|  |  |
| --- | --- |
| *voenmeh* | МИНОБРНАУКИ РОССИИ  федеральное государственное бюджетное образовательное учреждение  высшего образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»**  **(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** |
| БГТУ.СМК-Ф-4.2-К5-02 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Факультет |  | О |  | Естественнонаучный |
|  |  | шифр |  | наименование |
| Кафедра |  | О7 |  | Информационные системы и программная инженерия |
|  |  | шифр |  | наименование |
| Дисциплина |  | Информационные технологии и программирование | | |

|  |
| --- |
| ПРАКТИЧЕСКАЯ РАБОТА №1 |
| Классы |
| Вариант №24 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы | | | |  | И924Б |
| Фокин В. К. | | | | | |
| Фамилия И.О. | | | | | |
| **Преподаватель:** | | | | | |
| Гладевич А. А. | |  |  | | |
| Фамилия И.О. Подпись | | | | | |
| Оценка |  | | | |  |
| «\_\_\_\_\_» |  | | | | 2023г. |

САНКТ-ПЕТЕРБУРГ

2023 г.

**Постановка задачи**

Работа состоит из последовательного выполнения трёх заданий: создание класса, иерархии классов и интерфейса. Все задания выполняются последовательно в одном файле с исходным текстом в рамках одного проекта. В отчёте к работе должны быть отдельно отражены результаты выполнения каждого задания.

**Вариант 24.** Цветы**.**

Класс для первой части – роза.

Варианты свойств: сорт, цвет, здоровье, цена.

Варианты методов: вдохнуть аромат, полить, насладиться, узнать, как ухаживать (статический).

Возможные классы иерархии: цветок(базовый), гладиолус, пион, ромашка.

Возможный интерфейс: ISmell, дополнительный класс – духи.

**Задание 1**

Разработать класс, согласно индивидуальному варианту, содержащий:

– элементы разного уровня доступа (public и private);

– не менее четырех свойств;

– не менее трех методов;

– перегрузку метода ToString();

– статический метод;

– константное или поле только для чтения;

– не менее трех конструкторов;

– перегрузку операции сравнения и одной любой арифметической.

Рекомендуемые поля и методы указаны в варианте. Также необходимо написать программу с меню, позволяющую протестировать разработанный класс.

Обязательные пункты меню:

– задание параметров конструируемого объекта;

– вывод свойств объекта;

– выполнение статического метода;

– выполнение методов объекта.

**Текст программы**

using System;

namespace pr1

{

class Rose

{

//элементы разного доступа

private string name;

private string sort;

private string color;

private string shape;

public int height;

//константное поле

public const string baseColor = "Красный";

//свойства

public string Name

{

set { name = value; }

get { return name; }

}

public string Sort

{

set { sort = value; }

get { return sort; }

}

public string Color

{

set { color = value; }

get { return color; }

}

public string Shape

{

set { shape = value; }

get { return shape; }

}

//конструкторы

public Rose() //по умолчанию

{

name = "Ирина";

sort = "Мисс Пигги";

color = baseColor;

shape = "Розетковидная";

height = 50;

}

public Rose(string Name, string Sort, string Color, string Shape, int Height) //с параметрами

{

name = Name;

sort = Sort;

color = Color;

shape = Shape;

height = Height;

}

public Rose(Rose r) //копирования

{

name = r.name;

sort = r.sort;

color = r.color;

shape = r.shape;

height = r.height;

}

//методы

public void smellThat()

{

Console.WriteLine($"Вы вдыхаете аромат розы {name}. Сорт - {sort}");

Console.WriteLine("\*пахнет, как дух юности...\*");

}

public void water()

{

Console.WriteLine($"Роза {name} полита.");

Console.WriteLine("\*это наполняет Вас решимостью\*");

}

public void stroke()

{

Console.WriteLine($"Вы погладили розу {name}.");

Console.WriteLine("В ответ она уколола Вас.");

}

//статический метод

public static void wiki()

{

Console.WriteLine("-----------ИНСТРУКЦИЯ-----------");

Console.WriteLine("| Как ухаживать за розой |");

Console.WriteLine("--------------------------------");

Console.WriteLine("| 1. Сначала розу нужно полить |");

Console.WriteLine("| 2. Затем погладить |");

Console.WriteLine("| 3. В конце ее нужно понюхать |");

Console.WriteLine("| 4. Вуаля! Она довольна :) |");

Console.WriteLine("--------------------------------");

}

//перегрузка ToString

public override string ToString()

{

return $"Имя: {name}\nСорт: {sort}\nЦвет: {color}\nФорма: {shape}\nВысота: {height} см";

}

//перегрузка операторов сравнения

public static bool operator >(Rose r1, Rose r2)

{

if (r1.height > r2.height)

{

return true;

}

else

{

return false;

}

}

public static bool operator <(Rose r1, Rose r2)

{

if (r1.height < r2.height)

{

return true;

}

else

{

return false;

}

}

//перегрузка оператора сложения

public static Rose operator +(Rose left, Rose right)

{

Rose result = new Rose();

result.name = left.name + " & " + right.name;

result.sort = left.sort + " & " + right.sort;

result.color = left.color + " & " + right.color;

result.shape = left.shape + " & " + right.shape;

result.height = left.height + right.height;

return result;

}

}

class Program

{

static void Main(string[] args)

{

int menu, MENU;

int cnt = 3, h = 0, high, num;

string n, sr, c, sh;

bool integer = true;

Rose[] arrRose = new Rose[20];

arrRose[0] = new Rose("Наталья", "Пиано Ред", "Желтый", "Плоская", 43);

arrRose[1] = new Rose("Владлена", "Ламбада", "Розовый", "Кувшинчатая", 30);

arrRose[2] = new Rose("Лидия", "Лидия", "Оранжевый", "Помпонная", 70);

arrRose[3] = new Rose("Анатолий", "Пимпинеллифолия", "Небесный", "Крестовидно-розеточная", 80);

do

{

Console.WriteLine("-----------------------------Меню-----------------------------");

Console.WriteLine("| 1. Добавить розу |");

Console.WriteLine("| 2. Оглядеться и посмотреть какие розы есть вокруг |");

Console.WriteLine("| 3. Инструкция как сохранить розе отличное настроение |");

Console.WriteLine("| 4. Сделать розе приятно |");

Console.WriteLine("| 0. Сбежать от роз |");

Console.WriteLine("--------------------------------------------------------------");

if (int.TryParse(Console.ReadLine(), out menu))

{

switch (menu)

{

case 1:

Console.Clear();

Console.WriteLine("Введите имя вашего цветочка: ");

n = Console.ReadLine();

Console.WriteLine("Введите сорт: ");

sr = Console.ReadLine();

Console.WriteLine("Введите цвет: ");

c = Console.ReadLine();

Console.WriteLine("Введите форму: ");

sh = Console.ReadLine();

do

{

Console.WriteLine("Введите высоту: ");

if (int.TryParse(Console.ReadLine(), out high) && high <= 80 && high >= 30)

{

h = high;

integer = true;

}

else

{

integer = false;

Console.WriteLine("Введите целое значение от 30 до 80 в сантиметрах");

}

} while (integer == false);

cnt++;

arrRose[cnt] = new Rose(n, sr, c, sh, h);

break;

case 2:

Console.Clear();

Console.WriteLine("--------------------------------");

for (int i = 0; arrRose[i] != null; i++)

{

Console.WriteLine(i + 1 + ". " + arrRose[i].ToString());

Console.WriteLine("--------------------------------");

}

break;

case 3:

Console.Clear();

Rose.wiki();

break;

case 4:

Console.Clear();

do

{

Console.WriteLine("--------------------------");

Console.WriteLine("| 1. Полить розу |");

Console.WriteLine("| 2. Понюхать розу |");

Console.WriteLine("| 3. Погладить розу |");

Console.WriteLine("| 0. Вернуться |");

Console.WriteLine("--------------------------");

if (int.TryParse(Console.ReadLine(), out MENU))

{

switch (MENU)

{

case 1:

Console.Clear();

Console.WriteLine("--------------------------------");

for (int i = 0; arrRose[i] != null; i++)

{

Console.WriteLine(i + 1 + ". " + arrRose[i].ToString());

Console.WriteLine("--------------------------------");

}

do

{

Console.WriteLine("Введите номер розы, котороую нужно полить: ");

if (int.TryParse(Console.ReadLine(), out num) && num <= cnt + 1 && num > 0)

{

int i = num;

arrRose[i - 1].water();

integer = true;

}

else

{

integer = false;

Console.WriteLine("Цветка под данным номером нет");

}

} while (integer == false);

break;

case 2:

Console.Clear();

Console.WriteLine("--------------------------------");

for (int i = 0; arrRose[i] != null; i++)

{

Console.WriteLine(i + 1 + ". " + arrRose[i].ToString());

Console.WriteLine("--------------------------------");

}

do

{

Console.WriteLine("Введите номер розы, котороую нужно нюхнуть: ");

if (int.TryParse(Console.ReadLine(), out num) && num <= cnt + 1 && num > 0)

{

int i = num;

arrRose[i - 1].smellThat();

integer = true;

}

else

{

integer = false;

Console.WriteLine("Цветка под данным номером нет");

}

} while (integer == false);

break;

case 3:

Console.Clear();

Console.WriteLine("--------------------------------");

for (int i = 0; arrRose[i] != null; i++)

{

Console.WriteLine(i + 1 + ". " + arrRose[i].ToString());

Console.WriteLine("--------------------------------");

}

do

{

Console.WriteLine("Введите номер розы, котороую нужно погладить: ");

if (int.TryParse(Console.ReadLine(), out num) && num <= cnt + 1 && num > 0)

{

int i = num;

arrRose[i - 1].stroke();

integer = true;

}

else

{

integer = false;

Console.WriteLine("Цветка под данным номером нет");

}

} while (integer == false);

break;

case 0:

Console.Clear();

break;

default:

Console.WriteLine("Пункта меню под данным номером не существует");

break;

}

}

} while (MENU != 0);

break;

case 0:

Console.Clear();

break;

default:

Console.WriteLine("Пункта меню под данным номером не существует");

break;

}

}

} while (menu != 0);

}

}

}

**Результат работы программы**

При запуске программы появляется главное меню, что представлено на рисунке 1.

