INCOMPLETE DOCUMENT

code and title for 11 to 24 accurate,
rest INCOMPLETE including the aim

Rexiel Scarlet

2-10-2024

Contents

1 Fibonacci Series

1.1 Aim

- Write a program to print the first n terms of the Fibonacci series

1.2 Code

```java
package org.projects.prog1;
import java.util.Scanner;

public class Fibo {
  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the number of terms");
    int limit = sc.nextInt();

    for (int i = 0; i < limit; i++) {
      System.out.print(fibo(i) + " ");
    }
  }

  static int fibo(int num) {
    return (num <= 1)? num : fibo(num - 1) + fibo (num - 2);
  }
}
```

1.3 Output

2 Operations On Complex Numbers

2.1 Aim

- Write a program to create a class Complex to store complex numbers
- Complex has two fields to store the real and imaginary parts
- create static functions to add and multiply two complex numbers
- create a member function to display the complex number

2.2 Code

```
package org.projects.prog2;
import java.util.Scanner;

public class Complex {
  int real, img;

  Complex(int real, int img) {
    this.real = real;
    this.img = img;
  }

  void display() {
    // some formating to correctly display complex numbers
    String fmt = String.format(
      "%d %c %di",
      real,
      (img > 0)? '+' : '-',
      (img > 0)? img : (-1 * img)
    );

    System.out.println(fmt);
  }

  static Complex add(Complex n1, Complex n2) {
    return new Complex(n1.real + n2.real, n1.img + n2.img);
  }
  static Complex multiply(Complex n1, Complex n2) {
    // alg
    // (a + bi) * (c + di)
    // = ac + adi + bci - bd =
    // = (ac - bd) + (ad + bc)i
    int real = n1.real * n2.real - (n1.img * n2.img);
    int img = (n1.real * n2.img) + (n1.img * n2.real);
    return new Complex(real, img);
  }
  static void cmp(Complex n1, Complex n2) {
    if (n1.real == n2.real && n1.img == n2.img) {
      System.out.println("The complex numbers are equal");
    } else {
      System.out.println("The complex numbers are not equal");
    }
  }

  public static void main(String args[]) {
    Complex
```

```
      c1 = new Complex(2, 3),
      c2 = new Complex(1, -2),
      c3 = add(c1, c2),
      c4 = multiply(c1, c2);

    c1.display();
    c2.display();
    c3.display();
    c4.display();

    cmp(c1, c2);
    cmp(c1, c1);
  }
}
```

2.3 Output

11 Validation using Swing

11.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

11.2 Code

```
package org.projects.prog11;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.*;

//11. Write a swing program for validating the form having a numeric field,
//character field, phone number, and email ID.

public class Swing {
  public static void main(String[] args) {
    JFrame frame = new JFrame("Swing Example");
    frame.setLayout(new GridLayout(3, 2));
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(300, 130);

    JTextField phoneNum = new JTextField();
    frame.add(new JLabel("Phone"));
    frame.add(phoneNum);

    JTextField email = new JTextField();
    frame.add(new JLabel("Email"));
    frame.add(email);

    JButton submit = new JButton("Submit");
    JLabel result = new JLabel();
    result.setHorizontalAlignment(JLabel.CENTER);

    submit.addActionListener( new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        String em = email.getText();
        String phone = phoneNum.getText();
        result.setText("invalid");

        if (em.isEmpty() || phone.isEmpty()) { return; }

        try {
          Long.parseLong(phone);
          if (phone.length() < 10) { return; }
        } catch (Exception error) { return; }

        if (!em.contains("@")) { return; }

        result.setText("Valid");
      }
    });
```

```
    frame.add(submit);
    frame.add(result);
    frame.setVisible(true);
  }
}
```

11.3 Output

12 Text Field in Swing

12.1 Aim

- Create a Swing program that accepts a number
- if the number is even add it to list1
- else add it to list 2

12.2 Code

```
package org.projects.prog12;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.*;


public class Swing {
  public static void main(String[] args) {
    JFrame frame = new JFrame("Swing Example");
    frame.setLayout(new GridLayout(4, 2));
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(300, 130);

    JTextField number = new JTextField();
    frame.add(new JLabel("Number: "));
    frame.add(number);

    JLabel list1 = new JLabel();
    list1.setHorizontalAlignment(JLabel.CENTER);
    frame.add(new JLabel("List 1"));
    frame.add(list1);

    JLabel list2 = new JLabel();
    list2.setHorizontalAlignment(JLabel.CENTER);
    frame.add(new JLabel("List 2"));
    frame.add(list2);

    JButton submit = new JButton("Submit");

    submit.addActionListener( new ActionListener() {
      public void actionPerformed(ActionEvent e) throws NumberFormatException {
        int num = Integer.parseInt(number.getText());
        number.setText("");
        if (num % 2 == 0) {
          list1.setText(list1.getText() + " " + num);
        } else {
          list2.setText(list2.getText() + " " + num);
        }
      }
    });

    frame.add(submit);
    frame.setVisible(true);
  }
```

```
}
```

