студенти. Добавете поле за името на дисциплината за която имат съответната оценка.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;

public class InterfaceStudent extends JFrame{
    private static final long serialVersionUID = 1L;
    protected JButton ad;
    protected JTextField tf[]= new JTextField[2];
    protected JTextArea slist;

    protected JPanel contr,plst;
    protected AdSt adSt;
    protected ArrayList<Student> prs=new ArrayList<Student>(10);
    InterfaceStudent(int x, int y, int ln, int ht){
        this.setLayout(new BorderLayout());
        this.setBounds(x, y, ln, ht);
        this.setVisible(true);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setTitle("Студенти");
        slist = new JTextArea(5, 10);
        slist.setEditable(false);
        contr = new JPanel(new FlowLayout());
        plst = new JPanel(new FlowLayout());
        ad = new JButton("добавете");
        contr.add(ad);
        tf[0]= new JTextField("име",10);
        tf[1]= new JTextField("оценка",4);
        contr.add(tf[0]);
        contr.add(tf[1]);
        ad.addActionListener(adSt=new AdSt());
```

```

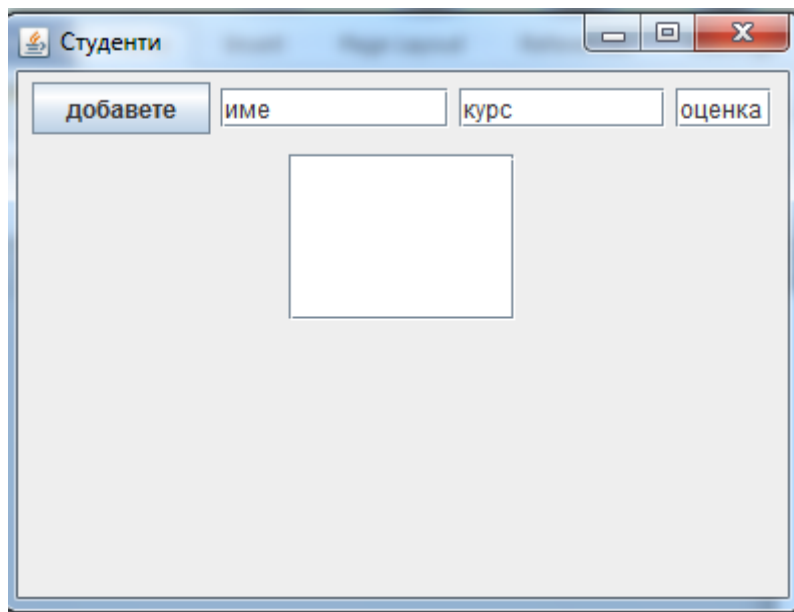
tf[1].addActionListener(adSt);
plst.add(new JScrollPane(slist));
add("North",contr);
add("Center",plst);
revalidate();
}
class AdSt implements ActionListener{
    public void actionPerformed(ActionEvent e ){
        Student s;
        int nt;
        String n=tf[0].getText();
        try{
            nt=Integer.parseInt(tf[1].getText());
        }
        catch(NumberFormatException ex){
            tf[1].setText("note?");
            return;
        }
        prs.add(s=new Student(n,nt));
        tf[0].setText("име");tf[1].setText("оценка");
        slist.append(s+"\n");
        slist.setCaretPosition(slist.getDocument().getLength());
        revalidate();
    }
}

public static void main(String [] arg){
    new InterfaceStudent(20,20,400,300);
}