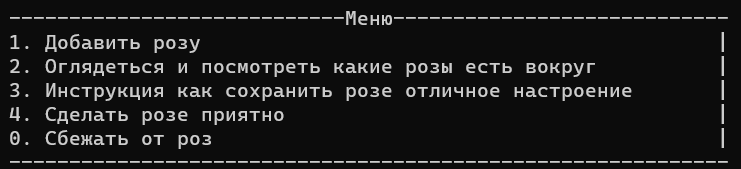


Рисунок 1 − Меню программы

При выборе пункта «1» можно добавить розу. Нужно ввести ее имя, сорт, цвет, форму и высоту в сантиметрах, что показано на рисунке 2.

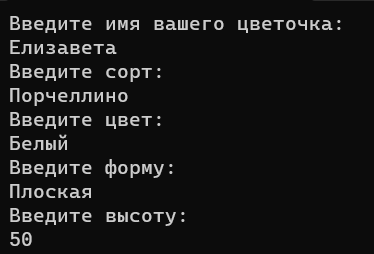


Рисунок 2 − Добавление нового объекта

При попытке ввести буквы в высоту, появится ошибка и программа попросит еще раз произвести ввод цифрами, что показано на рисунке 3.

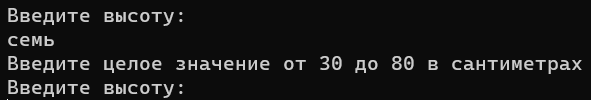


Рисунок 3 − Ошибка при вводе

При выборе пункта «2» можно посмотреть список всех имеющихся роз, что показано на рисунке 4.

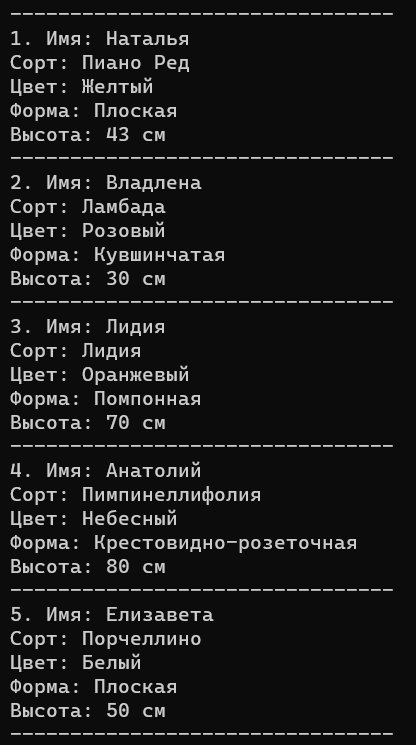


Рисунок 4 − Список всех объектов

При выборе пункта «3» можно вызвать статический метод, что показано на рисунке 5.

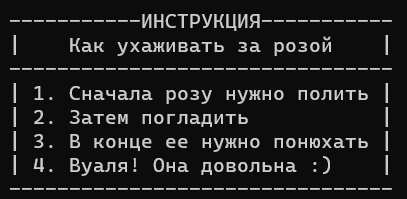


Рисунок 5 − Статический метод

При выборе пункта «4» можно вызвать методы класса, что показано на рисунке 6.

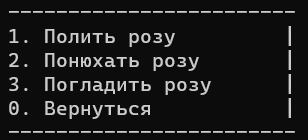


Рисунок 6 − Вызов методов класса

При выборе пункта «1» в меню методов предоставляется возможность выбрать розу и полить ее, что показано на рисунках 7 и 8.

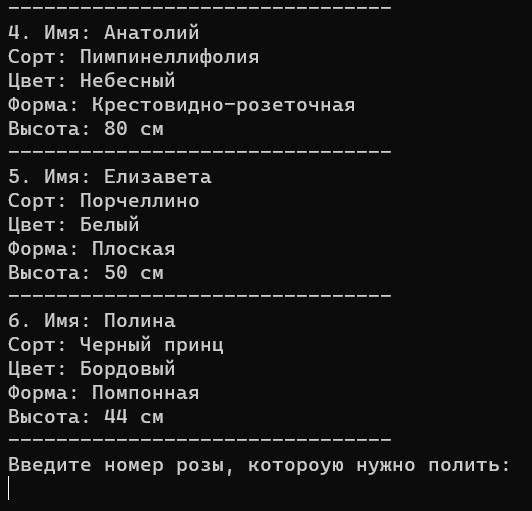


Рисунок 7 − Выбор розы

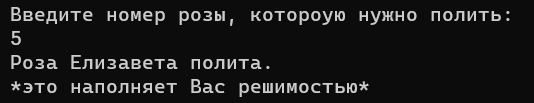


Рисунок 8 − Результат работы метода

При выборе пункта «2» в меню методов предоставляется возможность выбрать розу и понюхать ее, что показано на рисунках 9 и 10.

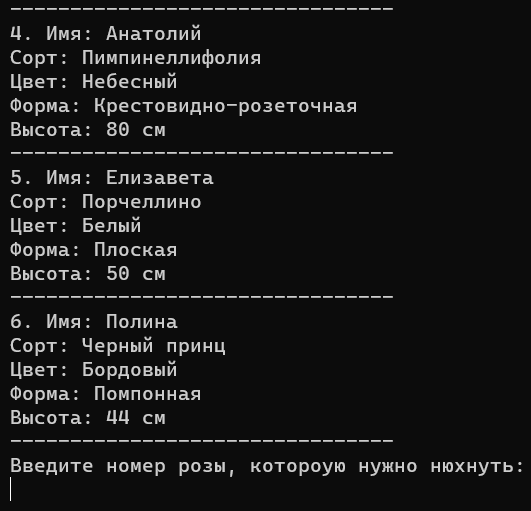


Рисунок 9 − Выбор розы

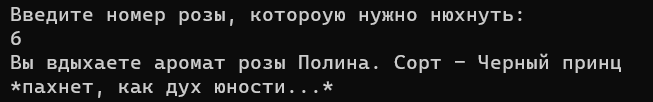


Рисунок 10 − Результат работы метода

При выборе пункта «3» в меню методов предоставляется возможность выбрать розу и погладить ее, что показано на рисунках 11 и 12.

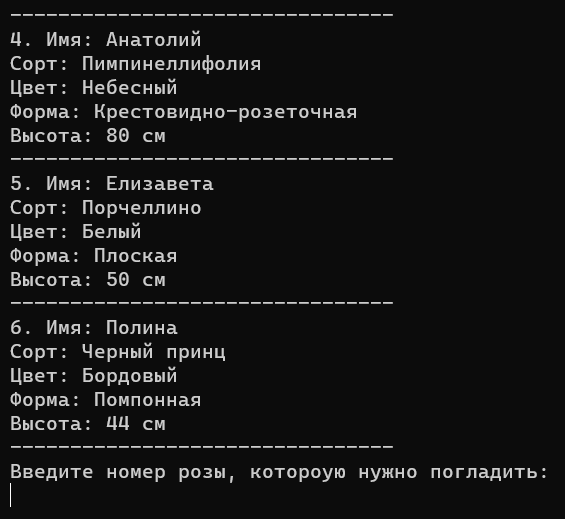


Рисунок 11 − Выбор розы

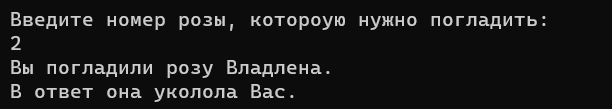


Рисунок 12 − Результат работы метода

При выборе несуществующего пункта меню выводится сообщение об ошибке.

При выборе пункта «0» в меню методов, оно закрывается. При выборе этого пункта в главном меню, программа прекращает работу.

**Задание 2**

Разработать иерархию классов, включающую созданный в задании 1. В иерархию должно входить минимум пять классов. Вновь добавленный корневой класс иерархии (базовый) должен быть абстрактным и содержать общие для остальных поля и методы, а хотя бы один класс из потомков – sealed. У каждого из классов должно быть хотя бы одно собственное свойство и метод. Также должны быть продемонстрированы виртуальные и переопределённые методы. Переопределенные должны вызывать методы базового класса если это оправдано. В каждом классе должен быть метод, выводящий имя данного класса. В объектах дочерних классов он должен вызывать аналогичный метод родительского класса.

Необходимо написать программу с меню, позволяющую протестировать разработанную иерархию. Обязательные пункты меню:

– задание свойств каждого объекта (хотя бы по одному объекту на не абстрактный класс);

– вывод свойств объекта;

– выполнение методов объекта;

– вывод имени класса объекта.

**Текст программы**

using System;

namespace pr1

{

abstract class flower

{

protected string name;

protected string sort;

protected string color;

protected string shape;

//свойства

public string Name

{

set { name = value; }

get { return name; }

}

public string Sort

{

set { sort = value; }

get { return sort; }

}

public string Color

{

set { color = value; }

get { return color; }

}

public string Shape

{

set { shape = value; }

get { return shape; }

}

//конструкторы

public flower() //по умолчанию

{

name = "";

sort = "";

color = "";

}

public flower(string Name, string Sort, string Color) //с параметрами

{

name = Name;

sort = Sort;

color = Color;

}

public flower(flower f) //копирования

{

name = f.name;

sort = f.sort;

color = f.color;

}

//виртуальный метод

public virtual string getClass()

{

return "Цветок";

}

//абстрактный метод

public abstract string toString();

//другие методы

public void smell()

{

Console.WriteLine($"Цветок класса {this.getClass()} по имени {name} был понюхан");

}

public void water()

{

Console.WriteLine($"Цветок класса {this.getClass()} по имени {name} облился");

}

public void stroke()

{

Console.WriteLine($"Цветок класса {this.getClass()} по имени {name} стал выглаженым");

}

}

class rose : flower

{

private int height;

//свойство

public int Height

{

set { height = value; }

get { return height; }

}

//конструкторы

public rose() : base() //по умолчанию

{

height = 0;

}

public rose(string Name, string Sort, string Color, string Shape, int H) : base(Name, Sort, Color) //с параметрами

{

height = H;

}

public rose(rose r) : base(r) //копирования

{

height = r.height;

}

//переопределенный вирутальный метод

public override string getClass()

{

return "Роза";

}

//переопределенный абстрактный метод

public override string toString()

