

ГУАП

КАФЕДРА № 44

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

старший преподаватель

должность, уч. степень, звание

подпись, дата

В. А. Ушаков

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4

Функциональное программирование

по курсу: ИТ-модуль "Разработка мобильных приложений"

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4142

подпись, дата

К.С. Некрасов

инициалы, фамилия

Санкт-Петербург 2023

Варианты

формула:

$$(4142 + 14) \% 10 + 1 = 7$$

Элемент ввода аргумента

$$(4142 + 14) \% 6 + 1 = 5 \text{ (JScrollBar)}$$

Элемент ввода точности

$$(4142 + 14) \% 4 + 1 = 1 \text{ (JTextField)}$$

Задание

$$\frac{1}{(1+x)} = 1 - x + x^2 - x^3 + x^4 - x^5$$

Листинг кода

Main.kt

```
import java.awt.EventQueue

//Создаем и показываем главное окно
private fun createAndShowGUI() {
    val frame = MainWindow("Simple")
    frame.isVisible = true
}

//Точка входа
fun main() {
    EventQueue.invokeLater { createAndShowGUI() }
}
```

MainWindow.kt

```
import java.awt.Color
import java.awt.Dimension
import java.awt.Insets
import javax.swing.*
import javax.swing.table.DefaultTableModel
import kotlin.math.pow
```

```

//Глобальнаяпеременнаяспособавводаданных
varselectedWay:Int=1;

//Глобальнаяпеременнаядляаргумента
varargument:Double=0.0;

//Переменныедлявводадиапазона
varstartArgument:Double=0.0;
varendArgument:Double=0.0;
varstep:Double=0.0;

//Таблицадлядиапазона
valtableModel=DefaultTableModel(arrayOf("x","Эталон","Результ"),0);
valtable=JTable(tableModel)
valtablePane=JScrollPane(table,JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,JS

//Кнопкивыборавариантаввода
valway1Button=JRadioButton("Ввододногоаргумента",true)
valway2Button=JRadioButton("Вводдиапазона")

//Элементыинтерфейса
valmaclaurinFormulaLabel=JLabel("ФормуларазложенияфункцииврядМаклорена:

//Элементыинтерфейсавводаединогоаргумента
valmaclaurinResultLabel=JLabel("Результатсразложением:")
valmaclaurinResult=JLabel()
valreferenceResultLabel=JLabel("Эталонныйрезультат:")
valreferenceResult=JLabel()

//кнопкаиинициализациивычислений
valcalculateButton=JButton("Вычислить")
valsetDefaultButton=JButton("Поумолч.")

//кнопказакрытияприложения
valcloseBtn=JButton("x")

//поледлявводаточности
valprecisionLabel=JLabel("Точностьвычислений(знаковпослезапятой):")
valprecisionInput=JTextField()

```

```

//панелидляспособовввода
valway1Pane=JPanel()
valway2Pane=JPanel()

//Элементывводадиапазонааргументов
valargumentLabel=JLabel("Аргументфункции:")
valargumentValueLabel=JLabel()
valargumentInput=JScrollBar(0,0,1,-9999,10000)
valstartArgumentLabel=JLabel("Первыйаргумент:")
valstartArgumentValueLabel=JLabel()
valstartArgumentInput=JScrollBar(0,0,1,-9999,10000)
valstepArgumentLabel=JLabel("Шаг:")
valstepArgumentValueLabel=JLabel()
valstepArgumentInput=JScrollBar(0,0,1,-9999,10000)
valendArgumentLabel=JLabel("Последнийаргумент:")
valendArgumentValueLabel=JLabel()
valendArgumentInput=JScrollBar(0,0,1,-9999,10000)

//главноеокно
classMainWindow(title:String):JFrame(){
  init{
    //заголовок
    setTitle(title)

    //выходпозакрытию
    defaultCloseOperation=EXIT_ON_CLOSE

    //настройкиокна
    setSize(500,520)
    setLocationRelativeTo(null)

    //панели
    valpanel=JPanel()
    panel.layout=null
    way1Pane.layout=null
    way2Pane.layout=null

    //размещениекнопкизакрытия
    closeBtn.setBounds(470,10,20,20)
    closeBtn.font=closeBtn.font.deriveFont(12.0f)

