Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт/Факультет - ИИТиАД

**Название работы -Наследование**

Отчет по лабораторной работе № 4

по дисциплине

Технология программирования

Выполнил cтудент ИСТб-20-2 М.А. Пигарев

номер группы подпись И. О. Фамилия

21.09.21

дата

Принял Доцент З.А. Бахвалова

21.09.21 Должность подпись И. О. Фамилия

дата

Иркутск –2021

Оглавление

[Постановка задачи 3](#_Toc86864589)

[Словесное описание алгоритма 3](#_Toc86864590)

[Таблица спецификаций 4](#_Toc86864591)

[Таблица тестов 5](#_Toc86864592)

[Код программы 6](#_Toc86864593)

[Результаты тестов 11](#_Toc86864594)

[Ссылка на Git 14](#_Toc86864595)

### Постановка задачи

Для раздачи напитков (общее свойство: объем)

* Сок (используемый фрукт, наличие мякоти)
* Газировка (вид, количество пузыриков)
* Алкоголь (крепость, тип)

### Class Drinkables

Поля class Drinkables

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Назначение | Тип | Диапазон |
| rnd | Генератор случайных чисел для формирования случайных полей объектов класса (используется в классах наследниках) | Random | integer |
| Volume | Объём напитка | Enum VolumeType | { пол, один, два } |

Методы class Drinkables

|  |  |  |
| --- | --- | --- |
| Название | Назначение | Тип |
| virtual GetInfo | Возвращает в код строчку, которая содержит в себе информацию об объёме объекта | String |
| virtual GetTYPE | Возвращает тип объекта | String |

Словесное описание метода virtual GetInfo

1. Создаём пустую строки s и str.
2. Смотрим на объём напитка, если он равен “пол” или “два”, то записываем в s “ литра”, если он равен “один”, то записываем в s “ литр”.
3. Строку str заполняем в виде: “Объём: ”+ значение объёма +s.
4. Возвращаем строку str

Таблица спецификаций метода virtual GetInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Назначение | Тип | Диапазон |
| s | Хранение окончания для объёма (литра, литр) | string | - |
| str | Хранение информации об объёме | string | - |

Словесное описание метода virtual GetTYPE

Нужен для того, чтобы вызывать этот метод у наследников. В случае перебора очереди напитков не известно, какой тип напитка сейчас обрабатывается, поэтому необходим общий метод у предка, чтобы вызвать функцию (для каждого наследника эта функция выводит тип напитка).

### Class Juice

Поля class Juice

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Назначение | Тип | Диапазон |
| type | Какого типа сок | Enum JuicesType | { яблоко, виноград, апельсин } |
| pulp | Наличие мякоти у сока | Enum PulpType | { пол, один, два } |

Методы class Juices

|  |  |  |
| --- | --- | --- |
| Название | Назначение | Тип |
| override GetInfo | Возвращает в код строчку, которая содержит в себе информацию об всех полях объекта | String |
| override GetTYPE | Возвращает тип объекта | String |
| Juice(int v, int p,int j) | Конструктор с заданными значениями | - |
| Juice() | Пустой конструктор | - |
| Juice Generate() | Генерация случайного объекта сока | - |

Словесное описание метода override GetInfo.

Записывает в строку всю информацию о полях объекта в удобном формате для чтения пользователя. Два глобальных поля объекта считываются напрямую, а объём считывается посредством метода предка, который записывает информацию об объёме в строку.

Таблица спецификаций метода override GetInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Назначение | Тип | Диапазон |
| str | Хранение информации об объекте | var | - |

Словесное описание метода override GetTYPE.

Записывает в строку тип объекта. В дальнейшем это будет использоваться для вывода картинки на экран (для каждого типа объекта есть своя картинка).

Таблица спецификаций метода override GetTYPE

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Назначение | Тип | Диапазон |
| str | Хранение типа объекта | var | - |

### Class Gas

Поля class Gas

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Назначение | Тип | Диапазон |
| type | Какого типа газировка | Enum GasType | { кола, спрайт, виноград } |
| count | Количество пузырьков | int | >0 |

Методы class Gas

|  |  |  |
| --- | --- | --- |
| Название | Назначение | Тип |
| override GetInfo | Возвращает в код строчку, которая содержит в себе информацию об всех полях объекта | String |
| override GetTYPE | Возвращает тип объекта | String |
| Gas(int v, int p,int j) | Конструктор с заданными значениями | - |
| Gas() | Пустой конструктор | - |
| Gas Generate() | Генерация случайного объекта газировки | - |

Словесное описание метода override GetInfo.

Записывает в строку всю информацию о полях объекта в удобном формате для чтения пользователя. Два глобальных поля объекта считываются напрямую, а объём считывается посредством метода предка, который записывает информацию об объёме в строку.

Таблица спецификаций метода override GetInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Назначение | Тип | Диапазон |
| str | Хранение информации об объекте | var | - |

Словесное описание метода override GetTYPE.

Записывает в строку тип объекта. В дальнейшем это будет использоваться для вывода картинки на экран (для каждого типа объекта есть своя картинка).

