МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное

учреждение высшего образования

«Казанский национальный исследовательский технический

университет им. А.Н. Туполева-КАИ»

(КНИТУ-КАИ)

Институт компьютерных технологий и защиты информации

Кафедра автоматизированных систем обработки информации и управления

**ОТЧЕТ**

**по прохождению технологической (учебной) практики**

Направление подготовки / специальность:

09.03.02 «Информационные системы и технологии»

Выполнил обучающийся группы 4119

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_Галиакберов Д.А.

(подпись) (расшифровка подписи)

Руководитель практики от университета

доцент \_\_\_\_\_\_\_\_ Медведев М.В.

(должность) (подпись) (расшифровка подписи)

Отчет защищен с оценкой: \_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты «\_\_\_» \_\_\_\_\_\_\_\_\_\_20\_\_ г.

Казань 2023

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

Обучающегося Галиакберов Данияр Анастасович

(Ф.И.О. полностью)

группы 4119 направления 09.03.02 «Информационные системы и технологии», профиль подготовки: «Информационные системы» Института КТЗИ, период практики с « » 20\_\_ г. по « » 20\_\_ г.

Место прохождения практики:

кафедра автоматизированных систем обработки информации и управления.

(наименование организации, предприятия / наименование подразделения организации, предприятия)

Задание на практику:

Игра «Прятки». На форме присутствуют четыре круга. По нажатию на кнопку в окне появляется прямоугольник, который начинает двигаться в очерченный по его размерам центр окна. Как только прямоугольник достигнет центра, круги должны «убегать» каждый в свою сторону формы.

Задание получил, ознакомлен и согласен:

« » 20\_\_ г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (расшифровка подписи)

Руководитель практики от университета

доцент \_\_\_\_\_\_\_\_\_ Медведев М.В.

(должность) (подпись) (расшифровка подписи)

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc139799702)

[РАЗРАБОТКА ПРИЛОЖЕНИЯ 5](#_Toc139799703)

[ЗАКЛЮЧЕНИЕ 9](#_Toc139799704)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 10](#_Toc139799705)

[ПРИЛОЖЕНИЕ 1 11](#_Toc139799706)

# ВВЕДЕНИЕ

В процессе учебной практики проводилось изучение и практическое применение принципов объектно-ориентированного программирования (ООП), работы с потоками и использования событий. ООП – это подход к разработке программного обеспечения, в котором программа организуется вокруг объектов, которые взаимодействуют друг с другом. Основные принципы ООП включают абстракцию, наследование, полиморфизм и инкапсуляцию. Работа с потоками в программировании позволяет выполнять несколько задач параллельно. Это особенно полезно в случаях, когда требуется эффективное использование ресурсов или обработка больших объемов данных. Использование событий позволяет создавать реактивные системы, где компоненты программы могут реагировать на определенные сигналы или события. Данные концепции являются важными для разработки программного обеспечения, обеспечивая гибкость, эффективность и масштабируемость системы.

Компетенции, формируемые в результате прохождения практики:

ПК-7. Способность проводить анализ требований к программному обеспечению, выполнять работы по проектированию программного обеспечения.

Календарный график прохождения практики

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Срок выполнения этапов прохождения практики | Наименование этапов прохождения практики | Краткое содержание выполненных работ |
| 1 | 29.06 | Получение задания | Анализ поставленной задачи |
| 2 | 30.06 – 07.07 | Разработка приложения | Разработка приложения, с применением ООП и потоков. |
| 3 | 08.07 – 09.07 | Составление отчета | Составление подробного отчета выполненной работы |
| 4 | 12.07 | Сдача отчета | Сдача отчета |

# РАЗРАБОТКА ПРИЛОЖЕНИЯ

Первым этапом проектирования приложения было выделение основных объектов системы. Для поставленной задачи главными объектами системы являются круги: зеленый и красный. Для кругов описан класс Dot, который передает координаты, представленный на рис. 1.