```

```

closeBtn.margin=Insets(0,0,0,0);
closeBtn.foreground=Color.RED
closeBtn.isFocusPainted=false

//логикакнопкизакрытия
closeBtn.addActionListener{
dispose()
}

//размещениеостальныхэлементовинтерфейса
maclaurinFormulaLabel.setBounds(10,10,400,25)

precisionLabel.setBounds(10,50,400,25);
precisionInput.setBounds(310,50,50,25)

//размещениеилогикакнопоквыборавариантаввода
way1Button.setBounds(10,90,200,25);
way1Button.addActionListener{selectedWay=1}
way1Button.addActionListener{way2Button.isSelected=false}
way1Button.addActionListener{way2Pane.isVisible=false}
way1Button.addActionListener{way1Pane.isVisible=true}
way2Button.setBounds(220,90,200,25);
way2Button.addActionListener{selectedWay=2}
way2Button.addActionListener{way1Button.isSelected=false}
way2Button.addActionListener{way1Pane.isVisible=false}
way2Button.addActionListener{way2Pane.isVisible=true}

//Элементывводавводаодногоаргумента
way1Pane.setBounds(0,130,500,300)
argumentLabel.setBounds(10,10,130,25);
argumentValueLabel.setBounds(140,10,100,25)
argumentInput.setBounds(250,10,240,25)
argumentInput.accessibleContext.addPropertyChangeListener{val1->if(val1
maclaurinResultLabel.setBounds(10,50,400,25)
maclaurinResult.setBounds(350,50,400,25)
referenceResultLabel.setBounds(10,90,400,25)
referenceResult.setBounds(350,90,400,25)

//Элементывводавводадиапазонааргументов
way2Pane.setBounds(0,130,500,300)

```

```

startArgumentLabel.setBounds(10,10,130,25);
startArgumentValueLabel.setBounds(140,10,100,25)
startArgumentInput.setBounds(250,10,240,25)
startArgumentInput.accessibleContext.addPropertyChangeListener{val1->if(
endArgumentLabel.setBounds(10,50,150,25);
endArgumentValueLabel.setBounds(160,50,100,25)
endArgumentInput.setBounds(250,50,240,25)
endArgumentInput.accessibleContext.addPropertyChangeListener{val1->if(v
stepArgumentLabel.setBounds(10,90,130,25);
stepArgumentValueLabel.setBounds(140,90,100,25)
stepArgumentInput.setBounds(250,90,240,25)
stepArgumentInput.accessibleContext.addPropertyChangeListener{val1->if(

//панельтаблицы
tablePane.setBounds(10,130,480,150)
tablePane.preferredSize=Dimension(400,150)
table.setBounds(0,0,480,200)

//размещениеилогикакнопкивычислений
calculateButton.setBounds(10,460,120,25)
calculateButton.addActionListener{
valprecisionText=precisionInput.text;
valprecision:Double;
try{
//пробуемперевеститекствdouble
precision=precisionText.toDouble();
if(selectedWay==1){
//считаемивыводимрезультатэталонный
referenceResult.text=calculateReference(argument).toString()
//считаемивыводимрезультатпоряду
maclaurinResult.text=calculateMaclaurin(argument,10.0.pow(-precision)).
}else{
//проверяемчтоаргументынорм
if(startArgument>endArgument){
throwIllegalArgumentException("Начальныйаргументдолженбытьбольшеконечно
}
//опустошаемтаблицу
tableModel.setNumRows(0);
//проходимподиапазонуизабиваемрезультатыиэталонвтаблицу
vari=startArgument;