{

return $"Имя: {name}\nСорт: {sort}\nЦвет: {color}\nФорма: {shape}\nВысота: {height} см";

}

//собственный метод

public void injection()

{

Console.WriteLine("Эта роза решила приКОЛОТЬся :))");

}

}

class tulip : flower

{

private int bugs;

//свойство

public int Bugs

{

set { bugs = value; }

get { return bugs; }

}

//конструкторы

public tulip() : base() //по умолчанию

{

bugs = 0;

}

public tulip(string Name, string Sort, string Color, string Shape, int Bugs) : base(Name, Sort, Color) //с параметрами

{

bugs = Bugs;

}

public tulip(tulip t) : base(t) //копирования

{

bugs = t.bugs;

}

//переопределенный вирутальный метод

public override string getClass()

{

return "Тюльпан";

}

//переопределенный абстрактный метод

public override string toString()

{

return $"Имя: {name}\nСорт: {sort}\nЦвет: {color}\nФорма: {shape}\nЖучков на цветке: {bugs}";

}

//собственный метод

public void pollinate()

{

Console.WriteLine("Тюльпан помер от аллергии на пыль..");

}

}

class chamomile : flower

{

private int petals;

//свойство

public int Petals

{

set { petals = value; }

get { return petals; }

}

//конструкторы

public chamomile() : base() //по умолчанию

{

petals = 0;

}

public chamomile(string Name, string Sort, string Color, string Shape, int Petals) : base(Name, Sort, Color) //с параметрами

{

petals = Petals;

}

public chamomile(chamomile c) : base(c) //копирования

{

petals = c.petals;

}

//переопределенный вирутальный метод

public override string getClass()

{

return "Ромашка";

}

//переопределенный абстрактный метод

public override string toString()

{

return $"Имя: {name}\nСорт: {sort}\nЦвет: {color}\nФорма: {shape}\nЛепестков: {petals}";

}

//собственный метод

public void sing\_a\_song()

{

Console.WriteLine("-----------------------------------------------------");

Console.WriteLine("| Ромашки |");

Console.WriteLine("-----------------------------------------------------");

Console.WriteLine("| И сотни роз не стоят ни одной твоей ромашки |");

Console.WriteLine("| Ромашки летят в костёр |");

Console.WriteLine("| Все ночи без тебя – они так тяжки |");

Console.WriteLine("| Мне тяжело одной |");

Console.WriteLine("| Нашу любовь не купишь за бумажки |");

Console.WriteLine("| Она дороже всех твоих духов |");

Console.WriteLine("| Ведь сотни роз не стоят и ромашки, ромашки |");

Console.WriteLine("| |");

Console.WriteLine("| Если любить, то сердцем, а не глазами |");

Console.WriteLine("| Губы соврут, но душу ты не обманешь |");

Console.WriteLine("| Не ранишь больно, но стреляешь словами, словами |");

Console.WriteLine("| О, да |");

Console.WriteLine("| Это про меня |");

Console.WriteLine("| Ты не променял мой, мой океан |");

Console.WriteLine("| И о, да |");

Console.WriteLine("| Ты не променял |");

Console.WriteLine("| |");

Console.WriteLine("| И сотни роз не стоят ни одной твоей ромашки |");

Console.WriteLine("| Ромашки летят в костёр |");

Console.WriteLine("| Все ночи без тебя – они так тяжки |");

Console.WriteLine("| Мне тяжело одной |");

Console.WriteLine("| Нашу любовь не купишь за бумажки |");

Console.WriteLine("| Она дороже всех твоих духов |");

Console.WriteLine("| Ведь сотни роз не стоят и ромашки, ромашки |");

Console.WriteLine("| |");

Console.WriteLine("| И сотни роз не стоят ни одной твоей ромашки |");

Console.WriteLine("| Ромашки летят в костёр |");

Console.WriteLine("| Все ночи без тебя – они так тяжки |");

Console.WriteLine("| Мне тяжело одной |");

Console.WriteLine("| Нашу любовь не купишь за бумажки |");

Console.WriteLine("| Она дороже всех твоих духов |");

Console.WriteLine("| Ведь сотни роз не стоят и ромашки, ромашки |");

Console.WriteLine("-----------------------------------------------------");

}

}

sealed class peonies : flower

{

private double price;

//свойство

public double Price

{

set { price = value; }

get { return price; }

}

//конструкторы

public peonies() : base() //по умолчанию

{

price = 0;

}

public peonies(string Name, string Sort, string Color, string Shape, double Price) : base(Name, Sort, Color) //с параметрами

{

price = Price;

}

public peonies(peonies p) : base(p) //копирования

{

price = p.price;

}

//переопределенный вирутальный метод

public override string getClass()

{

return "Пион";

}

//переопределенный абстрактный метод

public override string toString()

{

return $"Имя: {name}\nСорт: {sort}\nЦвет: {color}\nФорма: {shape}\nЦена: {price}";

}

//собственный метод

public void gift()

{

Console.WriteLine("Эти пионы попали в хорошие руки!");

}

}

class Program

{

static void Main(string[] args)

{

int menu, MENU, umen, UMEN;

int cnt = 3, h = 0;

double p = 0;

string n, s, c, sh;

bool integer = true;

flower[] flowers = new flower[20];

flowers[0] = new rose("Наташа", "Пиано Ред", "Желтый", "Плоская", 45);

flowers[1] = new tulip("Ирка", "Принц Карнавал", "Красный", "Лилиецветная", 13);

flowers[2] = new chamomile("Роза", "Персидская", "Розовый", "Крестовидно-розеточная", 17);

flowers[3] = new peonies("Владлена", "Дюшес де Немур", "Розовый", "Помпонная", 500);

do

{

Console.WriteLine("-----------------------------Меню-----------------------------");

Console.WriteLine("| 1. Добавить цветок |");

Console.WriteLine("| 2. Посмотреть список цветочков |");

Console.WriteLine("| 3. Ухаживать за цветами |");

Console.WriteLine("| 4. Совершить кое-какие действия над цветочками... |");

Console.WriteLine("| 0. Выход |");

Console.WriteLine("--------------------------------------------------------------");

if (int.TryParse(Console.ReadLine(), out menu))

{

switch (menu)

{

case 1:

Console.Clear();

do

{

Console.WriteLine("----------------------------------");

Console.WriteLine("| Какой цветок добавить? |");

Console.WriteLine("| 1. Роза |");

Console.WriteLine("| 2. Тюльпан |");

Console.WriteLine("| 3. Ромашка |");

Console.WriteLine("| 4. Пион |");

Console.WriteLine("| 0. Выход |");

Console.WriteLine("----------------------------------");

if (int.TryParse(Console.ReadLine(), out MENU))

{

switch (MENU)

{

case 1:

Console.Clear();

Console.WriteLine("Введите имя для цветка: ");

n = Console.ReadLine();

Console.WriteLine("Введите сорт: ");

s = Console.ReadLine();

Console.WriteLine("Введите цвет: ");

c = Console.ReadLine();

Console.WriteLine("Введите форму: ");

sh = Console.ReadLine();

do

{

Console.WriteLine("Введите высоту в сантиметрах: ");

if (int.TryParse(Console.ReadLine(), out int high) && high <= 80 && high >= 30)

{

h = high;

integer = true;

}

else

{

integer = false;

Console.WriteLine("Введите целое значение от 30 до 80 в сантиметрах");

}

} while (integer == false);

cnt++;

flowers[cnt] = new rose(n, s, c, sh, h);

break;

case 2:

Console.Clear();

Console.WriteLine("Введите имя для цветка: ");

n = Console.ReadLine();

Console.WriteLine("Введите сорт: ");

s = Console.ReadLine();

Console.WriteLine("Введите цвет: ");

c = Console.ReadLine();

Console.WriteLine("Введите форму: ");

sh = Console.ReadLine();

do

{

Console.WriteLine("Введите количество жучков на цветке: ");

if (int.TryParse(Console.ReadLine(), out int bg) && bg <= 30 && bg >= 0)

{

h = bg;

integer = true;

}

else

{

integer = false;

Console.WriteLine("Введите целое значение от 0 до 30");

}

} while (integer == false);

cnt++;

flowers[cnt] = new tulip(n, s, c, sh, h);

break;

case 3:

Console.Clear();

Console.WriteLine("Введите имя для цветка: ");

n = Console.ReadLine();

Console.WriteLine("Введите сорт: ");

s = Console.ReadLine();

Console.WriteLine("Введите цвет: ");

c = Console.ReadLine();

Console.WriteLine("Введите форму: ");

sh = Console.ReadLine();

do

{

Console.WriteLine("Введите количество лепестков: ");

if (int.TryParse(Console.ReadLine(), out int pl) && pl <= 20 && pl >= 5)

{

h = pl;

integer = true;

}

else

{

integer = false;

Console.WriteLine("Введите целое значение от 5 до 20");

}

} while (integer == false);

cnt++;

flowers[cnt] = new chamomile(n, s, c, sh, h);

break;

case 4:

Console.Clear();

Console.WriteLine("Введите имя для цветка: ");

n = Console.ReadLine();

Console.WriteLine("Введите сорт: ");

s = Console.ReadLine();

Console.WriteLine("Введите цвет: ");

c = Console.ReadLine();

Console.WriteLine("Введите форму: ");

sh = Console.ReadLine();

do

{

Console.WriteLine("Введите цену: ");

if (double.TryParse(Console.ReadLine(), out double doub) && doub <= 700 && doub >= 100)

{

p = doub;

integer = true;

}

else

{

integer = false;

Console.WriteLine("Введите значение от 100 до 700");

}

} while (integer == false);

cnt++;

flowers[cnt] = new peonies(n, s, c, sh, p);

break;

case 0:

Console.Clear();

break;

default:

Console.WriteLine("Пункта меню под данным номером не существует");

break;

}

}

} while (MENU != 0);

break;

case 2:

Console.Clear();

Console.WriteLine("--------------------------------");

for (int i = 0; flowers[i] != null; i++)

{

Console.WriteLine(i + 1 + ". Класс: " + flowers[i].getClass() + "\n" + flowers[i].toString());

Console.WriteLine("--------------------------------");

}

break;

case 3:

Console.Clear();

do

{

Console.WriteLine("----------------------------------");

