Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рязанский государственный радиотехнический университет

имени В.Ф.Уткина»

Кафедра «Электронные вычислительные машины»

Отчет по лабораторной работе №7

на тему

«События»

по дисциплине  
**«Визуальное программирование»**

Выполнили:

Студенты группы №140

Бригада 5

Сафаров Д. А.

Тимохин Е. С.

Проверили:  
ст. преп. Хизриева Н.И.

ст. преп. Бастрычкин А.С.

**Цель работы**

Изучить механизм работы с событиями в языке C#.

**Задание**

Написать программу для симуляции поляны с грибами.

На поляне изначально находится некоторое количество грибов. Как только на поляну приходит грибник, выводится сообщение:  
«На поляну пришёл грибник <имя грибника>. В данный момент   
на поляне N грибов».

Когда грибник собирает X количество грибов, выводится уведомление «Грибник <имя грибника> собрал X грибов».

Когда грибник уходит с поляны, выводится сообщение:   
«Грибник <имя грибника> ушёл с поляны. На поляне осталось M грибов».

Как только все грибы заканчиваются, все грибники уходят с поляны.

**Практическая часть**

Код программы представлен в приложении А. Результат её работы представлен на рисунке 1.

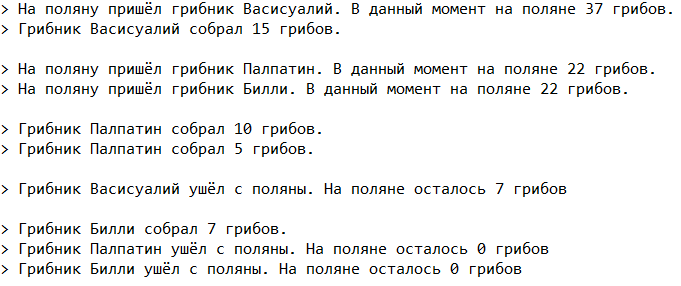


Рисунок 1 – Результаты работы программы

Приложение А. Текст программы

**Nest.cs**

using System.Collections;

using System.Reflection;

using System.Xml.Linq;

using static ShroomsBuzz.Doper;

namespace ShroomsBuzz

{

public class Nest

{

//Количество грибов

private int \_stash;//new Random().Next(1, 50);

// Конструктор без аргументов

public Nest()

{

\_stash = 31;

}

// Конструктор с количеством грибов

public Nest(int stashAmount)

{

if (stashAmount < 0) throw new Exception("Invalid stash amount!");

\_stash = stashAmount;

}

// Событие - на поляне кончились грибы

public static event EventHandler OnDopeOver;

// Обработчик прихода грибника

public void DoperVisitHandler (object sender, EventArgs e)

{

//if (sender == null) throw new Exception(

"DoperVisitHandler: sender is null!");

OnDopeOver += (sender as Doper).DopeOverHandler;

Console.WriteLine($"> На поляну пришёл грибник {(sender as Doper).Name}.

В данный момент на поляне {\_stash} грибов.");

}

// Обработчик сбора грибов

public void GettingDopeHandler(object sender, EventArgs e)

{

int amount = (e as ShroomsEventArgs).ShroomsAmount;

if (amount <= 0) throw new Exception("Invalid dope amount!");

if (\_stash <= amount)

amount = \_stash;

\_stash -= amount;

Console.WriteLine($"> Грибник {(sender as Doper).Name}

собрал {amount} грибов.");

if (\_stash <= 0)

OnDopeOver(this, new EventArgs());

}

// Обработчик ухода грибника

public void DoperLeaveHandler(object sender, EventArgs e)

{

OnDopeOver -= (sender as Doper).DopeOverHandler;

Console.WriteLine($"> Грибник {(sender as Doper).Name} ушёл с поляны.

На поляне осталось {\_stash} грибов");

}

}

}

**Doper.cs**

using System.Text;

namespace ShroomsBuzz

{

public class Doper

{

// Поляна пребывания

private Nest? \_currentNest = null;

// Имя

public string Name { get; private set; }

// Конструктор без аргументов

public Doper()

{

StringBuilder nameBuilder = new("Васян");

nameBuilder.Append(new Random().Next(1000, 9999));

Name = nameBuilder.ToString();

}

// Конструктор с именем

public Doper(string name)

{

Name = name;

}

// Событие – грибник пришёл на поляну

public event EventHandler OnNestVisit;

// Событие – грибник собрал грибочки

public event EventHandler OnDopeGet;

// Событие – грибник покинул поляну

public event EventHandler OnNestLeave;

// Аргументы события сбора грибочков

public class ShroomsEventArgs : EventArgs

{

public int ShroomsAmount;

public ShroomsEventArgs(int shroomsAmount) =>

this.ShroomsAmount = shroomsAmount;

}

// Прийти на полянку

public void VisitNest(Nest nest)

{

if (\_currentNest != null) throw new Exception(

"Doper already in the nest!");

//nest.AdmitDoper(this);

this.\_currentNest = nest;

OnNestVisit += \_currentNest.DoperVisitHandler;

OnDopeGet += \_currentNest.GettingDopeHandler;

OnNestLeave += \_currentNest.DoperLeaveHandler;

OnNestVisit?.Invoke(this, new EventArgs());

}

// Собрать грибочки

public void GetDope(int amount)

{

if (\_currentNest == null) throw new Exception(

"This doper not in the nest!");

OnDopeGet?.Invoke(this, new ShroomsEventArgs(amount));

}

// Уйти с полянки

public void LeaveNest()

{

if (\_currentNest == null) throw new Exception(

"This doper not in the nest!");

OnNestLeave?.Invoke(this, new EventArgs());

OnNestVisit -= \_currentNest.DoperVisitHandler;

OnDopeGet -= \_currentNest.GettingDopeHandler;

OnNestLeave -= \_currentNest.DoperLeaveHandler;

\_currentNest = null;

}

// Обработчик события – на поляне кончились грибы

public void DopeOverHandler(object sender, EventArgs e)

{

if (\_currentNest != (sender as Nest)) return;

LeaveNest();

}

}

}

**Program.cs**

using ShroomsBuzz;

Nest nestDeep = new Nest(37);

Doper doperVasya = new("Васисуалий");

Doper doperSenya = new("Палпатин");

Doper doperEgor = new("Билли");

doperVasya.VisitNest(nestDeep);

doperVasya.GetDope(15);

Console.WriteLine();

doperSenya.VisitNest(nestDeep);

doperEgor.VisitNest(nestDeep);

Console.WriteLine();

doperSenya.GetDope(10);

doperSenya.GetDope(5);

Console.WriteLine();

doperVasya.LeaveNest();

Console.WriteLine();

doperEgor.GetDope(12);