```

```

while(i<endArgument){
    valreference=calculateReference(i).toString()
    valres=calculateMaclaurin(i,10.0.pow(-precision)).toString()
    tableModel.addRow(arrayOf(i,reference,res))
    i+=step
}
}
}catch(err:NumberFormatException){
    //еслитекстнеперевелсяпоказываемошибку
    JOptionPane.showMessageDialog(this,"Введено не число");
}catch(err:IllegalArgumentException){
    //еслиплохиепргументыпоказываемошибку
    JOptionPane.showMessageDialog(this,err.message);
}
}

//кнопкапоумолч
setDefaultButton.setBounds(140,460,100,25)
setDefaultButton.addActionListener{setDefaults()}

//добавляемэлементынапанели
panel.add(closeBtn)
panel.add(maclaurinFormulaLabel)
way1Pane.add(maclaurinResultLabel)
way1Pane.add(maclaurinResult)
way1Pane.add(referenceResultLabel)
way1Pane.add(referenceResult)
panel.add(precisionLabel)
panel.add(precisionInput)
panel.add(way1Button)
panel.add(way2Button)
way1Pane.add(argumentLabel)
way1Pane.add(argumentValueLabel)
way1Pane.add(argumentInput)
way2Pane.add(startArgumentLabel)
way2Pane.add(startArgumentValueLabel)
way2Pane.add(startArgumentInput)
way2Pane.add(endArgumentLabel)
way2Pane.add(endArgumentValueLabel)
way2Pane.add(endArgumentInput)

```

```

way2Pane.add(stepArgumentLabel)
way2Pane.add(stepArgumentValueLabel)
way2Pane.add(stepArgumentInput)
way2Pane.add(tablePane)
panel.add(way1Pane)
panel.add(way2Pane)
panel.add(calculateButton)
panel.add(setDefaultButton)

contentPane=panel
}
}

//функциячтобпоставитьзначениевлейблвпеременную
funsetArgumentValue(newValue:Double){
argumentValueLabel.text=newValue.toString();
argument=newValue;
}

//функциячтобпоставитьзначениевлейблвпеременную
funsetStepArgumentValue(newValue:Double){
stepArgumentValueLabel.text=newValue.toString();
step=newValue;
}

//функциячтобпоставитьзначениевлейблвпеременную
funsetStartArgumentValue(newValue:Double){
startArgumentValueLabel.text=newValue.toString();
startArgument=newValue;
}

//функциячтобпоставитьзначениевлейблвпеременную
funsetEndArgumentValue(newValue:Double){
endArgumentValueLabel.text=newValue.toString();
endArgument=newValue;
}

//функциякотораявсесбрасывает
funsetDefault(){
argumentInput.value=0
}

```



```

precisionInput.text=""
referenceResult.text=""
maclaurinResult.text=""
startArgumentInput.value=0
endArgumentInput.value=0
stepArgumentInput.value=0
setArgumentValue(0.0);
setStartArgumentValue(0.0)
setStepArgumentValue(0.0)
setEndArgumentValue(0.0)
}

```

Calculations.kt

```

import kotlin.math.abs
import kotlin.math.pow

```

//в подсчете по формуле Маклорена считаем пока разнице между предыдущим значением и сейчас

```

fun calculateMaclaurin(argument: Double, precision: Double): Double {
    var result = 0.0
    var prevResult: Double;
    var n = 0.0;

    do {
        prevResult = result;
        result += (-1.0).pow(n) * argument.pow(n);
        n++
    } while (abs(prevResult - result) >= precision)
    return result
}

```

//в подсчете эталонного значения просто считаем

```

fun calculateReference(argument: Double): Double {
    return 1.0 / (1 + argument);
}

```

Результат работы программы

Simple

Формула разложения функции в ряд Маклорена: $(-1)^n \cdot x^n$

Точность вычислений (знаков после запятой): 10

☐ Ввод одного аргумента ☒ Ввод диапазона

Первый аргумент: -0.2708000000000000...

Последний аргумент: 0.2812

Шаг: 0.0521

x	Эталон	Результ
-0.2708000000000000004	1.3713658804168953	1.371365880394119
-0.2187000000000000003	1.2799180852425445	1.2799180852348777
-0.1666000000000000003	1.1999040076793859	1.1999040076641592
-0.1145000000000000002	1.129305477131564	1.1293054771258297
-0.0624000000000000002	1.0665529010238908	1.0665529010229362
-0.01030000000000000017	1.0104071940992219	1.0104071940992099
0.041799999999999998	0.9598771357266269	0.9598771357270008
0.093899999999999998	0.9141603437242891	0.9141603437288638

Simple

Формула разложения функции в ряд Маклорена: $(-1)^n \cdot x^n$

Точность вычислений (знаков после запятой): 10

☒ Ввод одного аргумента ☐ Ввод диапазона

Аргумент функции: -0.4999

Результат с разложением: 1.9996000799262132

Эталонный результат: 1.9996000799840032

Вычислить По умолч.