Console.WriteLine("| 1. Понюхать |");

Console.WriteLine("| 2. Полить |");

Console.WriteLine("| 3. Погладить |");

Console.WriteLine("| 0. Выход |");

Console.WriteLine("----------------------------------");

if (int.TryParse(Console.ReadLine(), out umen))

{

switch (umen)

{

case 1:

Console.Clear();

for (int i = 0; flowers[i] != null; i++)

{

flowers[i].smell();

}

break;

case 2:

Console.Clear();

for (int i = 0; flowers[i] != null; i++)

{

flowers[i].water();

}

break;

case 3:

Console.Clear();

for (int i = 0; flowers[i] != null; i++)

{

flowers[i].stroke();

}

break;

case 0:

Console.Clear();

break;

default:

Console.WriteLine("Пункта меню под данным номером не существует");

break;

}

}

} while (umen != 0);

break;

case 4:

Console.Clear();

do

{

Console.WriteLine("----------------------------------");

Console.WriteLine("| Какие цветы пощупать?? |");

Console.WriteLine("| 1. Потрогать розы |");

Console.WriteLine("| 2. Опылить тюльпаны |");

Console.WriteLine("| 3. Спеть про ромашки |");

Console.WriteLine("| 4. Подарить пионы |");

Console.WriteLine("| 0. Выход |");

Console.WriteLine("----------------------------------");

if (int.TryParse(Console.ReadLine(), out UMEN))

{

switch (UMEN)

{

case 1:

rose ros = new rose();

ros.injection();

break;

case 2:

tulip tulp = new tulip();

tulp.pollinate();

break;

case 3:

chamomile cham = new chamomile();

cham.sing\_a\_song();

break;

case 4:

peonies peon = new peonies();

peon.gift();

break;

case 0:

Console.Clear();

break;

default:

Console.WriteLine("Пункта меню под данным номером не существует");

break;

}

}

} while (UMEN != 0);

break;

case 0:

Console.Clear();

break;

default:

Console.WriteLine("Пункта меню под данным номером не существует");

break;

}

}

} while (menu != 0);

}

}

}

**Результат работы программы**

При запуске программы появляется главное меню, что показано на рисунке 13.

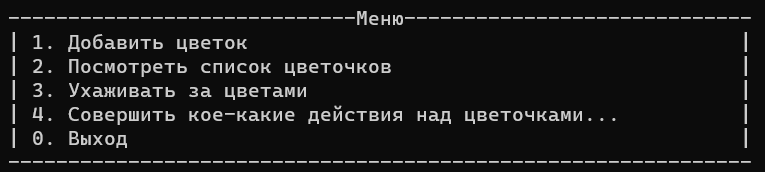


Рисунок 13 − Главное меню

При выборе пункта «1» в главном меню можно добавить цветок. Нужно выбрать в какой класс добавить объект, что показано на рисунке 14.

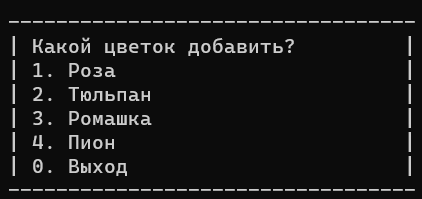


Рисунок 14 − Выбор объекта при добавлении

При выборе пункта «1» в данном меню нужно ввести имя, сорт, цвет, форму и высоту в сантиметрах. При вводе высоты меньше 30 см и больше 80 см появляется ошибка, что показано на рисунке 15.

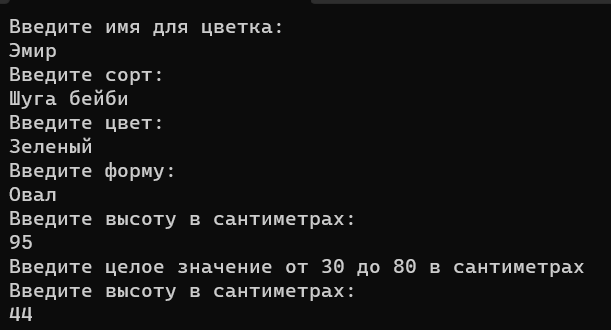


Рисунок 15 − Добавление розы

При выборе пункта «2» в данном меню нужно ввести имя, сорт, цвет, форму и количество жучков. При вводе количества жучков меньше 0 и больше 30 появляется ошибка, что показано на рисунке 16.

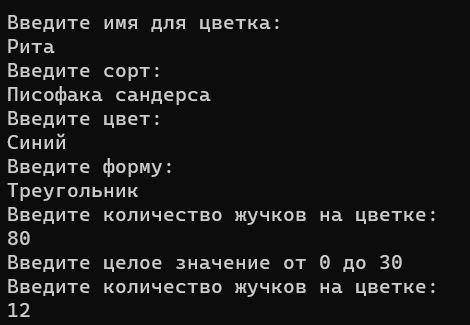


Рисунок 16 − Добавление тюльпана

При выборе пункта «3» в данном меню нужно ввести имя, сорт, цвет, форму и количество лепестков. При вводе количества жучков меньше 5 и больше 20 появляется ошибка, что показано на рисунке 17.

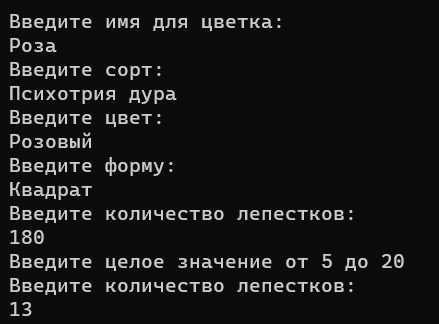


Рисунок 17 − Добавление ромашки

При выборе пункта «4» в данном меню нужно ввести имя, сорт, цвет и цену. При вводе цены меньше 100 и больше 700 появляется ошибка, что показано на рисунке 18.

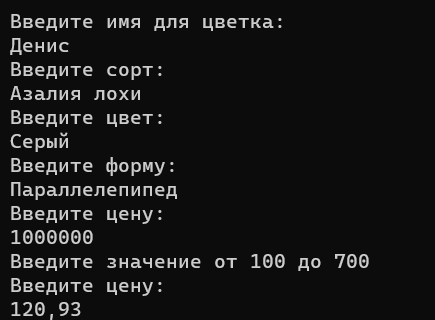


Рисунок 18 − Добавление пионов

При выборе несуществующего пункта меню появится сообщение об этом, что показано на рисунке 19.

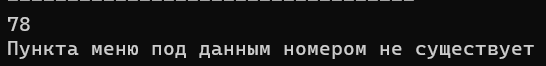


Рисунок 19 − Выбор несуществующего пункта меню

При выборе пункта «0» происходит выход в главное меню.

При выборе пункта «2» в главном меню можно посмотреть список объектов и информацию о каждом, что показано на рисунке 20.

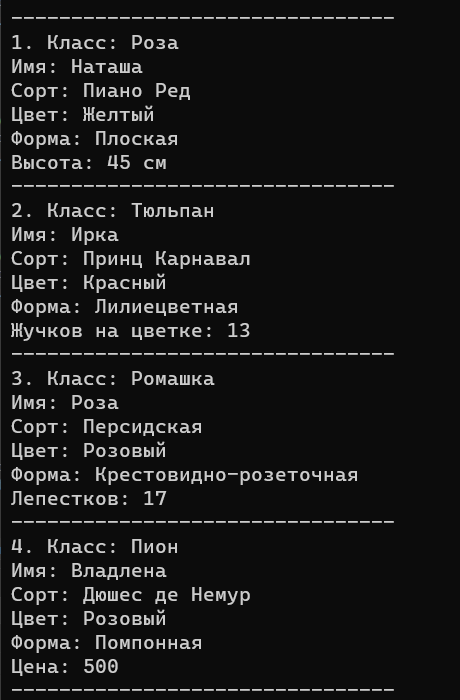


Рисунок 20 − Вывод списка объектов

При выборе пункта «3» в главном меню можно запустить общие методы, что показано на рисунке 21.

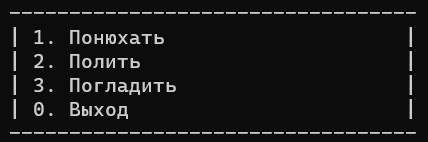


Рисунок 21 − Запуск общих методов

При выборе пункта «1» в данном меню выполнится общий метод smell(). Метод выполняется для всех объектов, что показано на рисунке 22.

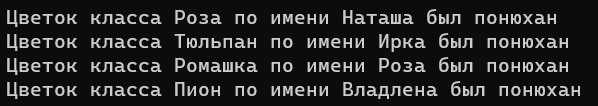


Рисунок 22 − Метод smell()

При выборе пункта «2» в данном меню выполнится общий метод water(). Метод выполняется для всех объектов, что показано на рисунке 23.

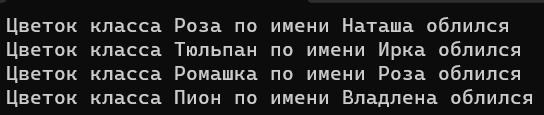


Рисунок 23 − Метод water()

При выборе пункта «3» в данном меню выполнится общий метод stroke(). Метод выполняется для всех объектов, что показано на рисунке 24.

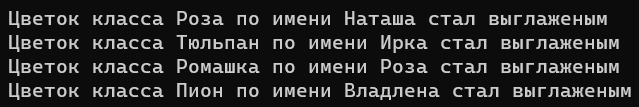


Рисунок 24 − Метод stroke()

При выборе несуществующего пункта меню появится сообщение об ошибке, что показано на рисунке 25.



Рисунок 25 − Выбор несуществующего пункта меню

При выборе пункта «0» происходит выход в главное меню.

При выборе пункта «4» в главном меню можно выполнить собственные методы. Для этого нужно выбрать класс, что показано на рисунке 26.

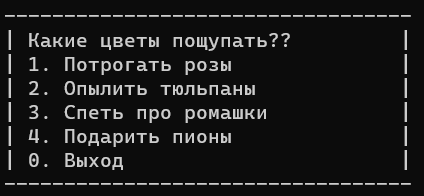


Рисунок 26 − Выбор класса для выполнения собственного метода

При выборе пункта «1» в данном меню выполнится собственный метод для класса rose, что показано на рисунке 27.