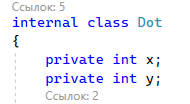


Рисунок 1 – Класс Dot

Данный класс предназначен только для передачи координат при помощи get и set методов классу Circle, в котором и описан основной функционал создания объекта (круга).

Также в этом классе описан метод Move(), отвечающий за перемещение координат на определенное число.

В классе Circle присутствует конструктор, устанавливающий исходное положение круга. Также в этом классе прописаны get методы, а также:

* Move() – предназначен для перемещения круга в форме. В нем прописано условие при котором круг движется, пока не достигнет границы, в противном случае круг будет удален;
* Show() – отвечает за отрисовку круга;
* Clear() – удаляет круг с формы.

Класс представлен на рис. 2.

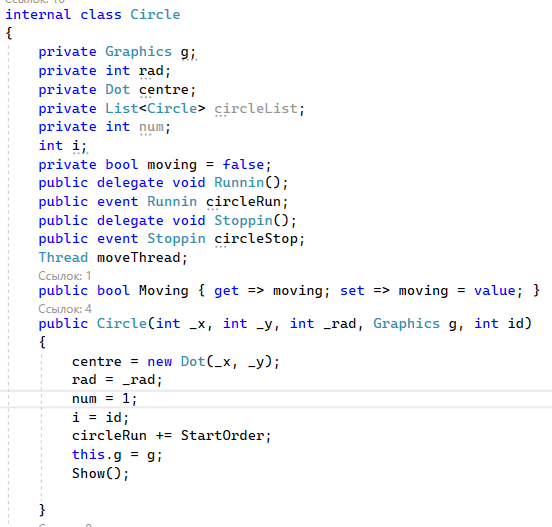


Рисунок 2 – Класс Circle

Класс Rectangle предназначен для отрисовки «прямоугольника» и он содержит в себе методы:

* RectangleMovingStop – останавливает прямоугольник;
* Move – двигает прямоугольник;
* Clear – очищает прямоугольник;
* Show – отрисовывает прямоугольник;

Он представлен на рис. 3.

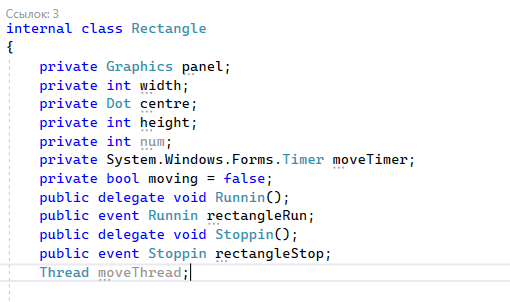
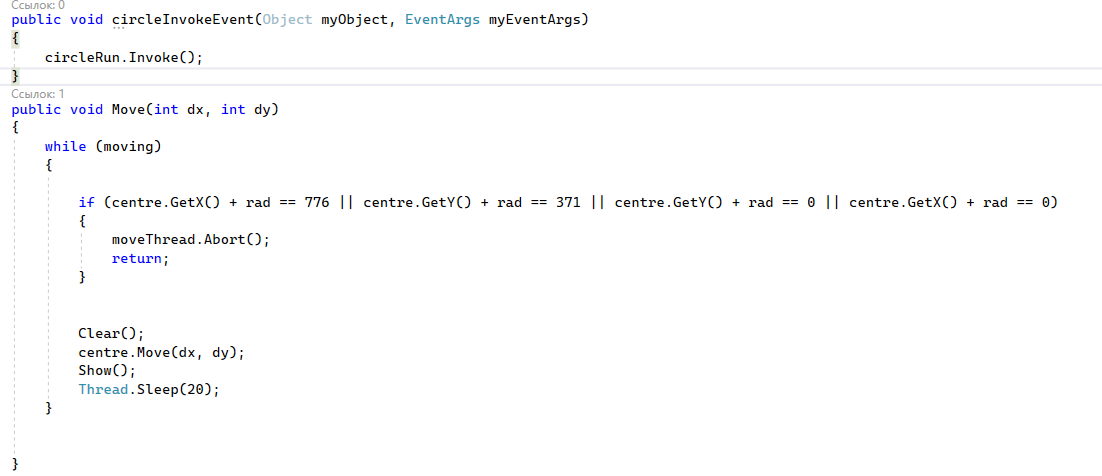


Рисунок 3 – Класс Rectangle

При нажатии на форму создаются круги. При нажатии на кнопку «Запуск» появляется прямоугольник который движется в центр.

