**Вопрос 1 Изучение возможностей памяти человека.**

Цель работы: Практическое изучение возможностей и ограничений памяти человека.

Основные понятия

Как известно, возможности памяти человека ограничены. В частности считается, что человек не может оперировать более чем семью (плюс-минус два) объектами одновременно. При этом реальное количество таких объектов может варьироваться в зависимости от типа объектов.

Будем считать, что возможны следующие графические формы представления чисел:

* в виде арабских цифр (например: «4»);
* в виде римских цифр (например: «IV»);
* словесная форма (например: «четыре»);
* пиктографическая (например, «( ( ( (»).

Кроме того каждому числу можно придать визуальную индивидуальность с помощью цвета, гарнитуры, стиля и размера шрифта.

Необходимо проанализировать, в какой степени влияет используемая форма представления на способность человека к запоминанию чисел.

**Задание**

Разработайте программу, реализующую следующие функции:

* генерацию набора цифр от нуля до девяти (количество объектов в наборе должно изменяться от одного до девяти, одинаковые цифры в наборе не допускаются);
* вывод набора цифр пользователю в одной из определённых в индивидуальном задании графических форм в течение фиксированного промежутка времени;
* вывод пользователю всех цифр (в заданной графической форме) для того, чтобы он мог указать, какие из них входили в набор;
* индикация ошибок пользователя (при их наличии).

Графическая форма:

Римские цифры, словесная форма, цвет шрифта.



Рис. 1. – Общая форма приложения. Начало тестирования



Рис. 2. – Ввод цифр которые отображались



Рис. 3. – Вывод результата тестирования

Исходный код программы:

**import** java.awt.EventQueue;

**import** javax.swing.JFrame;

**import** javax.swing.JButton;

**import** java.awt.Color;

**import** java.awt.event.ActionListener;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.\*;

**import** java.util.Random;

**import** java.util.Timer;

**import** java.util.TimerTask;

**import** java.util.stream.Collectors;

**import** java.util.stream.Stream;

**import** java.awt.event.ActionEvent;

**import** javax.swing.JLabel;

**import** javax.swing.JComboBox;

**import** javax.swing.JTextField;

**public** **class** **Main** {

 **JFrame** frame; // Главное окно приложения

 **JLabel** label; // Вывод цифр пользователю

 **JLabel** label\_1; // Вывод результата сравнения цифр пользователю

 **JComboBox**<**String**> comboBox; // Выбор графической формы

 **JComboBox**<**CData**> comboBox\_1; // Выбор цвета шрифта

 **JButton** btnNewButton\_1; // Ввод пользователем результата

 **Timer** timer; // Таймер времени показа цифр

 **int** count; // Количество цифр для показа

 **ArrayList**<**Integer**> digits; // Цифры для показа

 **CData**[] cdata = **new** **CData**[] { **new** **CData**("Красный", **Color**.red), **new** **CData**("Синий", **Color**.blue),

 **new** **CData**("Зеленый", **Color**.green) }; // Данные для цвета

 **private** **JTextField** textField; // Поле ввода пользователем цифр которые

 // показывались

 **private** **JTextField** textField\_1; // Поле ввода пользователем времени показа

 // цифр

 **private** **JLabel** lblNewLabel; // Надпись: время вывода

 **private** **JLabel** lblNewLabel\_1;// Надпись: цвет шрифта

 **private** **JLabel** lblNewLabel\_2; // Надпись: графическая форма

 /\*\*

 \* Launch the application.

 \*/

 **public** **static** **void** **main**(**String**[] args) {

 **EventQueue**.invokeLater(**new** **Runnable**() {

 **public** **void** **run**() {

 **try** {

 **Main** window = **new** **Main**();

 window.frame.setVisible(**true**);

 } **catch** (**Exception** e) {

 e.printStackTrace();

 }

 }

 });

 }

 /\*\*

 \* Create the application.

 \*/

 **public** **Main**() {

 initialize();

 }

 // Данные для выбора цвета

 **class** **CData** {

 **public** **String** scolor;

 **public** **Color** color;

 **@Override**

 **public** **String** **toString**() {

 **return** scolor;

 }

 **public** **CData**(**String** scolor, **Color** color) {

 **this**.scolor = scolor;

 **this**.color = color;

 }

 }

 // Словесные цифры

 **String** **WordDigits**(**int** digit) {

 **switch** (digit) {

 **case** 0:

 **return** "Ноль";

 **case** 1:

 **return** "Один";

 **case** 2:

 **return** "Два";

 **case** 3:

 **return** "Три";

 **case** 4:

 **return** "Четыре";

 **case** 5:

 **return** "Пять";

 **case** 6:

 **return** "Шесть";

 **case** 7:

 **return** "Семь";

 **case** 8:

 **return** "Восемь";

 **case** 9:

 **return** "Девять";

 **default**:

 **return** "Ноль";

 }

 }

 // Римские цифры

 **String** **RimeDigits**(**int** digit) {

 **switch** (digit) {

 **case** 0:

 **return** "0";

 **case** 1:

 **return** "I";

 **case** 2:

 **return** "II";

 **case** 3:

 **return** "III";

 **case** 4:

 **return** "IV";

 **case** 5:

 **return** "V";

 **case** 6:

 **return** "VI";

 **case** 7:

 **return** "VII";

 **case** 8:

 **return** "VIII";

 **case** 9:

 **return** "IX";

 **default**:

 **return** "0";

 }

 }

 /\*\*

 \* Initialize the contents of the frame.