Рисунок 27 − Собственный метод класса rose

При выборе пункта «2» в данном меню выполнится собственный метод для класса tulip, что показано на рисунке 28.



Рисунок 28 − Собственный метод класса tulip

При выборе пункта «3» в данном меню выполнится собственный метод для класса chamomile, что показано на рисунке 29.



Рисунок 29 − Собственный метод класса chamomile

При выборе пункта «4» в данном меню выполнится собственный метод для класса tulip, что показано на рисунке 30.



Рисунок 30 − Собственный метод класса peonies

При выборе несуществующего пункта меню появится сообщение об ошибке, что показано на рисунке 31.

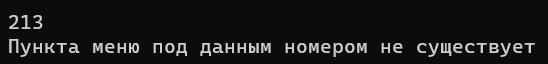


Рисунок 31 − Выбор несуществующего пункта меню

При выборе пункта «0» в данном меню происходит выход в главное меню.

При выборе несуществующего пункта в главном меню появится сообщение об ошибке, что показано на рисунке 32.

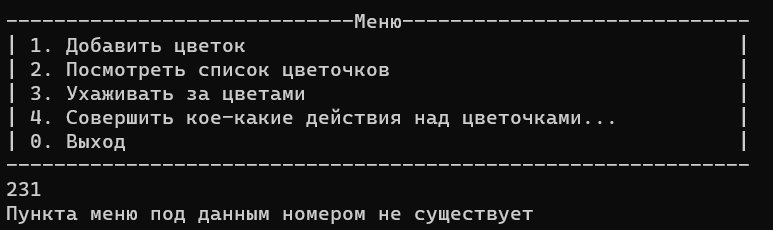


Рисунок 32 − Выбор несуществующего пункта меню

При выборе пункта «0» в главном меню работа программы завершится.

**Задание 3**

Разработать общий интерфейс для доступа к объектам классов разработанной иерархии. Интерфейс должен предоставлять доступ к общим для всех классов иерархии методам. Необходимо добавить новый класс, не входящий в иерархию, однако также реализующий разработанный интерфейс.

Создать список объектов классов, реализующих разработанный интерфейс. Создать функцию, принимающую в качестве параметра объект любого класса, реализующего интерфейс.

Написать программу с меню, позволяющую протестировать работу списка. Обязательные пункты меню:

– добавление нового объекта выбранного пользователем класса в список;

– вывод свойств объекта из списка;

– выполнение методов объекта из списка;

– вывод всех объектов в списке с указанием их классов;

– выполнение созданной функции с указанным объектом из списка.

**Текст программы**

using System;

using System.Collections.Generic;

namespace pr1

{

interface ISmell

{

string smell();

string toString();

string getClass();

}

class perfume : ISmell

{

private string name;

private string flower;

private double price;

private double time;

//свойства

public string Name

{

set { name = value; }

get { return name; }

}

public string Flower

{

set { flower = value; }

get { return flower; }

}

public double Price

{

set { price = value; }

get { return price; }

}

public double Time

{

set { time = value; }

get { return time; }

}

//конструкторы

public perfume() //по умолчанию

{

name = "";

flower = "";

price = 0;

time = 0;

}

public perfume(string Name, string Flower, double Price, double Time) //с параметрами

{

name = Name;

flower = Flower;

price = Price;

time = Time;

}

public perfume(perfume p)

{

name = p.name;

flower = p.flower;

price = p.price;

time = p.time;

}

//методы интерфейса

public string smell()

{

return $"Вы вдохнули аромат одеколона {name} и почувствовали привкус Советского Союза";

}

public string getClass()

{

return "Парфюм";

}

public string toString()

{

return $"Класс: {this.getClass()}\nИмя: {name}\nСделан из этих цветов: {flower}\nЦена: {price}\nДлится часов: {time}";

}

//собственный метод

public void spray()

{

Console.WriteLine($"Теперь Вы пахнете парфюмом {name}");

}

}

abstract class flower : ISmell

{

protected string name;

protected string sort;

protected string color;

protected string shape;

//свойства

public string Name

{

set { name = value; }

get { return name; }

}

public string Sort

{

set { sort = value; }

get { return sort; }

}

public string Color

{

set { color = value; }

get { return color; }

}

public string Shape

{

set { color = value; }

get { return shape; }

}

//конструкторы

public flower() //по умолчанию

{

name = "";

sort = "";

color = "";

shape = "";

}

public flower(string Name, string Sort, string Color, string Shape) //с параметрами

{

name = Name;

sort = Sort;

color = Color;

shape = Shape;

}

public flower(flower f) //копирования

{

name = f.name;

sort = f.sort;

color = f.color;

shape = f.shape;

}

public abstract string getClass();

//методы интерфейса

public string smell()

{

return $"Цветок {this.getClass()} по имени {name} был понюхан";

}

public virtual string toString()

{

return $"Класс: {this.getClass()}\nИмя: {name}\nСорт: {sort}\nЦвет: {color}";

}

//другие методы

public void water()

{

Console.WriteLine($"Цветок {this.getClass()} по имени {name} облился");

}

public void stroke()

{

Console.WriteLine($"Цветок {this.getClass()} по имени {name} стал выглаженым");

}

}

class rose : flower

{

private int height;

//свойство

public int Height

{

set { height = value; }

get { return height; }

}

//конструктор по умолчанию

public rose() : base()

{

height = 0;

}

//конструктор с параметрами

public rose(string Name, string Sort, string Color, string Shape, int H) : base(Name, Sort, Color, Shape)

{

height = H;

}

public rose(rose r) : base(r) //копирования

{

height = r.height;

}

//переопределенные методы интерфейса

public override string getClass()

{

return "Роза";

}

public override string toString()

{

return $"Класс: {this.getClass()}\nИмя: {name}\nСорт: {sort}\nЦвет: {color}\nФорма: {shape}\nВысота: {height}";

}

public new string smell()

{

return $"Цветок {this.getClass()} по имени {name} был понюхан";

}

//собственный метод

public void injection()

{

Console.WriteLine($"Эта роза ({name}) решила приКОЛОТЬся :))");

}

}

class tulip : flower

{

private int bugs;

//свойство

public int Bugs

{

set { bugs = value; }

get { return bugs; }

}

//конструкторы

public tulip() : base() //по умолчанию

{

bugs = 0;

}

public tulip(string Name, string Sort, string Color, string Shape, int Bugs) : base(Name, Sort, Color, Shape) //с параметрами

{

bugs = Bugs;

}

public tulip(tulip t) : base(t) //копирования

{

bugs = t.bugs;

}

//переопределенные методы интерфейса

public override string getClass()

{

return "Тюльпан";

}

public override string toString()

{

return $"Класс: {this.getClass()}\nИмя: {name}\nСорт: {sort}\nЦвет: {color}\nЖучков на цветке: {bugs}";

}

//собственный метод

public void pollinate()

{

Console.WriteLine($"Тюльпан ({name}) помер от аллергии на пыль..");

}

}

class chamomile : flower

{

private int petals;

//свойство

public int Petals

{

set { petals = value; }

get { return petals; }

}

//конструкторы

public chamomile() : base() //по умолчанию

{

petals = 0;

}

public chamomile(string Name, string Sort, string Color, string Shape, int Petals) : base(Name, Sort, Color, Shape) //с параметрами

{

petals = Petals;

}

public chamomile(chamomile c) : base(c)

{

petals = c.petals;

}

//переопределенные методы интерфейса

public override string getClass()

{

return "Ромашка";

}

public override string toString()

{

return $"Класс: {this.getClass()} \nИмя: {name}\nСорт: {sort}\nЦвет: {color}\nФорма: {shape}\nЛепестков: {petals}";

}

//собственный метод

public void sing\_a\_song()

{

Console.WriteLine("-----------------------------------------------------");

Console.WriteLine("| Ромашки |");

Console.WriteLine("-----------------------------------------------------");

Console.WriteLine("| И сотни роз не стоят ни одной твоей ромашки |");

Console.WriteLine("| Ромашки летят в костёр |");

Console.WriteLine("| Все ночи без тебя – они так тяжки |");

Console.WriteLine("| Мне тяжело одной |");

Console.WriteLine("| Нашу любовь не купишь за бумажки |");

Console.WriteLine("| Она дороже всех твоих духов |");

Console.WriteLine("| Ведь сотни роз не стоят и ромашки, ромашки |");

Console.WriteLine("| |");

Console.WriteLine("| Если любить, то сердцем, а не глазами |");

Console.WriteLine("| Губы соврут, но душу ты не обманешь |");

Console.WriteLine("| Не ранишь больно, но стреляешь словами, словами |");

Console.WriteLine("| О, да |");

Console.WriteLine("| Это про меня |");

Console.WriteLine("| Ты не променял мой, мой океан |");

Console.WriteLine("| И о, да |");

Console.WriteLine("| Ты не променял |");

Console.WriteLine("| |");

Console.WriteLine("| И сотни роз не стоят ни одной твоей ромашки |");

Console.WriteLine("| Ромашки летят в костёр |");

Console.WriteLine("| Все ночи без тебя – они так тяжки |");

Console.WriteLine("| Мне тяжело одной |");

Console.WriteLine("| Нашу любовь не купишь за бумажки |");

Console.WriteLine("| Она дороже всех твоих духов |");

Console.WriteLine("| Ведь сотни роз не стоят и ромашки, ромашки |");

Console.WriteLine("| |");

Console.WriteLine("| И сотни роз не стоят ни одной твоей ромашки |");

Console.WriteLine("| Ромашки летят в костёр |");

Console.WriteLine("| Все ночи без тебя – они так тяжки |");

Console.WriteLine("| Мне тяжело одной |");

Console.WriteLine("| Нашу любовь не купишь за бумажки |");

Console.WriteLine("| Она дороже всех твоих духов |");

Console.WriteLine("| Ведь сотни роз не стоят и ромашки, ромашки |");

Console.WriteLine("-----------------------------------------------------");

}

}

sealed class peonies : flower

{

private double price;

//свойство

public double Price

{

set { price = value; }

get { return price; }

}

//конструкторы

public peonies() : base() //по умолчанию

{

price = 0;

}

public peonies(string Name, string Sort, string Color, string Shape, double Price) : base(Name, Sort, Color, Shape) //с параметрами