При достижении прямоугольником центра круги оповещаются с помощью события (Рис. 4) и начинают разбегаться по разным сторонам формы. Когда круги достигают углов формы они останавливаются.

 Рисунок 4 – Событие перемещения круга

По нажатию на панель, происходит генерация кругов (Рис.5).

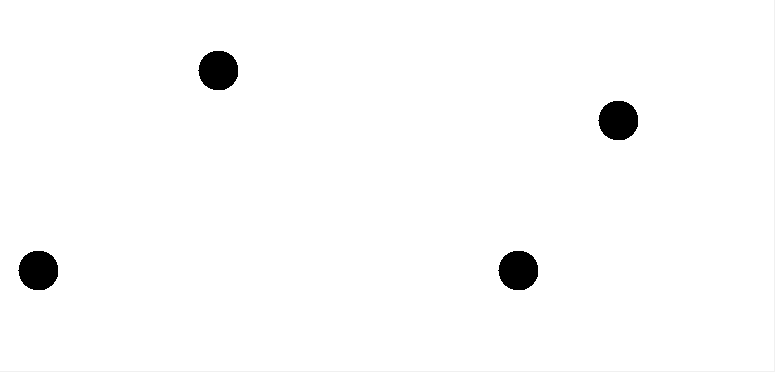


Рисунок 5 – Сгенерированные круги.

Таким образом результат программы представлен на (Рис. 6) Прямоугольник обозначен пшеничным цветом. Круги обозначены черным цветом.

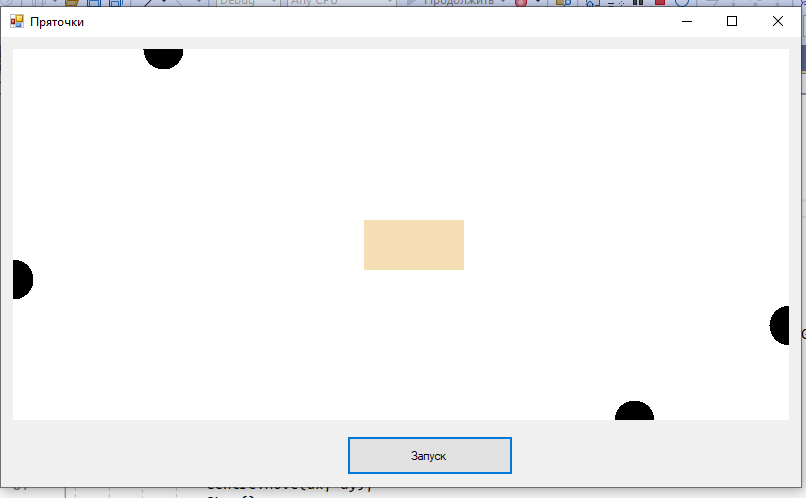


Рисунок 6 – Результат выполнения программы

# ЗАКЛЮЧЕНИЕ

В рамках учебной практики были освоены и применены основные принципы объектно-ориентированного программирования (ООП), механизм работы с потоками и использование событий. Основной задачей было разработать приложение, в котором основным объектом является круг. Для работы с кругом были созданы отдельные классы, в которых была прописана соответствующая логика.

