Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Глазовский государственный педагогический институт имени В.Г. Короленко»

**ОТЧЕТ**

**по учебной практике**

**ПМ 03Участие в интеграции программных модулей**

**Специальность 09.02.03 Программирование в компьютерных системах**

|  |
| --- |
| Выполнил(а): |
| студент(ка) ОП СПО группа 43 |
| Шадрин М.С. |
|  |
| Руководитель практики: ст. преподаватель Касаткин К.А. |

г. Глазов 2019

Оглавление

[Введение 4](#_Toc4074454)

[Цель разработки 5](#_Toc4074455)

[Обзор аналогов 5](#_Toc4074456)

[Постановка задачи 5](#_Toc4074457)

[Описание алгоритма работы программы 6](#_Toc4074458)

[Исходный код программы 7](#_Toc4074459)

[Тестирование программы 24](#_Toc4074460)

[Руководство по установке 24](#_Toc4074461)

# Введение

На данный момент существует великое множество компьютерных игр, так же разнообразны подходы в их создании.

Например, популярные в последнее время небольшие игры (чаще всего многопользовательские, через сеть Интернет) создаются по технологии Flash с использованием языка ActionScript. Такие игры крайне популярны на разнообразных развлекательных интернет-ресурсах и а социальных сетях.

Большие, сложные игры с реалистичной 3D графикой пишутся на C++, отдельные модули к ним могут быть написаны на других языках (например AI - «искусственный интеллект» - на Lisp). Вообще, язык для создания игры выбирается как компромисс между требованиями к игре, к компьютерному железу и к стоимости разработки.

На подавляющем большинстве мобильных телефонов установлены ARM процессоры. Поэтому игры для мобильных телефонов на платформе Java пишутся на специальном языке для мобильных телефонов MobileBasic, а также на диалекте языка Pascal - MIDletPascal.

В рамках учебной практики необходимо разработать небольшую программу игры в «крестики-нолики» поле размера 3х3 клеток с применением графики на языке высокого уровня с использованием объектно-ориентированного программирования.

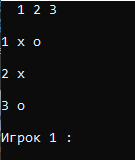
«Крестики-нолики» - популярная игра людей всех возрастов. Она отличается простотой, способствует развитию внимания и элементарной логики. В данной работе эта игра реализована на языке Java.

# Цель разработки

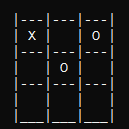
Создать интересную и логическую игру, для детей и взрослых, игра будет в открытом доступе и все желающие могут воспользоваться.

# Обзор аналогов

1. Крестики-нолики создал пользователь D-MAN. В его коде, ход осуществляется введением двух цифр - сначала по вертикали,затем по горизонтали.



1. Крестики-нолики с компьютером. В этом коде, ход осуществляется с помощью нажатием кнопки мыши на квадрат и нажатием на пробел. Пробел рисует Х.

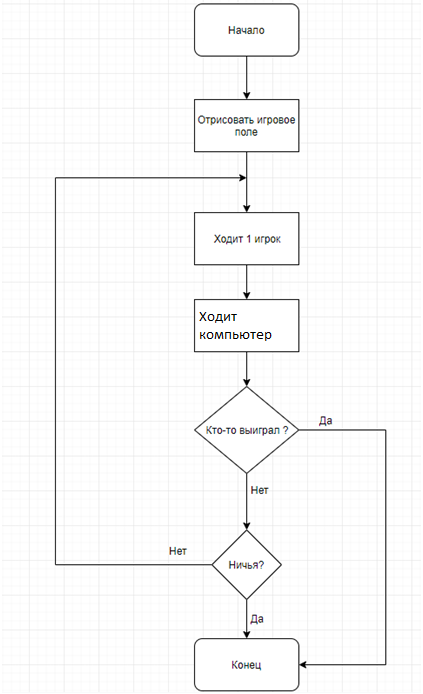


# Постановка задачи

1. Написать программу, реализующую идею игры. Выполнить переопределение стандартных операций для абстрактных типов данных. Использовать конструкторы, деструкторы. Использовать графический режим.

# Описание алгоритма работы программы

1. Отрисовать игровое поле.
2. Позволить сделать шаг игроку.
3. Проверить выиграл ли кто-то.
4. Если да - выиграл, иначе - вернуться к шагу 2.



# Исходный код программы

Для написания игры, я взял язык программирования Java. Так как этот язык новый для меня и я хотел изучить его. import java.awt.\*;

import java.awt.event.\*;

import java.util.Random;

import java.util.Vector;

import javax.swing.\*;

import javax.swing.border.\*;

public class XO extends JFrame implements ActionListener {

/////////////////////////////////////////////////////////////////////////

////////-------------Fields-------------------------------------------///

private static final long serialVersionUID = 1L;

private JPanel jContentPane = null;

private JPanel jPanel = null;

private JPanel jPanel1 = null;

private JButton New = null;

private JButton Ex = null;

private JButton[] pole = null;

private JLabel StateGame = null;

private int m[]= new int [9];

private Random rand = new Random();

private int ButtonsInGame, vin;

final int NULL = 0,

X = 1,

O = (-1);