{

price = Price;

}

public peonies(peonies p) : base(p)

{

price = p.price;

}

//переопределенный метод интерфейса

public override string getClass()

{

return "Пион";

}

public override string toString()

{

return $"Класс: {this.getClass()} \nИмя: {name}\nСорт: {sort}\nЦвет: {color}\nФорма: {shape}\nЦена: {price}";

}

//собственный метод

public void gift()

{

Console.WriteLine($"Этот пион ({name}) попал в хорошие руки!");

}

}

class Program

{

static void SmellObject(ISmell obj)

{

Console.WriteLine(obj.smell());

}

static void Main(string[] args)

{

int menu, MENU, umen, UMEN;

int h = 0;

double p = 0, t = 0;

string n, s, c, sh;

bool integer = true, need = false;

List<ISmell> list = new List<ISmell>();

rose Rose = new rose("Наташа", "Пиано Ред", "Желтый", "Плоская", 45);

list.Add(Rose);

tulip Tulip = new tulip("Ирка", "Принц Карнавал", "Красный", "Лилиецветная", 13);

list.Add(Tulip);

chamomile Cham = new chamomile("Роза", "Персидская", "Розовый", "Крестовидно-розеточная", 17);

list.Add(Cham);

peonies Peon = new peonies("Владлена", "Дюшес де Немур", "Розовый", "Помпонная", 500);

list.Add(Peon);

perfume Perf = new perfume("'Павел'", "Крапива", 150, 8);

list.Add(Perf);

do

{

Console.WriteLine("-----------------------------Меню-----------------------------");

Console.WriteLine("| 1. Добавить объект в список |");

Console.WriteLine("| 2. Посмотреть список объектов |");

Console.WriteLine("| 3. Выполнение методов объектов |");

Console.WriteLine("| 4. Выполнение функции |");

Console.WriteLine("| 0. Выход |");

Console.WriteLine("--------------------------------------------------------------");

if (int.TryParse(Console.ReadLine(), out menu))

{

switch (menu)

{

case 1:

Console.Clear();

do

{

Console.WriteLine("----------------------------------");

Console.WriteLine("| 1. Роза |");

Console.WriteLine("| 2. Тюльпан |");

Console.WriteLine("| 3. Ромашка |");

Console.WriteLine("| 4. Пион |");

Console.WriteLine("| 5. Духи |");

Console.WriteLine("| 0. Выход |");

Console.WriteLine("----------------------------------");

if (int.TryParse(Console.ReadLine(), out MENU))

{

switch (MENU)

{

case 1:

Console.Clear();

Console.WriteLine("Введите имя для цветка: ");

n = Console.ReadLine();

Console.WriteLine("Введите сорт: ");

s = Console.ReadLine();

Console.WriteLine("Введите цвет: ");

c = Console.ReadLine();

Console.WriteLine("Введите форму: ");

sh = Console.ReadLine();

do

{

Console.WriteLine("Введите высоту в сантиметрах: ");

if (int.TryParse(Console.ReadLine(), out int integ) && integ <= 80 && integ >= 30)

{

h = integ;

integer = true;

}

else

{

integer = false;

Console.WriteLine("Введите целое значение от 30 до 80 в сантиметрах");

}

} while (integer == false);

list.Add(new rose(n, s, c, sh, h));

break;

case 2:

Console.Clear();

Console.WriteLine("Введите имя для цветка: ");

n = Console.ReadLine();

Console.WriteLine("Введите сорт: ");

s = Console.ReadLine();

Console.WriteLine("Введите цвет: ");

c = Console.ReadLine();

Console.WriteLine("Введите форму: ");

sh = Console.ReadLine();

do

{

Console.WriteLine("Введите количество жучков на цветке: ");

if (int.TryParse(Console.ReadLine(), out int integ) && integ <= 30 && integ >= 0)

{

h = integ;

integer = true;

}

else

{

integer = false;

Console.WriteLine("Введите целое значение от 0 до 30");

}

} while (integer == false);

list.Add(new tulip(n, s, c, sh, h));

break;

case 3:

Console.Clear();

Console.WriteLine("Введите имя для цветка: ");

n = Console.ReadLine();

Console.WriteLine("Введите сорт: ");

s = Console.ReadLine();

Console.WriteLine("Введите цвет: ");

c = Console.ReadLine();

Console.WriteLine("Введите форму: ");

sh = Console.ReadLine();

do

{

Console.WriteLine("Введите количество лепестков: ");

if (int.TryParse(Console.ReadLine(), out int integ) && integ <= 20 && integ >= 5)

{

h = integ;

integer = true;

}

else

{

integer = false;

Console.WriteLine("Введите целое значение от 5 до 20");

}

} while (integer == false);

list.Add(new chamomile(n, s, c, sh, h));

break;

case 4:

Console.Clear();

Console.WriteLine("Введите имя для цветка: ");

n = Console.ReadLine();

Console.WriteLine("Введите сорт: ");

s = Console.ReadLine();

Console.WriteLine("Введите цвет: ");

c = Console.ReadLine();

Console.WriteLine("Введите форму: ");

sh = Console.ReadLine();

do

{

Console.WriteLine("Введите цену: ");

if (double.TryParse(Console.ReadLine(), out double doub) && doub <= 700 && doub >= 100)

{

p = doub;

integer = true;

}

else

{

integer = false;

Console.WriteLine("Введите значение от 100 до 700");

}

} while (integer == false);

list.Add(new peonies(n, s, c, sh, p));

break;

case 5:

Console.Clear();

Console.WriteLine("Введите название для парфюма: ");

n = Console.ReadLine();

Console.WriteLine("Введите цветок: ");

s = Console.ReadLine();

do

{

Console.WriteLine("Введите цену: ");

if (double.TryParse(Console.ReadLine(), out double doub) && doub <= 1500 && doub >= 100)

{

p = doub;

integer = true;

}

else

{

integer = false;

Console.WriteLine("Введите значение от 100 до 1500");

}

} while (integer == false);

do

{

Console.WriteLine("Введите время: ");

if (double.TryParse(Console.ReadLine(), out double doub) && doub <= 12 && doub >= 1)

{

p = doub;

integer = true;

}

else

{

integer = false;

Console.WriteLine("Введите значение от 1 до 12");

}

} while (integer == false);

list.Add(new perfume(n, s, p, t));

break;

case 0:

Console.Clear();

break;

default:

Console.WriteLine("Пункта меню под данным номером не существует");

break;

}

}

} while (MENU != 0);

break;

case 2:

Console.Clear();

Console.WriteLine("--------------------------------");

for (int i = 0; i < list.Count; i++)

{

Console.WriteLine(i + 1 + ". " + list[i].toString());

Console.WriteLine("--------------------------------");

}

break;

case 3:

Console.Clear();

do

{

Console.WriteLine("----------------------------------");

Console.WriteLine("| Выберите действие |");

Console.WriteLine("| 1. Потрогать розы |");

Console.WriteLine("| 2. Опылить тюльпаны |");

Console.WriteLine("| 3. Спеть про ромашки |");

Console.WriteLine("| 4. Подарить пионы |");

Console.WriteLine("| 5. Пшикнуть парфюм |");

Console.WriteLine("| 0. Выход |");

Console.WriteLine("----------------------------------");

if (int.TryParse(Console.ReadLine(), out UMEN))

{

switch (UMEN)

{

case 1:

for (int i = 0; i < list.Count; i++)

{

if (list[i] is rose)

{

Console.WriteLine(i + 1 + ". " + list[i].toString() + "\nКласс: " + list[i].getClass());

Console.WriteLine("-------------------------------------");

}

}

Console.WriteLine("Введите номер с которым хотите взаимодействовать: ");

do

{

if (int.TryParse(Console.ReadLine(), out int sc) && sc >= 1 && sc <= list.Count + 1)

{

int j = sc;

if (list[j - 1] is rose)

{

rose ROSE = (rose)list[j - 1];

ROSE.injection();

need = true;

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

} while (need == false);

break;

case 2:

for (int i = 0; i < list.Count; i++)

{

if (list[i] is tulip)

{

Console.WriteLine(i + 1 + ". " + list[i].toString() + "\nКласс: " + list[i].getClass());

Console.WriteLine("-------------------------------------");

}

}

Console.WriteLine("Введите номер с которым хотите взаимодействовать: ");

do

{

if (int.TryParse(Console.ReadLine(), out int sc) && sc >= 1 && sc <= list.Count + 1)

{

int j = sc;

if (list[j - 1] is tulip)

{

tulip TULIP = (tulip)list[j - 1];

TULIP.pollinate();

need = true;

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

} while (need == false);

break;

case 3:

for (int i = 0; i < list.Count; i++)

{

if (list[i] is chamomile)

{

Console.WriteLine(i + 1 + ". " + list[i].toString() + "\nКласс: " + list[i].getClass());

Console.WriteLine("-------------------------------------");

}

}

Console.WriteLine("Введите номер с которым хотите взаимодействовать: ");

do

{

if (int.TryParse(Console.ReadLine(), out int sc) && sc >= 1 && sc <= list.Count + 1)

{

int j = sc;

if (list[j - 1] is chamomile)

{

chamomile CHAM = (chamomile)list[j - 1];

CHAM.sing\_a\_song();

need = true;

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

} while (need == false);

break;

case 4:

for (int i = 0; i < list.Count; i++)

{

if (list[i] is peonies)

{

Console.WriteLine(i + 1 + ". " + list[i].toString() + "\nКласс: " + list[i].getClass());

Console.WriteLine("-------------------------------------");

}

}

Console.WriteLine("Введите номер с которым хотите взаимодействовать: ");

do

{

if (int.TryParse(Console.ReadLine(), out int sc) && sc >= 1 && sc <= list.Count + 1)

{

int j = sc;

if (list[j - 1] is peonies)

{

peonies CHAM = (peonies)list[j - 1];

CHAM.gift();

need = true;

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

} while (need == false);

break;

case 5:

for (int i = 0; i < list.Count; i++)

{

if (list[i] is perfume)

{

Console.WriteLine(i + 1 + ". " + list[i].toString() + "\nКласс: " + list[i].getClass());

Console.WriteLine("-------------------------------------");