В результате выполненной работы было разработано приложение, имитирующее игру "Прятки". Оно успешно применяет основные принципы ООП, механизмы работы с потоками и использование событий. Полученные знания и навыки могут быть использованы для разработки более сложных и масштабируемых приложений.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Албахари, Д. C# 7.0. Карманный справочник : скорая помощь для программистов на C# 7.0 / Д. Албахари, Б. Албахари; пер. с англ. Ю. Н. Артеменко. – М.: Диалектика, 2017. - 224 с.
2. Троелсен, Э. Язык программирования C# 7 и платформы .NET и .NET Core / Э. Троелсен, Ф. Джепикс; пер. с англ. Ю. Н. Артеменко. – 8-е изд. – М.: Диалектика, 2019. – 1328 с.
3. Рихтер Д. CLR via C# : программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Д. Рихтер; пер. с англ. Е. Матвеев. – 4-е изд. – М.: Питер, 2019. – 895 с.
4. Learn Microsoft [Электронный ресурс]: Документация по C#. – URL: https://learn.microsoft.com/ru-ru/dotnet/csharp/ (дата обращения: 02.07.2023)

# ПРИЛОЖЕНИЕ 1

Код класса Form1:

namespace SummerPractice4119

{

public partial class Form1 : Form

{

Circle circle;

Circle circle1;

Circle circle2;

Rectangle sas;

public static Timer timer;

Circle circle3;

bool started = false;

Graphics g;

public delegate void ListHandler();

public delegate void Stoppin();

public static event Stoppin rectStop;

public static event ListHandler sasun;

public Form1()

{

InitializeComponent();

g = panelMain.CreateGraphics();

}

private void buttonStart\_Click(object sender, EventArgs e)

{

if (!started)

{

sas = new Rectangle(250, 70, g);

sas.Moving = true;

started = true;

timer = new Timer();

timer.Interval = 20;

sas.rectangleStop += circleRun;

timer.Tick += sas.MoveTimerEventX;

timer.Start();

sas.rectangleStop += sas.RectangleMovingStop;

}

}

public void circleRun()

{

timer.Tick -= sas.MoveTimerEventX;

timer.Stop();

circle.StartOrder();

circle1.StartOrder();

circle2.StartOrder();

circle3.StartOrder();

}

private void panelMain\_Click(object sender, EventArgs e)

{

circle = new Circle(600, 100, 20, panelMain.CreateGraphics(),1);

circle1 = new Circle(500, 250, 20, panelMain.CreateGraphics(),1);

circle2 = new Circle(200, 50, 20, panelMain.CreateGraphics(),-1);

circle3 = new Circle(20, 250, 20, panelMain.CreateGraphics(),-1);

}

}

Код класса Dot:

namespace Basketball4VAR

{

internal class Tochka

{

private int x, y;

public Tochka(int \_x, int \_y)

{

x = \_x;

y = \_y;

}

public void Move(int dx, int dy)

{

x += dx;

y += dy;

}

public int GetX()

{

return x;

}

public int GetY()

{

return y;

}

public void SetX(int \_x)

{

x = \_x;

}

public void SetY(int \_y)

{

y = \_y;

}

}

}

Код класса Circle:

namespace SummerPractice4119

{

internal class Circle

{

private Graphics g;

private int rad;

private Dot centre;

private List<Circle> circleList;

private int num;

int i;

private bool moving = false;

public delegate void Runnin();

public event Runnin circleRun;

public delegate void Stoppin();

public event Stoppin circleStop;

Thread moveThread;

public bool Moving { get => moving; set => moving = value; }

public Circle(int \_x, int \_y, int \_rad, Graphics g, int id)

{

centre = new Dot(\_x, \_y);

rad = \_rad;

num = 1;

i = id;

circleRun += StartOrder;

this.g = g;

Show();

}

public int getCircleX()

{

return centre.GetX();

}

public int getCircleY()

{

return centre.GetY();

}

public void circleInvokeEvent(Object myObject, EventArgs myEventArgs)

{

circleRun.Invoke();

}

public void Move(int dx, int dy)

{

while (moving)

{

if (centre.GetX() + rad == 776 || centre.GetY() + rad == 371 || centre.GetY() + rad == 0 || centre.GetX() + rad == 0)