 \*/

 **private** **void** **initialize**() {

 frame = **new** **JFrame**();

 frame.setBounds(100, 100, 560, 272);

 frame.setDefaultCloseOperation(**JFrame**.EXIT\_ON\_CLOSE);

 frame.getContentPane().setLayout(**null**);

 label = **new** **JLabel**("Исходные числа: ");

 label.setBounds(10, 11, 524, 14);

 label.setForeground(**Color**.red);

 frame.getContentPane().add(label);

 **JButton** btnNewButton = **new** **JButton**("Начать тест");

 btnNewButton.addActionListener(**new** **ActionListener**() {

 // Вывод чисел для теста

 **public** **void** **actionPerformed**(**ActionEvent** e) {

 label.setText("Исходные числа: ");

 label\_1.setText("Результат: ");

 textField.setVisible(**false**);

 btnNewButton\_1.setVisible(**false**);

 btnNewButton.setVisible(**false**);

 **if** (digits != **null**) digits.clear();

 **else**

 digits = **new** **ArrayList**<**Integer**>();

 // Генерация цифр от 1 до 9 со значением от 0 до 9

 java.util.**Random** rnd = **new** **Random**(**new** **Date**().getTime());

 count = rnd.nextInt(9) + 1;

 java.util.stream.**Stream**.generate(() -> rnd.nextInt(10)).takeWhile(elem -> {

 **if** (digits.size() < count) **return** **true**;

 **else**

 **return** **false**;

 }).filter(elem -> !digits.contains(elem)).forEach(elem -> digits.add(elem));

 // Вывод цифр

 **if** (comboBox.getSelectedItem().toString().equals("Римские цифры")) {

 digits.stream().forEach(elem -> label.setText(label.getText() + " " + RimeDigits(elem)));

 } **else** {

 digits.stream().forEach(elem -> label.setText(label.getText() + " " + WordDigits(elem)));

 }

 label.setForeground(((**CData**) comboBox\_1.getSelectedItem()).color);

 **int** value;

 **try** {

 value = 1000 \* **Integer**.parseInt(textField\_1.getText());

 } **catch** (**Exception** exep) {

 value = 3000;

 }

 timer.schedule(**new** **TimerTask**() {

 **@Override**

 **public** **void** **run**() {

 label.setText("Исходные числа: ");

 **if** (comboBox.getSelectedItem().toString().equals("Римские цифры")) {

 Stream.iterate(0, elem -> ++elem).limit(10)

 .forEach(elem -> label.setText(label.getText() + " " + RimeDigits(elem)));

 } **else** {

 Stream.iterate(0, elem -> ++elem).limit(10)

 .forEach(elem -> label.setText(label.getText() + " " + WordDigits(elem)));

 }

 textField.setVisible(**true**);

 btnNewButton\_1.setVisible(**true**);

 }

 }, value);

 }

 });

 btnNewButton.setBounds(170, 199, 181, 23);

 frame.getContentPane().add(btnNewButton);

 comboBox = **new** **JComboBox**<**String**>();

 comboBox.addItem("Римские цифры");

 comboBox.addItem("Словесная форма");

 comboBox.setBounds(403, 64, 131, 23);

 frame.getContentPane().add(comboBox);

 comboBox\_1 = **new** **JComboBox**<**CData**>();

 comboBox\_1.setBounds(403, 118, 131, 23);

 **Arrays**.stream(cdata).forEach(data -> comboBox\_1.addItem(data));

 frame.getContentPane().add(comboBox\_1);

 textField = **new** **JTextField**();

 textField.setBounds(10, 119, 356, 20);

 frame.getContentPane().add(textField);

 textField.setColumns(10);

 textField.setVisible(**false**);

 label\_1 = **new** **JLabel**("Результат: ");

 label\_1.setBounds(10, 68, 356, 14);

 frame.getContentPane().add(label\_1);

 textField\_1 = **new** **JTextField**();

 textField\_1.setText("3");

 textField\_1.setBounds(403, 177, 131, 20);

 frame.getContentPane().add(textField\_1);

 textField\_1.setColumns(10);

 btnNewButton\_1 = **new** **JButton**("Ответить");

 btnNewButton\_1.addActionListener(**new** **ActionListener**() {

 // Нажата кнопка ответить

 **public** **void** **actionPerformed**(**ActionEvent** e) {

 textField.setVisible(**false**);

 btnNewButton\_1.setVisible(**false**);

 btnNewButton.setVisible(**true**);

 label\_1.setText("Результат: ");

 **List**<**Integer**> udigits = **Arrays**.stream(textField.getText().split(",")).map(elem -> elem.trim())

 .map(elem -> {

 **try** {

 **return** **Integer**.parseInt(elem);

 } **catch** (**Exception** exep) {

 **return** -1;

 }

 }).collect(**Collectors**.toList());

 digits.forEach(elem -> {

 **if** (!udigits.contains(elem)) {

 label\_1.setText(label\_1.getText() + " " + elem);

 }

 });

 **if** (label\_1.getText().equals("Результат: "))

 label\_1.setText(label\_1.getText() + "все числа записаны верно.");

 **else**

 label\_1.setText(label\_1.getText() + " - не указаны.");

 }

 });

 btnNewButton\_1.setBounds(170, 199, 181, 23);

 frame.getContentPane().add(btnNewButton\_1);

 lblNewLabel = **new** **JLabel**("Время вывода");

 lblNewLabel.setBounds(403, 152, 131, 14);

 frame.getContentPane().add(lblNewLabel);

 lblNewLabel\_1 = **new** **JLabel**("Цвет шрифта");

 lblNewLabel\_1.setBounds(401, 98, 133, 14);

 frame.getContentPane().add(lblNewLabel\_1);

 lblNewLabel\_2 = **new** **JLabel**("Графическая форма");

 lblNewLabel\_2.setBounds(403, 39, 131, 14);

 frame.getContentPane().add(lblNewLabel\_2);

 btnNewButton\_1.setVisible(**false**);

 timer = **new** **Timer**();

 }

}