}

}

Console.WriteLine("Введите номер с которым хотите взаимодействовать: ");

do

{

if (int.TryParse(Console.ReadLine(), out int sc) && sc >= 1 && sc <= list.Count + 1)

{

int j = sc;

if (list[j - 1] is perfume)

{

perfume PERF = (perfume)list[j - 1];

PERF.spray();

need = true;

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

} while (need == false);

break;

case 0:

Console.Clear();

break;

default:

Console.WriteLine("Пункта меню под данным номером не существует");

break;

}

}

} while (UMEN != 0);

break;

case 4:

Console.Clear();

do

{

Console.WriteLine("----------------------------------");

Console.WriteLine("| Выберите чего понюхать |");

Console.WriteLine("| 1. Розы |");

Console.WriteLine("| 2. Тюльпаны |");

Console.WriteLine("| 3. Ромашки |");

Console.WriteLine("| 4. Пионы |");

Console.WriteLine("| 5. Парфюм |");

Console.WriteLine("| 0. Выход |");

Console.WriteLine("----------------------------------");

if (int.TryParse(Console.ReadLine(), out umen))

{

switch (umen)

{

case 1:

for (int i = 0; i < list.Count; i++)

{

if (list[i] is rose)

{

Console.WriteLine(i + 1 + ". " + list[i].toString() + "\nКласс: " + list[i].getClass());

Console.WriteLine("-------------------------------------");

}

}

Console.WriteLine("Введите номер с которым хотите взаимодействовать: ");

do

{

if (int.TryParse(Console.ReadLine(), out int sc) && sc >= 1 && sc <= list.Count + 1)

{

int j = sc;

if (list[j - 1] is rose)

{

SmellObject(list[j - 1]);

need = true;

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

} while (need == false);

break;

case 2:

for (int i = 0; i < list.Count; i++)

{

if (list[i] is tulip)

{

Console.WriteLine(i + 1 + ". " + list[i].toString() + "\nКласс: " + list[i].getClass());

Console.WriteLine("-------------------------------------");

}

}

Console.WriteLine("Введите номер с которым хотите взаимодействовать: ");

do

{

if (int.TryParse(Console.ReadLine(), out int sc) && sc >= 1 && sc <= list.Count + 1)

{

int j = sc;

if (list[j - 1] is tulip)

{

SmellObject(list[j - 1]);

need = true;

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

} while (need == false);

break;

case 3:

for (int i = 0; i < list.Count; i++)

{

if (list[i] is chamomile)

{

Console.WriteLine(i + 1 + ". " + list[i].toString() + "\nКласс: " + list[i].getClass());

Console.WriteLine("-------------------------------------");

}

}

Console.WriteLine("Введите номер с которым хотите взаимодействовать: ");

do

{

if (int.TryParse(Console.ReadLine(), out int sc) && sc >= 1 && sc <= list.Count + 1)

{

int j = sc;

if (list[j - 1] is chamomile)

{

SmellObject(list[j - 1]);

need = true;

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

} while (need == false);

break;

case 4:

for (int i = 0; i < list.Count; i++)

{

if (list[i] is peonies)

{

Console.WriteLine(i + 1 + ". " + list[i].toString() + "\nКласс: " + list[i].getClass());

Console.WriteLine("-------------------------------------");

}

}

Console.WriteLine("Введите номер с которым хотите взаимодействовать: ");

do

{

if (int.TryParse(Console.ReadLine(), out int sc) && sc >= 1 && sc <= list.Count + 1)

{

int j = sc;

if (list[j - 1] is peonies)

{

SmellObject(list[j - 1]);

need = true;

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

} while (need == false);

break;

case 5:

for (int i = 0; i < list.Count; i++)

{

if (list[i] is perfume)

{

Console.WriteLine(i + 1 + ". " + list[i].toString() + "\nКласс: " + list[i].getClass());

Console.WriteLine("-------------------------------------");

}

}

Console.WriteLine("Введите номер с которым хотите взаимодействовать: ");

do

{

if (int.TryParse(Console.ReadLine(), out int sc) && sc >= 1 && sc <= list.Count + 1)

{

int j = sc;

if (list[j - 1] is perfume)

{

SmellObject(list[j-1]);

need = true;

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

}

else

{

Console.WriteLine("Такого варианта нет");

need = false;

}

} while (need == false);

break;

case 0:

Console.Clear();

break;

default:

Console.WriteLine("Пункта меню под данным номером не существует");

break;

}

}

} while (umen != 0);

break;

case 0:

Console.Clear();

break;

default:

Console.WriteLine("Пункта меню под данным номером не существует");

break;

}

}

} while (menu != 0);

}

}

}

**Результат работы программы**

При запуске программы появляется главное меню, что показано на рисунке 33.

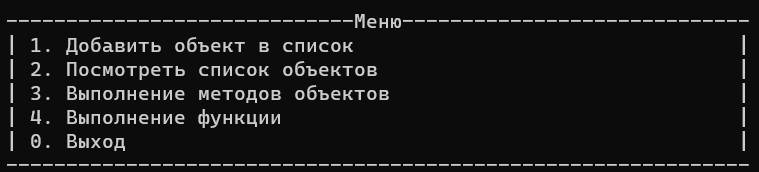


Рисунок 33 − Главное меню

При выборе пункта «1» в главном меню можно добавить объект в список. Для этого нужно выбрать класс, что показано на рисунке 34.

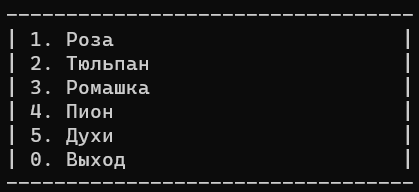


Рисунок 34 − Выбор класса

При выборе пункта «1» в данном меню можно добавить объект класса rose. Нужно ввести имя, сорт, цвет, форму и высоту в сантиметрах. При вводе высоты меньше 30 см и больше 80 см появляется ошибка, что показано на рисунке 35.

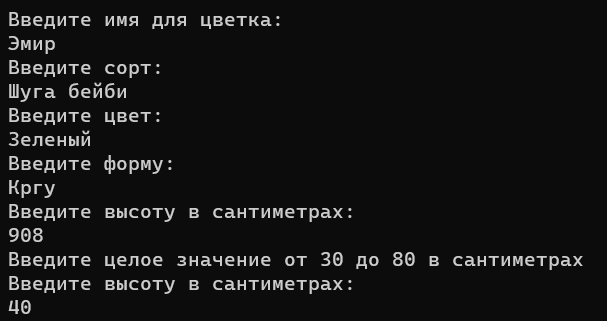


Рисунок 35 − Добавление объекта в класс rose

При выборе пункта «2» в данном меню можно добавить объект класса tulip. Нужно ввести имя, сорт, цвет, форму и количество жучков. При вводе количества жучков меньше 0 и больше 30 появляется ошибка, что показано на рисунке 36.

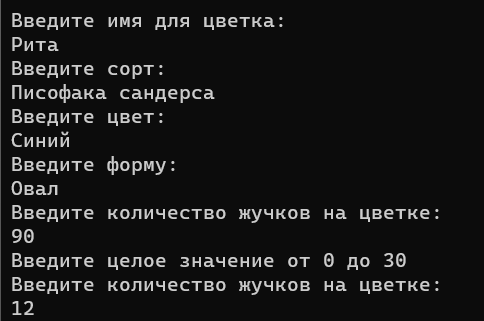


Рисунок 36 − Добавление объекта в класс tulip

При выборе пункта «3» в данном меню можно добавить объект класса chamomile. Нужно ввести имя, сорт, цвет, форму и количество лепестков. При вводе количества жучков меньше 5 и больше 20 появляется ошибка, что показано на рисунке 37.

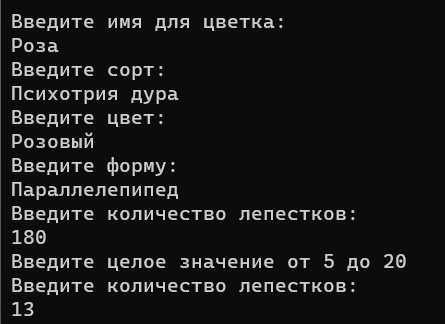


Рисунок 37 − Добавление объекта класса chamomile

При выборе пункта «4» в данном меню можно добавить объект класса peonies. Нужно ввести имя, сорт, цвет и цену. При вводе цены меньше 100 и больше 700 появляется ошибка, что показано на рисунке 38.

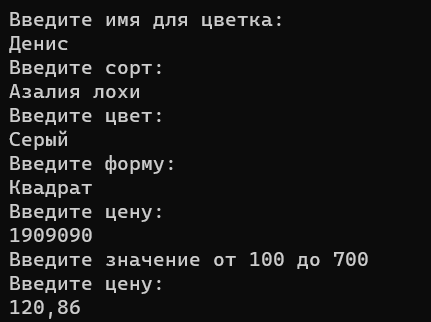


Рисунок 38 − Добавление объекта класса peonies

При выборе пункта «5» в данном меню можно добавить объект класса perfume. Нужно ввести имя, запах цветка, цену и время действия. При вводе цены меньше 100 и больше 700, а также при вводе времени меньше 1 и больше 12 появляется ошибка, что показано на рисунке 39.

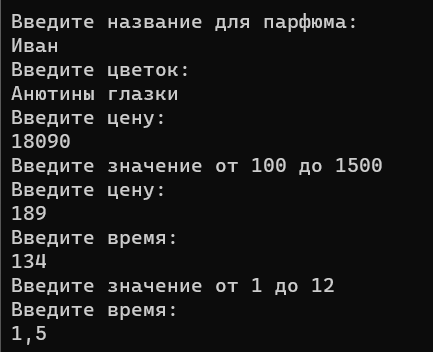


Рисунок 39 − Добавление объекта класса perfume

При выборе несуществующего пункта меню появится сообщение об ошибке, что показано на рисунке 40.



Рисунок 40 − Выбор несуществующего пункта меню

При выборе пункта «0» в данном меню произойдет выход в главное меню.

При выборе пункта «2» в главном меню можно просмотреть весь список объектов с указанием их свойств, что показано на рисунке 41.

