**Лабораторная работа №4**

**Иванченко Павла**

Тема: средства для рисования в Java

Цель: изучить средства для рисования в Java и разработать приложение с графическим интерфейсом.

**Постановка задачи.**

Необходимо разработать тренажер клавиатуры. По прямоугольному полю должны двигаться круги с буквой внутри. Если нажата клавиша, соответствующая букве, игроку добавляется балл, иначе – нет. Если клавиша не нажата до того, как круг упал за поле, балл вычитается из счета игрока. Со временем игра должна ускоряться.

**Ход работы**

### Основные блоки программы отвечают за интерфейс, который виден пользователям, логику игры и логику работы программы.

Класс KeyboardTrainer - основной класс, который наследует JPanel и реализует ActionListener для обработки событий таймера.

public class KeyboardTrainer extends JPanel implements ActionListener {  
 private char currentLetter;  
 private double x, y;  
 private int score = 0;  
 private double speed = 2.0;  
 private Timer timer;  
 private Random random;  
 private JButton startButton;  
 private boolean gameRunning = false;  
 private long lastUpdateTime;

#### Логика игры описана в следующих методах:

* Генерация букв (generateNewLetter()) - случайным образом выбирает новую букву и задает ей стартовые координаты.

private void generateNewLetter() {  
 currentLetter = (char) ('a' + random.nextInt(26));  
 x = random.nextInt(getWidth() - 50);  
 y = 0;  
}

* Обработка нажатий клавиш - при правильном нажатии буквы: увеличивается счет, проверяется необходимость увеличения сложности, создается новая буква.

public void keyTyped(KeyEvent e) {  
 if (gameRunning && e.getKeyChar() == currentLetter) {  
 score++;  
 increaseDifficulty();  
 generateNewLetter();  
 checkGameOver();  
 }  
}

* Изменение сложности (increaseDifficulty()) - каждые 5 правильных нажатий скорость падения увеличивается.

private void increaseDifficulty() {  
 if (score % 5 == 0) {  
 speed += 0.5;  
 }  
}

* Завершение игры (checkGameOver()) - игра останавливается при достижении 50 очков.

private void checkGameOver() {  
 if (score >= 50) {  
 timer.stop();  
 gameRunning = false;  
 JOptionPane.showMessageDialog(this, "Поздравляем! Вы набрали 50 очков!");  
 startButton.setVisible(true);  
 }  
}