{

moveThread.Abort();

return;

}

Clear();

centre.Move(dx, dy);

Show();

Thread.Sleep(20);

}

}

public void StartOrder()

{

moving = true;

moveThread = new Thread(() => Move(i, i));

moveThread.Start();

}

public void Show()

{

SolidBrush b = new SolidBrush(Color.Black);

g.FillEllipse(b, centre.GetX(), centre.GetY(), 2 \* rad, 2 \* rad);

}

public void Clear()

{

SolidBrush b = new SolidBrush(Color.White);

g.FillEllipse(b, centre.GetX(), centre.GetY(), 2 \* rad, 2 \* rad);

}

public void StopSignal()

{

Moving = false;

}

Код класса Rectangle:

namespace SummerPractice4119

{

internal class Rectangle

{

private Graphics panel;

private int width;

private Dot centre;

private int height;

private int num;

private System.Windows.Forms.Timer moveTimer;

private bool moving = false;

public delegate void Runnin();

public event Runnin rectangleRun;

public delegate void Stoppin();

public event Stoppin rectangleStop;

Thread moveThread;

public bool Moving { get => moving; set => moving = value; }

public void rectangleInvokeEvent()

{

rectangleRun.Invoke();

}

public Rectangle(int \_x, int \_y, Graphics \_panel)

{

centre = new Dot(\_x, \_y);

this.panel = \_panel;

width = 100;

height = 50;

num = 1;

moveTimer = new System.Windows.Forms.Timer();

moveTimer.Interval = 20;

Show();

}

public void RectangleMovingStop()

{

Moving = false;

}

public void Move(int dx, int dy)

{

if (moving == true)

{

if (getRectangleX() <= 330 || getRectangleY() <= 170)

{

Clear();

centre.Move(dx, dy);

Show();

}

else

{

rectangleStop.Invoke();

}

}

}

public void Clear()

{

SolidBrush b = new SolidBrush(Color.White);

panel.FillRectangle(b, centre.GetX(), centre.GetY(), width, height);

}

public void Show()

{

SolidBrush b = new SolidBrush(Color.Wheat);

panel.FillRectangle(b, centre.GetX(), centre.GetY(), width, height);

}

public int getRectangleX()

{

return centre.GetX();

}

public int getRectangleY()

{

return centre.GetY();

}

public void MoveTimerEventX(Object myObject, EventArgs myEventArgs)

{

Move(1, 1);

}

}

**Отзыв ответственного лица от профильной организации**

**о прохождении практики**

Обучающийся Галиакберов Данияр Анастасович

(Ф.И.О. полностью)

группы 4119 ИКТЗИ КНИТУ-КАИ проходил технологическую (учебную) практику с « » 20\_\_\_ по «\_\_»\_\_\_\_\_\_\_\_20\_\_г. на кафедре автоматизированных систем обработки информации и управления КНИТУ-КАИ.

Практика была организована в соответствии с рабочей программой практики.

Кафедра автоматизированных систем обработки информации и управления КНИТУ-КАИ, именно ответственное лицо от профильной организации Ризаев М.И., заведующий лабораторией кафедры автоматизированных систем обработки информации и управления КНИТУ-КАИ, подтверждает участие в формировании следующих компетенций, осваиваемых при прохождении практики:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | Код компетенции | Наименование компетенции | Уровень освоения профессиональной компетенции (5 – наивысший балл) | | | | |
| 1 | 2 | 3 | 4 | 5 |
| 1 | ПК-7 | Способность проводить анализ требований к программному обеспечению, выполнять работы по проектированию программного обеспечения |  |  |  |  |  |

Обучающийся Галиакберов Д.А. зарекомендовал(а) себя как

(Ф.И.О.) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Работу обучающегося Галиакберов Д.А. оцениваю на \_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.) (по5-ти балльной шкале)

Ответственное лицо от профильной организации

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (расшифровка подписи)