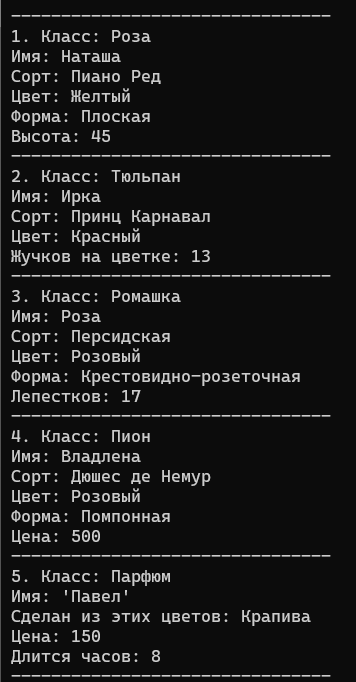


Рисунок 41 − Просмотр списка всех объектов

При выборе пункта «3» в главном меню можно выполнить собственные методы объектов. Для этого нужно выбрать класс, что показано на рисунке 42.

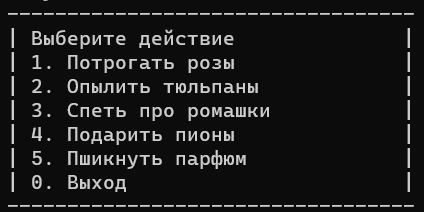


Рисунок 42 − Выбор класса для выполнения собственного метода

При выборе пункта «1» в данном меню нужно выбрать номер объекта, а затем выполнится собственный метод для класса rose. что показано на рисунке 43.

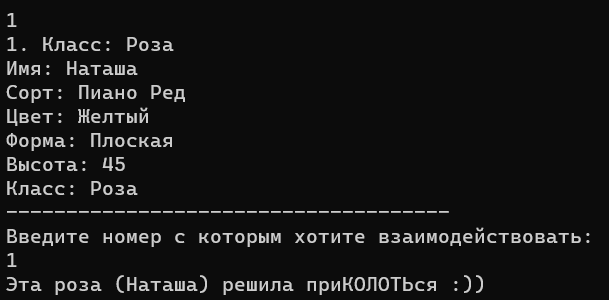


Рисунок 43 − Собственный метод класса rose

При выборе пункта «2» в данном меню нужно выбрать номер объекта, а затем выполнится собственный метод для класса tulip, что показано на рисунке 44.

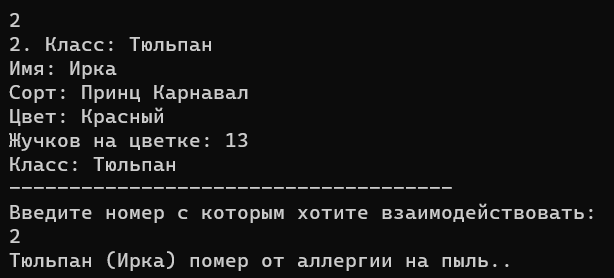


Рисунок 44 − Собственный метод класса tulip

При выборе пункта «3» в данном меню нужно выбрать номер объекта, а затем выполнится собственный метод для класса chamomile, что показано на рисунке 45.



Рисунок 45 − Собственный метод класса chamomile

При выборе пункта «4» в данном меню нужно выбрать номер объекта, а затем выполнится собственный метод для класса tulip, что показано на рисунке 46.

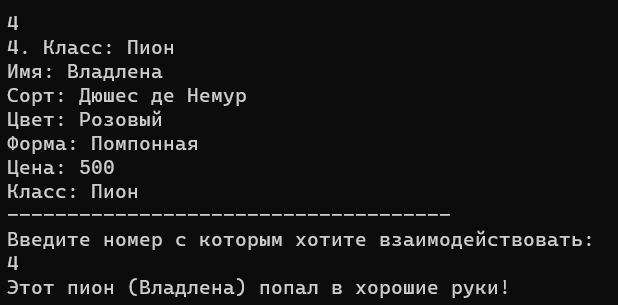


Рисунок 46 − Собственный метод класса peonies

При выборе пункта «5» в данном меню нужно выбрать номер объекта, а затем выполнится собственный метод для класса perfume, что показано на рисунке 47.

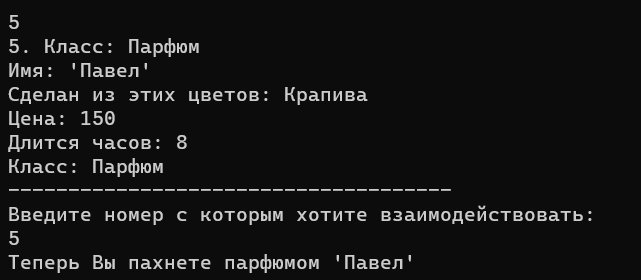


Рисунок 47 − Собственный метод класса perfume

При выборе несуществующего пункта меню появится сообщение об ошибке, что показано на рисунке 48.

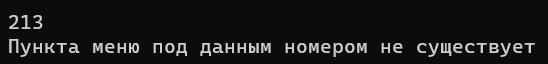


Рисунок 48 − Выбор несуществующего пункта меню

При выборе пункта «0» в данном меню происходит выход в главное меню.

При выборе пункта «4» в главном меню выполнится созданная в интерфейсе функция. Нужно выбрать класс, что показано на рисунке 49.

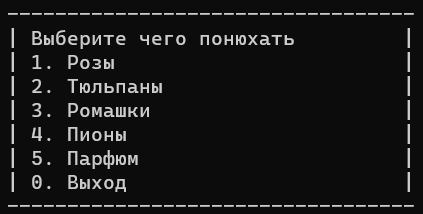


Рисунок 49 − Выбор класса для выполнения функции

При выборе пункта «1» в данном меню выполнится функция для объекта класса rose, что показано на рисунке 50.

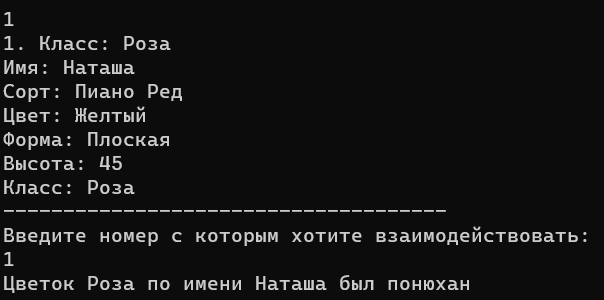


Рисунок 50 − Выполнение функции для объекта класса rose

При выборе пункта «2» в данном меню нужно выбрать номер объекта, а затем выполнится функция для объекта класса tulip, что показано на рисунке 51.

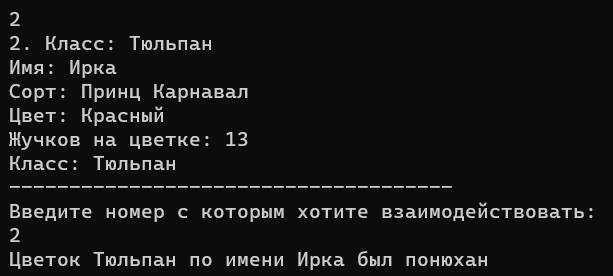


Рисунок 51 − Выполнение функции для объекта класса tulip

При выборе пункта «3» в данном меню нужно выбрать номер объекта, а затем выполнится функция для объекта класса chamomile, что показано на рисунке 52.

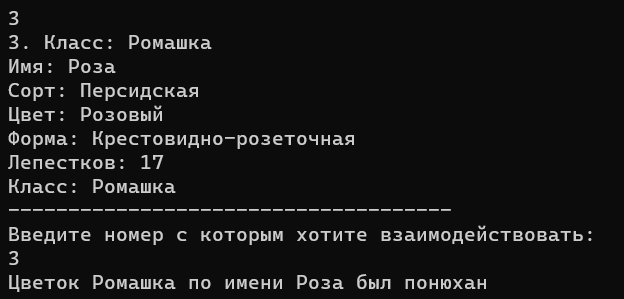


Рисунок 52 − Выполнение функции для объекта класса chamomile

При выборе пункта «4» в данном меню нужно выбрать номер объекта, а затем выполнится функция для объекта класса peonies, что показано на рисунке 53.

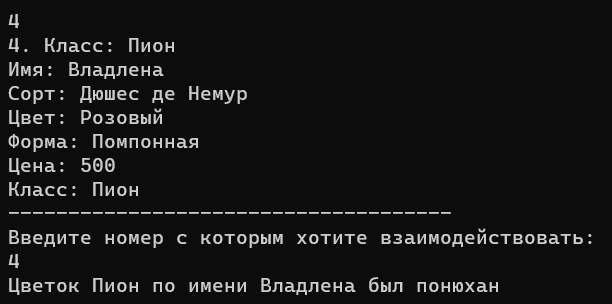


Рисунок 53 − Выполнение функции для объекта класса peonies

При выборе пункта «5» в данном меню нужно выбрать номер объекта, а затем выполнится функция для объекта класса rose, что показано на рисунке 54.

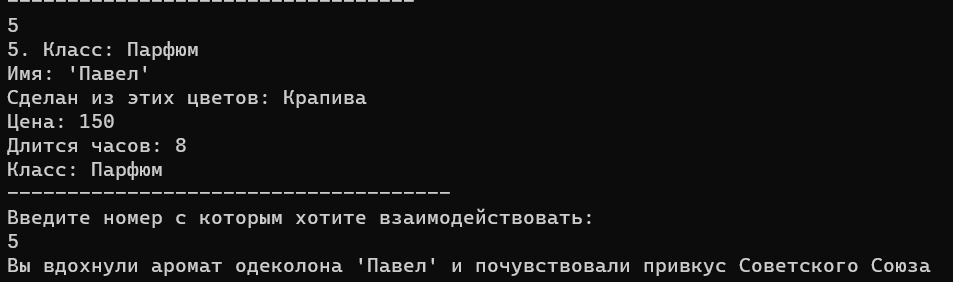


Рисунок 54 − Выполнение функции для объекта класса perfume

При выборе несуществующего пункта меню появится сообщение об ошибке, что показано на рисунке 55.

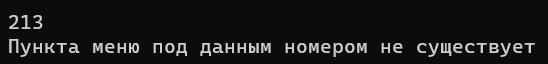


Рисунок 55 − Выбор несуществующего пункта меню

При выборе пункта «0» в данном меню происходит выход в главное меню.

При выборе пункта «0» в главном меню программа завершится.