/////////////////////////////////////////////////////////////////////////

///////------------------Main-----------------------------------------///

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

public void run() {

XO thisClass = new XO();

thisClass.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

thisClass.setVisible(true);

}

});

}

/////////////////////////////////////////////////////////////////////////

///---------------Interface builders----------------------------------///

/\*\*

\* Constructor

\*/

public XO() {

super();

initialize();

}

/\*\*

\* Initialize function

\*/

private void initialize() {

this.setSize(240, 268);

this.setContentPane(getJContentPane());

this.setTitle("JFrame");

}

/\*\*

\* Content builder

\*/

private JPanel getJContentPane() {

if (jContentPane == null) {

StateGame = new JLabel();

StateGame.setText(vin\_getter(3));

StateGame.setHorizontalAlignment(SwingConstants.CENTER);

StateGame.setBackground(Color.black);

StateGame.setDisplayedMnemonic(KeyEvent.VK\_UNDEFINED);

jContentPane = new JPanel();

jContentPane.setLayout(new BorderLayout());

jContentPane.setBackground(new Color(102, 102, 255));

jContentPane.add(getJPanel(), BorderLayout.NORTH);

jContentPane.add(getJPanel1(), BorderLayout.CENTER);

jContentPane.add(StateGame, BorderLayout.SOUTH);

}

return jContentPane;

}

/\*\*

\* Panel for buttons

\*/

private JPanel getJPanel() {

if (jPanel == null) {

jPanel = new JPanel();

jPanel.setLayout(new FlowLayout());

jPanel.setBorder(BorderFactory.createBevelBorder(BevelBorder.LOWERED));

jPanel.setBackground(new Color(102, 102, 102));

jPanel.add(getNew(), null);

jPanel.add(getEx(), null);

}

return jPanel;

}

/\*\*

\* Button "New game"

\*/

private JButton getNew() {

if (New == null) {

New = new JButton("New game");

New.setBackground(new Color(102, 255, 102));

New.addActionListener(this);

}

return New;

}

/\*\*

\* Button "Exit"

\*/

private JButton getEx() {

if (Ex == null) {

Ex = new JButton("Exit");

Ex.setBackground(new Color(255, 102, 102));

Ex.addActionListener(this);

}

return Ex;

}

/\*\*

\* Main Panel witch contains pole(Buttons[])

\*/

private JPanel getJPanel1() {

if (jPanel1 == null) {

GridLayout gridLayout = new GridLayout(3,3,1,1);

jPanel1 = new JPanel();

jPanel1.setBackground(new Color(51, 51, 51));

jPanel1.setLayout(gridLayout);

pole = new JButton[9];

for(int i=0;i<pole.length;i++){

pole[i] = new JButton();

setState(i, NULL);

pole[i].addActionListener(this);

pole[i].setForeground(new Color(200,200,255));

jPanel1.add(pole[i]);

pole[i].setOpaque(true);

pole[i].setContentAreaFilled(false);

pole[i].setBackground(new Color(102,255,102));

}

}

return jPanel1;

}

/\*\*

\* Function, witch set Buttons to X or O

\*/

private void setState(int index, int state){

pole[index].setText(toStrTipe(state));

pole[index].setEnabled(!(state==1||state==-1));

ButtonsInGame--;

}

////////////////////////////////////////////////////////////////////////////

//########################################################################//

////////////////////////////////////////////////////////////////////////////

/\*\*

\* This functions listen actions,

\*/

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource().equals(Ex))System.exit(0);

if(e.getSource().equals(New))New\_Game();

for(int i=0 ; i<9 ; i++){

if(e.getSource().equals(pole[i])){

setState(i, O);

set\_mass();

vin = lock\_winner();

if(vin!=0)StateGame.setText(vin\_getter(vin));

else{

setState(tactica1(), X);

set\_mass();

int ret = lock\_winner() ;

if(ret>0)StateGame.setText(vin\_getter(ret));

}

}

}

}

/\*\*

\* This function play with player

\*/

private int tactica1(){

if(ButtonsInGame==9)return 4;

if(ButtonsInGame==8)

while(true){

switch(getRand(3)){

case 0:

if(m[0]==0)return 0;

case 1:

if(m[2]==0)return 2;

case 2:

if(m[6]==0)return 6;

case 3:

if(m[8]==0)return 8;

continue;

}

}

if(ButtonsInGame<8){

int res = empty\_pole(X);

if(res!=-1)return res;

res = empty\_pole(O);

if(res!=-1)return res;

}

return rand\_pole();

}

//--------------------------------------------------------------//

/\*\*

\* This function work after "New game" pressed

\*/

protected void New\_Game() {

for(int i=0;i<9;i++){

setState(i, NULL);

pole[i].setContentAreaFilled(false);

}

ButtonsInGame = 9;

StateGame.setText(vin\_getter(3));

int in = tactica1();

if(in!=-1)setState(in, X);

}

//////////////////////////////////////////////////////////////////////////

//----------------------------Serchers----------------------------------//

//////////////////////////////////////////////////////////////////////////

///---------------------------win-loocker------------------------------///

/\*\*

\* This function return win-constant or -1.