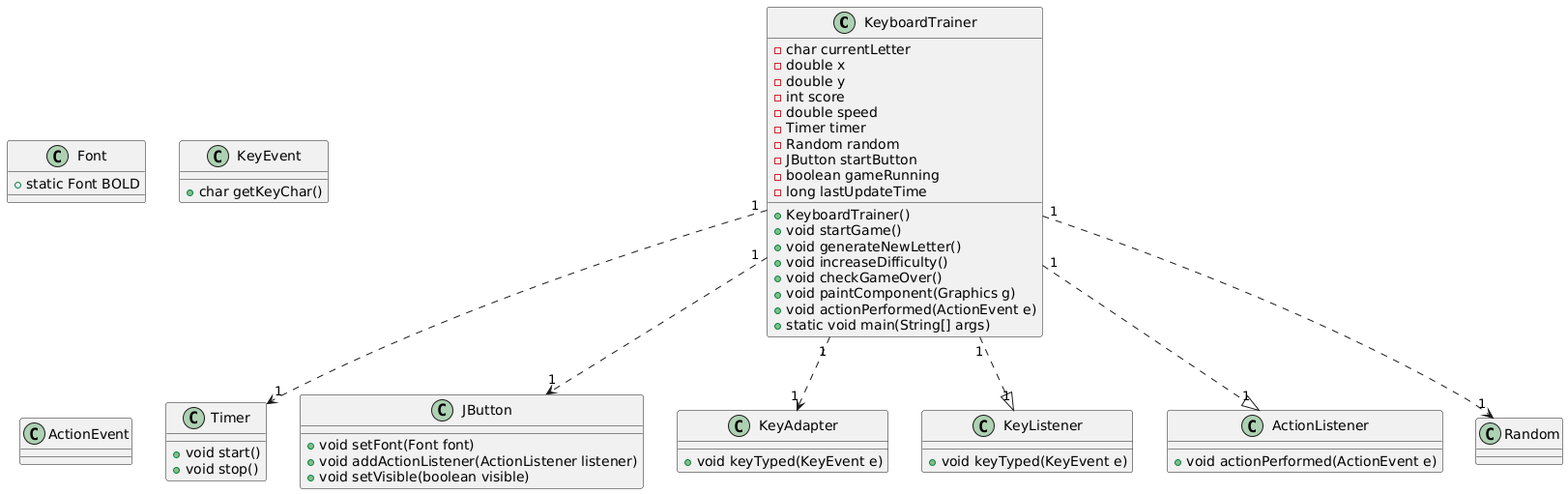
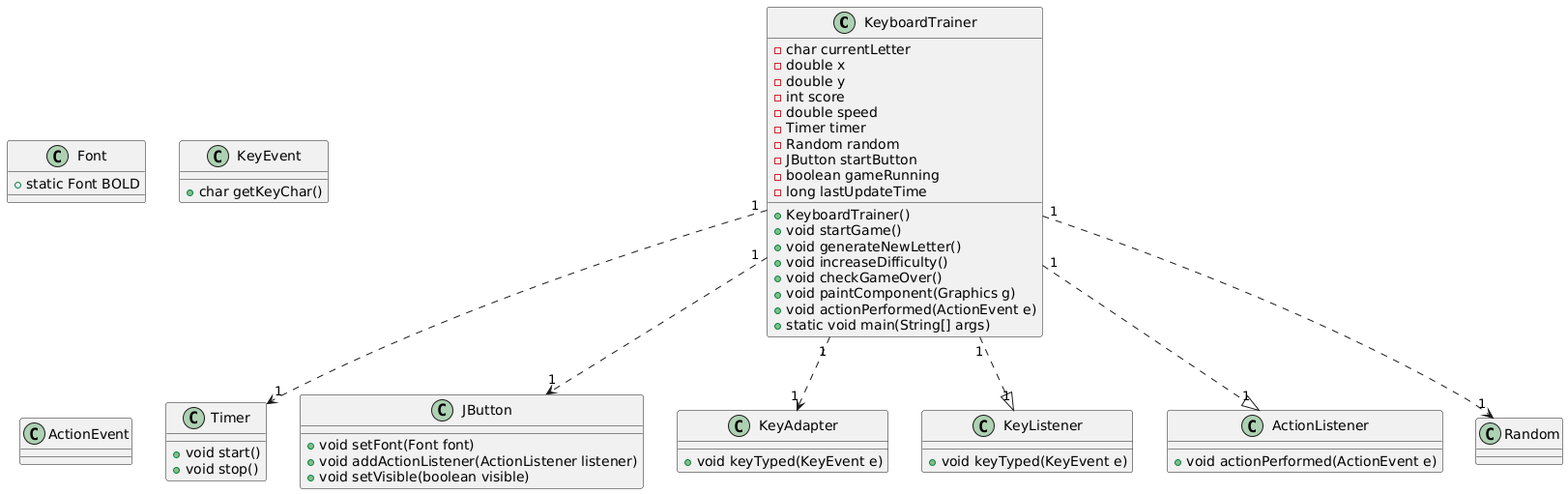