import java.io.*;
class Student{
    String name; int note;
    Student(String name, int note){this.name = name;this.note = note;}
    public String toString(){
        return name+""+note;
    }
}

```

Интерфейса за добавяне на студентите трябва да изглежда така:



Задача:

1.Проектирайте Java приложение, което въвежда данни за студент, добавя студента към студентска група и реализира търсене на студент по успех или име на студент.

1.Дефинирайте клас **Student**, който съдържа полета за : **име(String)** и **успех(double)** с **private** достъп. Напишете:

а)конструктор с два параметъра за инициализиране на полетата на класа;

б)реализирайте методите за достъп **setId(long id)/getId()**, които установяват/връщат успеха на съответния студент;

в)предефиниран метод **toString()**, който връща информация за студент във формата **име , успех**.

2. Дефинирайте клас за фрейма с име **Person**, който реализира клас **Student**. Клас **Person** ще обработва “събитие в действие” при натискане на бутон или избор на елемент от комбиниран бутон за избор.

Класът съдържа **private** полета, които включват:

-група от студенти като динамичен списък (колекция **java.util.ArrayList**);

-две текстови полета със съответни етикети пред тях **Име** и **Успех**;

-текстова област с плъзгачи за извеждане на резултати;

-бутон **Добави** (добавя данни за студент към групата) с етикет пред него **Добави студент**;

-комбиниран бутон за избор, съдържащ списък от успеха на студентите в групата, с етикет пред него **Студент с максимален успех**

а)декларирайте конструктор без параметри, който:

-създава прозорец с надпис **Студенти**;

-установява завършване на приложението при затваряне на прозореца;

-създава компонентите на ГПИ(графичен потребителски интерфейс);

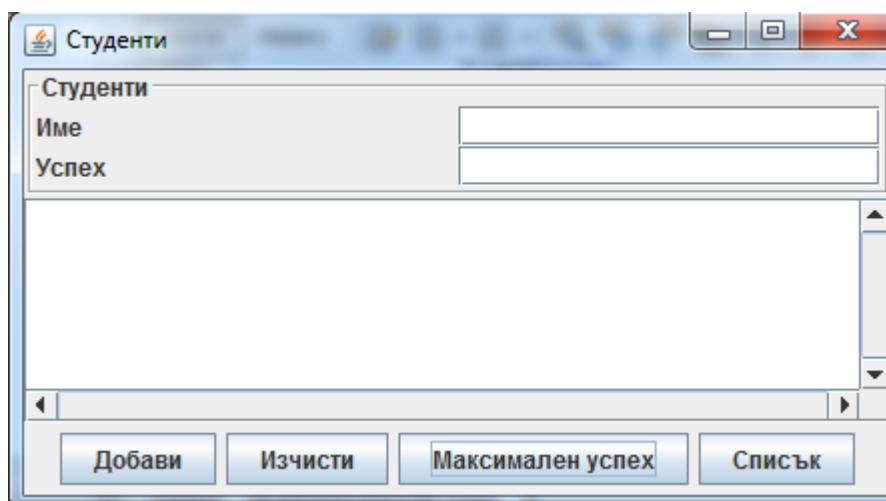
-добавя компонентите от ГПИ към фрейма;

-добавя слушатели на “събитие с действие” при натискане на бутон **Добави** и избор на елемент от комбиниран бутон за избор;

-създава празен динамичен списък;

-установява размера на фрейма 450x250;

б)реализирайте подходящия метод, който се извиква при вдигане на “събитие с действие”:



Добави

ако се хвърли изключение при изпълнение на

създава нов студент с въведените данни от текстовите полета

добавя студента в края на групата

добавя успеха на студента в списъка на комбинирания бутон за избор

изчиства текстовата област за резултатите

осъществява на търсене на успеха на студента при въвеждане на името на студента в полето за въвеждане на име на студента

добавя текста **Добавен студент..** в текстовата област изчиства текстовите полета обработва изключението добавя текста **Грешка** при въвеждане на данните в текстовата област

- търсене по максимален среден успех

търсене на максимален успех<- избран елемент от комбинирания бутон за избор

изчиства текстовата област за резултатите за всеки студент от групата повтаря

ако максималният успех студент е по- малък от средния успех на текущия студент

добавя текста **Намерен студент...** в текстовата област и изход

Изпит	
Тест	20
Задача 1	30
Задача 2	50

Оценка	
40-53	Среден 3
54-67	Добър 4
68-80	Много добър 5
81-100	Отличен 6