\*/

private int lock\_winner()

{

if(ButtonsInGame==0)return 2;

if(m[0]+m[1]+m[2]==3||m[0]+m[1]+m[2]==-3){

line(0,1,2);

return m[0];

}

if(m[0]+m[3]+m[6]==3||m[0]+m[3]+m[6]==-3){

line(0,3,6);

return m[0];

}

if(m[6]+m[7]+m[8]==3||m[6]+m[7]+m[8]==-3){

line(6,7,8);

return m[8];

}

if(m[2]+m[5]+m[8]==3||m[2]+m[5]+m[8]==-3){

line(2,5,8);

return m[8];

}

if(m[0]+m[4]+m[8]==3||m[0]+m[4]+m[8]==-3){

line(0,4,8);

return m[4];

}

if(m[2]+m[4]+m[6]==3||m[2]+m[4]+m[6]==-3){

line(2,4,6);

return m[4];

}

if(m[1]+m[4]+m[7]==3||m[1]+m[4]+m[7]==-3){

line(1,4,7);

return m[4];

}

if(m[3]+m[4]+m[5]==3||m[3]+m[4]+m[5]==-3){

line(3,4,5);

return m[4];

}

return NULL;

}

/\*\*

\* This function returns text of state-label.

\*/

private String vin\_getter(int stat)

{

if(stat==X)return "Win player x";

if(stat==O)return "Win player o";

if(stat==2)return "Erounded";

if(stat==3)return "Unknown";

return null;

}

/\*\*

\* This function convert win-constant to win-symbol(X or O)

\*/

private String toStrTipe(int state){

if(state==1)return "X";

if(state==-1)return "O";

return "";

}

/\*\*

\* This function "draw win-line"

\*/

private void line(int a, int b, int c){

pole[a].setContentAreaFilled(true);

pole[b].setContentAreaFilled(true);

pole[c].setContentAreaFilled(true);

}

//////////////////////////////////////////////////////////////////////////

//-------------------------Utils----------------------------------------//

/\*\*

\* This function search win-pole to entered player

\*/

private int empty\_pole(int player)

{

int n = (player==O)?-2:2; // equals = 2,-2

System.out.print(toStrTipe(player)+": ");

if(m[0]+m[1]+m[2]==n)

{

if(m[0]==0)

return 0;

if(m[1]==0)

return 1;

if(m[2]==0)

return 2;

}

else if(m[3]+m[4]+m[5]==n)

{

if(m[3]==0)

return 3;

if(m[4]==0)

return 4;

if(m[5]==0)

return 5;

}

else if(m[6]+m[7]+m[8]==n)

{

if(m[6]==0)

return 6;

if(m[7]==0)

return 7;

if(m[8]==0)

return 8;

}

else if(m[0]+m[3]+m[6]==n)

{

if(m[0]==0)

return 0;

if(m[3]==0)

return 3;

if(m[6]==0)

return 6;

}

else if(m[1]+m[4]+m[7]==n)

{

if(m[1]==0)

return 1;

if(m[4]==0)

return 4;

if(m[7]==0)

return 7;

}

else if(m[2]+m[5]+m[8]==n)

{

if(m[2]==0)

return 2;

if(m[5]==0)

return 5;

if(m[8]==0)

return 8;

}

else if(m[0]+m[4]+m[8]==n)

{

if(m[0]==0)

return 0;

if(m[4]==0)

return 4;

if(m[8]==0)

return 8;

}

else if(m[6]+m[4]+m[2]==n)

{

if(m[6]==0)

return 6;

if(m[4]==0)

return 4;

if(m[2]==0)

return 2;

}

return -1;

}

//////////////////////////////////////////////////////////////////////////

//-------------------------Processing support---------------------------//

/\*\*

\* This function reset sub-pole for processing

\*/

private void set\_mass()

{

for(int ii=0;ii<9;ii++)

{

if(pole[ii].getText().equals("X"))

m[ii]=X;

else if(pole[ii].getText().equals("O"))

m[ii]=O;

else

m[ii]=NULL;

}

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println(" "+m[0]+" "+m[1]+" "+m[2]);

System.out.println(" "+m[3]+" "+m[4]+" "+m[5]);

System.out.println(" "+m[6]+" "+m[7]+" "+m[8]);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

//////////////////////////////////////////////////////////////////////////

///-----------------------Random---------------------------------------///

/\*\*

\* return unsigned integer with limit "int lim"

\*/

private int getRand(int lim){

if(lim<1)return -1;

return Math.abs(rand.nextInt(lim));

}

/\*\*

\*return random empty pole

\*/

private int rand\_pole(){

Vector <Integer> v = new Vector<Integer>();

for(int i=0;i<9;i++)

if(m[i]==0)v.add(i);

return v.get(getRand(v.size()));

}

} // @jve:decl-index=0:visual-constraint="4,10"

# Тестирование программы



# Руководство по установке

Скачать Intelly IDEA и библиотеку SDK 1.8 Запустить проект в Intelly IDEA.