Рис.1, 2 – UML-диаграмма классов

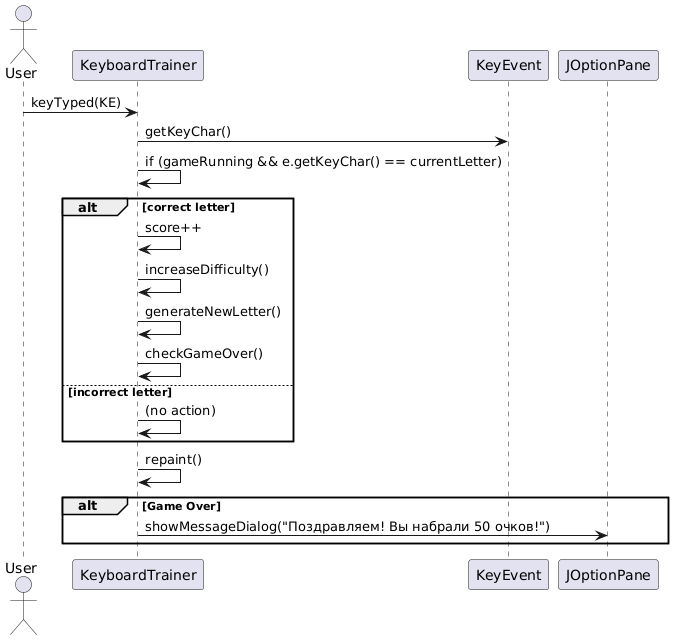


Рис. 3 – Диаграмма последовательностей для события keyTyped

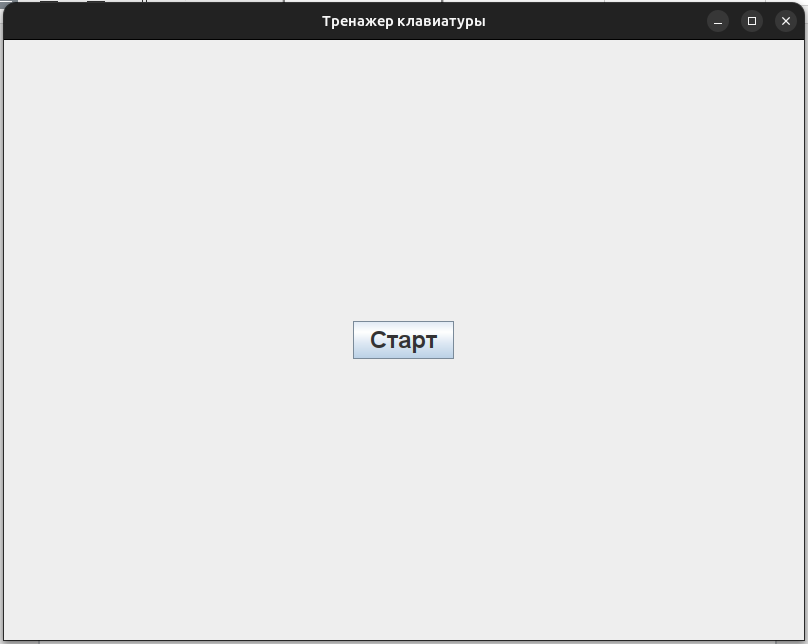


Рис. 4 – Стартовое окно программы

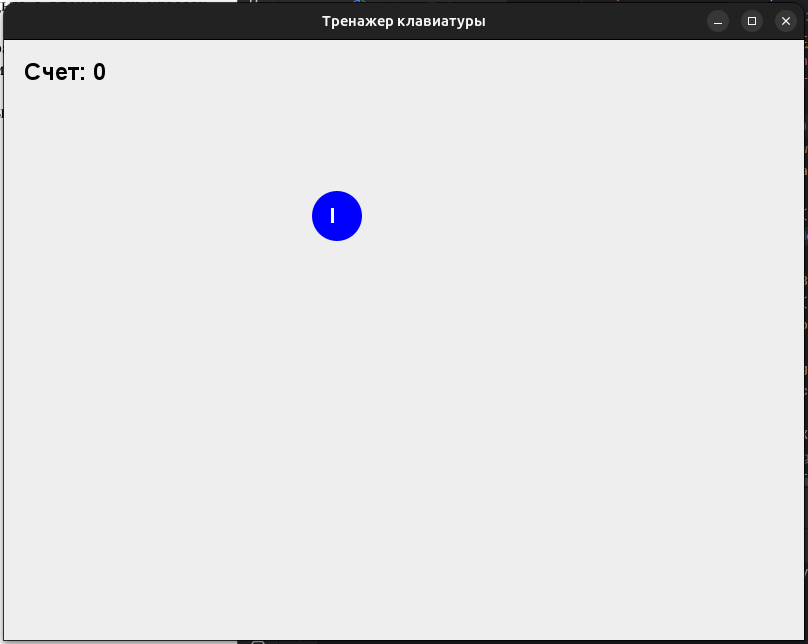


Рис. 5 – Программа в статусе gameRunning

ПРИЛОЖЕНИЕ А

Код программы

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.\*;  
import java.util.Random;  
  