12.3 Output

13 Traffic Light Simulation

13.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

13.2 Code

```
package org.projects.prog13;
import java.awt.Color;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.*;

//13. Write a java program to simulate a traffic light. The program lets the user
//select one of the three lights: red, yellow, or green. On selecting a button,
//an appropriate message with "Stop", \Ready" or \ Go" should appear above the
//buttons selected color.

public class Traffic {
  public static void main(String[] args) {
    JFrame frame = new JFrame("Traffic");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setLayout(new GridLayout(2, 1));
    frame.setSize(300, 130);

    JLabel sign = new JLabel();
    sign.setHorizontalAlignment(JLabel.CENTER);
    frame.add(sign);

    JPanel panel = new JPanel();
    panel.setLayout(new GridLayout(1, 3));

    JButton red = new JButton("STOP");
    red.setBackground(Color.RED);
    panel.add(red);

    JButton yellow = new JButton("READY");
    yellow.setBackground(Color.YELLOW);
    panel.add(yellow);

    JButton green = new JButton("GO");
    green.setBackground(Color.GREEN);
    panel.add(green);

    ActionListener on_click = new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        sign.setText(((JButton) e.getSource()).getText());
      }
    };

    red.addActionListener(on_click);
    yellow.addActionListener(on_click);
    green.addActionListener(on_click);
```

```
    frame.add(panel);
    frame.setVisible(true);
  }
}
```

13.3 Output

14 Applet In Java

14.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

14.2 Code

```
package org.projects.prog14;

import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class Tables extends Applet implements ActionListener {
  TextField t1;
  Button t2;
  String[] result;

  public void init() {
    t1 = new TextField(5);
    t2 = new Button("Submit");
    t2.addActionListener(this);

    result = new String[10];

    add(new Label("Enter a number"));
    add(t1);
    add(t2);
  }

  public void paint(Graphics g) {
    for (int i = 0; i < 10; i++) {
      g.drawString(result[i], 20, 40 + 20*i);
    }
  }

  public void actionPerformed(ActionEvent e) {
    int number = Integer.parseInt(t1.getText());
    for (int i = 1; i <= 10; i++) {
      result[i-1] = number + " x " + i + " = " + i * number;
    }
  }
}
```

14.3 Output

15 Savings Account

15.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

15.2 Code

```java
package org.projects.prog15;

//15. Write a Java program to define Account class. Derive Saving_Account class
//from Account class. Define appropriate constructors for both classes. Define
//the following methods in the Saving_Account class:
//  Display( ): To display account details including account number and balance in
//  the account.
//  Deposit( ): To deposit money in an account.

class Account {
  public String accName;
  public long accNumber;

  Account (String name, long number) {
    accName = name;
    accNumber = number;
  }
}

public class SavingsAccount extends Account {
  private int accBalance;

  SavingsAccount(String name, long number) {
    super(name, number);
    accBalance = 0;
  }

  public void display() {
    System.out.println("Account: " + accName);
    System.out.println("No: " + accNumber);
    System.out.println("Balance: " + accBalance);
  }

  public void deposit(int ammount) {
    accBalance += ammount;
  }

  public static void main(String args[]) {
    SavingsAccount account = new SavingsAccount("Bijo", 1239812398);
    account.deposit(1000);
    account.display();
  }
}
```

15.3 Output

16 Integer Divisions

16.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

16.2 Code

```
package org.projects.prog16;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Dialog {
  public static void main (String args[]) {
    JFrame frame = new JFrame("cool frame");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(300, 200);
    frame.setLayout(new GridLayout(2, 1));

    JPanel panel = new JPanel();
    panel.setLayout(new GridLayout(2, 2));

    panel.add(new JLabel("Num 1: "));
    JTextField num1 = new JTextField();
    panel.add(num1);

    panel.add(new JLabel("Num 2: "));
    JTextField num2 = new JTextField();
    panel.add(num2);

    frame.add(panel);

    JButton submit = new JButton("Submit");
    submit.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        JDialog dialog = new JDialog();
        dialog.setSize(300, 50);

        JLabel prompt = new JLabel();
        prompt.setHorizontalAlignment(JLabel.CENTER);
        try {
          int n1 = Integer.parseInt(num1.getText());
          int n2 = Integer.parseInt(num2.getText());
          int result = n1 / n2;

          prompt.setText(n1 + "/" + n2 + " = " + result);
        } catch (NumberFormatException _e) {
          prompt.setText("Invalid Number");
        } catch (ArithmeticException _e) {
          prompt.setText("Division by Zero");
        }

        dialog.add(prompt);
        dialog.setVisible(true);
```

```
      }
    });

    frame.add(submit);
    frame.setVisible(true);
  }
}
```

16.3 Output

17 Simple Interest

17.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

17.2 Code

```
package org.projects.prog17;

import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class Interest extends Applet implements ActionListener {
  TextField principle, rate, time;
  Label result;
  Button submit;

  public void init() {
    principle = new TextField("principle");
    rate = new TextField("rate%");
    time = new TextField("time");

    add(new Label("Fill the following"));
    add(principle);
    add(rate);
    add(time);

    submit = new Button("Submit");
    submit.addActionListener(this);
    add(submit);

    result = new Label();
    add(result);
  }

  public void actionPerformed(ActionEvent e) {
    int p = Integer.parseInt(principle.getText());
    int r = Integer.parseInt(rate.getText());
    int n = Integer.parseInt(time.getText());

    float si = (p * r * n)/100;
    result.setText("" + si);
  }
}
```

17.3 Output

18 Mouse Coordinates

18.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

18.2 Code

```
package org.projects.prog18;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class MouseProg extends Applet implements MouseMotionListener {
  int x, y;

  public void init() {
    x = 0;
    y = 0;

    addMouseMotionListener(this);
  }

  public void paint(Graphics g) {
    g.drawString("x = " + x + " y = " + y, 20, 20);
  }

  public void mouseMoved(MouseEvent e) {
    x = e.getX();
    y = e.getY();
    repaint();
  }
  public void mouseDragged(MouseEvent e) {}
}
```

18.3 Output

19 Simple Banner

19.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

19.2 Code

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class Banner extends Applet implements Runnable {
  String bannerText;
  Thread scroll;

  public void init() {
    bannerText = "REALLY COOL BANNER ";
    scroll = new Thread(this);
  }

  public void start() {
    scroll.start();
  }

  public void paint(Graphics g) {
    g.drawString(bannerText, 100, 100);
  }

  public void run() {
    try {
      while (true) {
        Thread.sleep(500);
        bannerText = bannerText.substring(1) + bannerText.charAt(0);
        repaint();
      }
    } catch (Exception e) {}
  }
}
```

19.3 Output

20 Grid Layout Manager

20.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

20.2 Code

```
package org.projects.prog20;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class GridLay {
  public static void main(String args[]) {
    JFrame frame = new JFrame("Simple Grid");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(300, 200);
    frame.setLayout(new GridLayout(3, 1));

    JPanel panel = new JPanel();
    panel.setLayout(new GridLayout(1, 2));
    panel.add(new JLabel("Enter Name"));

    JTextField name = new JTextField();
    panel.add(name);

    frame.add(panel);

    JButton submit = new JButton("Submit");
    JLabel hello = new JLabel();

    submit.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        hello.setText("Hello " + name.getText() + "!!");
      }
    });
    frame.add(submit);

    frame.add(hello);

    frame.setVisible(true);
  }
}
```

20.3 Output

21 Priority Threads

21.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

21.2 Code

```
package org.projects.prog21;

public class Threading {
  public static Runnable simple_runnable = new Runnable() {
    public void run() {
      String name = Thread.currentThread().getName();
      for (int i = 0; i < 10; i++) {
        System.out.println(name + " " + i);
      }
    }
  };

  public static void main (String args[]) {
    Thread th0 = new Thread(simple_runnable), th1 = new Thread(simple_runnable);

    th0.setPriority(Thread.MIN_PRIORITY);
    th1.setPriority(Thread.MAX_PRIORITY);

    th0.start();
    th1.start();
  }
}
```

21.3 Output

22 Employee Details

22.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

22.2 Code

```
package org.projects.prog22;
import java.sql.*;

public class Employee {
  public static void main(String args[]) {
    try {
      Class.forName("com.mysql.cj.jdbc.Driver");
      String username = "root", password = "coolPass", dbname = "dbone";
      Connection cn = DriverManager.getConnection(
        //"jdbc:mysql://localhost:3306/dbone", "root", "coolPass"
        "jdbc:mysql://localhost:3306/" + dbname, username, password
      );

      Statement state = cn.createStatement();
      ResultSet rs = state.executeQuery("select * from employee");

      while (rs.next()) {
        System.out.println("Id: " + rs.getInt(1));
        System.out.println("Name: " + rs.getString(2));
        System.out.println("Designation: " + rs.getString(3));
        System.out.println();
      }

      cn.close();
    } catch (Exception e) {}
  }
}
```

22.3 Output

23 Update In JDBC

23.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

23.2 Code