public class KeyboardTrainer extends JPanel implements ActionListener {  
 private char currentLetter;  
 private double x, y;  
 private int score = 0;  
 private double speed = 2.0;  
 private Timer timer;  
 private Random random;  
 private JButton startButton;  
 private boolean gameRunning = false;  
 private long lastUpdateTime;  
  
 public KeyboardTrainer() {  
 setPreferredSize(new Dimension(800, 600));  
 setLayout(new GridBagLayout());  
 setFocusable(true);  
 random = new Random();  
 timer = new Timer(5, this);  
  
 startButton = new JButton("Старт");  
 startButton.setFont(new Font("Arial", Font.BOLD, 24));  
 startButton.addActionListener(e -> startGame());  
  
 GridBagConstraints gbc = new GridBagConstraints();  
 add(startButton, gbc);  
  
 addKeyListener(new KeyAdapter() {  
 @Override  
 public void keyTyped(KeyEvent e) {  
 if (gameRunning && e.getKeyChar() == currentLetter) {  
 score++;  
 increaseDifficulty();  
 generateNewLetter();  
 checkGameOver();  
 }  
 }  
 });  
 }  
  
 private void startGame() {  
 gameRunning = true;  
 startButton.setVisible(false);  
 score = 0;  
 speed = 2.0;  
 generateNewLetter();  
 lastUpdateTime = System.nanoTime();  
 timer.start();  
 requestFocusInWindow();  
 }  
  
 private void generateNewLetter() {  
 currentLetter = (char) ('a' + random.nextInt(26));  
 x = random.nextInt(getWidth() - 50);  
 y = 0;  
 }  
  
 private void increaseDifficulty() {  
 if (score % 5 == 0) {  
 speed += 0.5;  
 }  
 }  
  
 private void checkGameOver() {  
 if (score >= 50) {  
 timer.stop();  
 gameRunning = false;  
 JOptionPane.showMessageDialog(this, "Поздравляем! Вы набрали 50 очков!");  
 startButton.setVisible(true);  
 }  
 }  
  
 @Override  
 protected void paintComponent(Graphics g) {  
 super.paintComponent(g);  
 if (gameRunning) {  
 Graphics2D g2d = (Graphics2D) g;  
 g2d.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON);  
  
 g2d.setColor(Color.BLUE);  
 g2d.fillOval((int)x, (int)y, 50, 50);  
  
 g2d.setColor(Color.WHITE);  
 g2d.setFont(new Font("Arial", Font.BOLD, 20));  
 g2d.drawString(String.valueOf(currentLetter), (int)x + 18, (int)y + 32);  
  
 g2d.setColor(Color.BLACK);  
 g2d.setFont(new Font("Arial", Font.BOLD, 24));  
 g2d.drawString("Счет: " + score, 20, 40);  
 }  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 if (gameRunning) {  
 long currentTime = System.nanoTime();  
 double deltaTime = (currentTime - lastUpdateTime) / 1\_000\_000\_000.0;  
 lastUpdateTime = currentTime;  
  
 y += speed \* deltaTime \* 60;  
  
 if (y + 50 > getHeight()) {  
 score--;  
 generateNewLetter();  
 checkGameOver();  
 }  
 repaint();  
 }  
 }  
  
 public static void main(String[] args) {  
 JFrame frame = new JFrame("Тренажер клавиатуры");  
 KeyboardTrainer trainer = new KeyboardTrainer();  
 frame.add(trainer);  
 frame.pack();  
 frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 frame.setLocationRelativeTo(null);  
 frame.setVisible(true);  
 }  
}