```
package org.projects.prog23;
import java.sql.*;
import java.util.Scanner;

public class Student {
  public static Scanner sc = new Scanner(System.in);
  public static void main(String args[]) {
    try {
      Class.forName("com.mysql.cj.jdbc.Driver");
      String username = "root", password = "coolPass", dbname = "dbone";
      Connection cn = DriverManager.getConnection(
        //"jdbc:mysql://localhost:3306/dbone", "root", "coolPass"
        "jdbc:mysql://localhost:3306/" + dbname, username, password
      ); System.out.println("1 - add Student");
      System.out.println("2 - update Student");

      switch (sc.nextInt()) {
        case 1:
          addStudent(cn);
          break;
        case 2:
          updateStudent(cn);
          break;
        default:
          System.out.println("Invalid Choice");
          break;
      }

      displayStudents(cn);
      cn.close();
    } catch (Exception e) {}

  }

  public static void addStudent(Connection cn) {
    sc.nextLine(); // ignore garbage
    System.out.println("Enter name");
    String name = sc.nextLine();

    System.out.println("Enter roll number, age");
    int roll = sc.nextInt(), age = sc.nextInt();

    System.out.println("Enter Grade");
    char grade = sc.next().charAt(0);

    // to create the table use the following mysql code
    // create table students (
    //   roll int unique not null,
```

```
    //   name varchar(20) not null,
    //   age int not null,
    //   grade char(1) not null
    // );

    try {
      cn.createStatement().execute(String.format(
        "INSERT INTO students VALUES (%d, '%s', %d, '%c')",
        roll, name, age, grade
      ));
    } catch (Exception e) {
      System.out.println(e.getMessage());
    }
  }

  public static void updateStudent(Connection cn) {
    System.out.println("Enter rol number to update");
    int roll = sc.nextInt();

    System.out.println("Enter new age");
    int age = sc.nextInt();

    System.out.println("Enter new grade");
    char grade = sc.next().charAt(0);

    try {
      cn.createStatement().executeUpdate(String.format(
        "UPDATE students SET age = %d, grade = '%c' WHERE roll = %d",
        age, grade, roll
      ));
    } catch (Exception e) {
      System.out.println(e.getMessage());
    }
  }

  public static void displayStudents(Connection cn) {
    try {
      ResultSet set = cn.createStatement().executeQuery("select * from students");
      while (set.next()) {
        System.out.println(String.format(
          "roll: %d, name: %s, age: %d, grade: %s",
          set.getInt(1), set.getString(2), set.getInt(3), set.getString(4)
        ));
      }
    } catch (Exception e) {}
  }
}
```

23.3 Output

24 Product Details

24.1 Aim

- Create a Swing form with fields phone number and email ID
- Validate the fields

24.2 Code

```
package org.projects.prog24;
import java.sql.*;
import java.util.Scanner;

public class Product {
  public static Scanner sc = new Scanner(System.in);
  public static void main(String args[]) {
    try {
      Class.forName("com.mysql.cj.jdbc.Driver");
      String username = "root", password = "coolPass", dbname = "dbone";
      Connection cn = DriverManager.getConnection(
        //"jdbc:mysql://localhost:3306/dbone", "root", "coolPass"
        "jdbc:mysql://localhost:3306/" + dbname, username, password
      );

      boolean done = false;
      while (!done) {
        System.out.println("1 - add Product");
        System.out.println("2 - delete Product");
        System.out.println("3 - display");

        switch (sc.nextInt()) {
          case 1:
            addProduct(cn);
            break;
          case 2:
            deleteProduct(cn);
            break;
          case 3:
            displayProducts(cn);
            break;
          default:
            System.out.println("Invalid Choice");
            done = true;
            break;
        }
      }
      cn.close();
    } catch (Exception e) {}
  }

  public static void addProduct(Connection cn) {
    sc.nextLine(); // ignore garbage
    System.out.println("Enter Product name");
    String name = sc.nextLine();

    System.out.println("Enter product price");
    float price = sc.nextFloat();
```

```
    // to create the table use the following mysql code
    // create table products (
    //   id int unique AUTO_INCREMENT,
    //   name varchar(20) not null,
    //   price decimal(8, 2) not null,
    // );

    try {
      cn.createStatement().execute(String.format(
        "INSERT INTO products (name, price) VALUES ('%s', %f)",
        name, price
      ));
    } catch (Exception e) {
      System.out.println(e.getMessage());
    }
  }

  public static void deleteProduct(Connection cn) {
    System.out.println("Enter and id to delete");
    int id = sc.nextInt();

    try {
      cn.createStatement().execute(String.format(
        "DELETE FROM products WHERE id = %d",
        id
      ));
    } catch (Exception e) {
      System.out.println(e.getMessage());
    }
  }

  public static void displayProducts(Connection cn) {
    try {
      ResultSet set = cn.createStatement().executeQuery("select * from products");
      while (set.next()) {
        System.out.println(String.format(
          "id: %d, name: %s, cost: %f",
          set.getInt(1), set.getString(2), set.getFloat(3)
        ));
      }
    } catch (Exception e) {}
  }
}
```

24.3 Output