Таблица спецификаций метода override GetTYPE

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Назначение | Тип | Диапазон |
| str | Хранение типа объекта | var | - |

### Class Alco

Поля class Alco

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Назначение | Тип | Диапазон |
| type | Какого типа алкоголь | Enum AlcoType | {водка, вино, пиво } |
| Number | Крепость напитка | int | [0.100] |

Методы class Alco

|  |  |  |
| --- | --- | --- |
| Название | Назначение | Тип |
| override GetInfo | Возвращает в код строчку, которая содержит в себе информацию об всех полях объекта | String |
| override GetTYPE | Возвращает тип объекта | String |
| Alco (int v, int p,int j) | Конструктор с заданными значениями | - |
| Alco () | Пустой конструктор | - |
| Alco Generate() | Генерация случайного объекта алкоголя | - |

Словесное описание метода override GetInfo.

Записывает в строку всю информацию о полях объекта в удобном формате для чтения пользователя. Два глобальных поля объекта считываются напрямую, а объём считывается посредством метода предка, который записывает информацию об объёме в строку.

Таблица спецификаций метода override GetInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Назначение | Тип | Диапазон |
| str | Хранение информации об объекте | var | - |

Словесное описание метода override GetTYPE.

Записывает в строку тип объекта. В дальнейшем это будет использоваться для вывода картинки на экран (для каждого типа объекта есть своя картинка).

Таблица спецификаций метода override GetTYPE

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Назначение | Тип | Диапазон |
| str | Хранение типа объекта | var | - |

### Class Form1

Поля class Form1

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Назначение | Тип | Диапазон |
| DrinkablesList | Очередь напитков | List<Drinkables> | - |

Методы class Form1

|  |  |  |
| --- | --- | --- |
| Название | Назначение | Тип |
| button1\_Click | Генерирует случайные напитки и заполняет ими очередь, после чего выводит информацию на экран | void |
| ShowInfo | Выводит на экран количество напитков каждого типа | void |
| button2\_Click | Выводит информацию об напитке, показывает картинку, удаляет напиток из очереди | void |
| ochered | Выводит на экран список очереди напитков | void |

Словесное описание метода override button1\_Click.

Очищает список напитков. В цикле из 10 итераций заполняет случайными напитками очередь, после выводится очередь (с помощью специального метода), а также выводится количество напитков каждого типа (с помощью специального метода).

Таблица спецификаций метода override button1\_Click

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Назначение | Тип | Диапазон |
| rnd | Генератор случайных чисел | var | - |

Словесное описание метода ShowInfo

Перебираем список напитков. Каждый объект будет соответствовать определённому типа напитка (сок, газировка, алкоголь), тогда увеличиваем соответствующий счётчик (изначально они все равны 0). После выводим на экран информацию в удобном для пользователе виде.

Таблица спецификаций метода ShowInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Назначение | Тип | Диапазон |
| JuiceCount | Количество объектов в очереди типа “Сок” | Int | - |
| GasCount | Количество объектов в очереди типа “Газировка” | Int | - |
| AlcoCount | Количество объектов в очереди типа “Алкоголь” | Int | - |

Словесное описание метода button2\_Click

Если количество элементов в списке рано 0, тогда обнуляем picturebox на форме и выводим сообщение о том, что очередь пуста, после заканчиваем работу функции.

Создаём переменную типа Drinkable и присваиваем ей текущий напиток в очереди. Удаляем текущий напиток из очереди. Вызываем метод, который выводит количество напитков каждого типа на экран. Вызываем метод, чтобы узнать тип напитка, на основании этого типа выводим картинку на экран. Вызываем метод, который выводит конкретную информацию об данном напитке, после вызываем метод, который показывает очередь всех напитков.

Словесное описание метода ochered

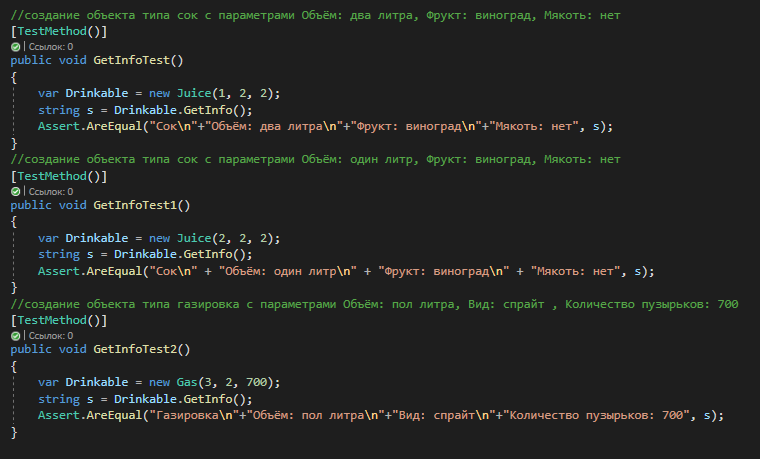
Создаём счётчик для того, чтобы отображать позицию каждого объекта в очереди и пустую строку, для вывода ответа. Перебираем список напитков, для каждого вида напитка приписываем к созданной строке счётчик и вид напитка, после увеличиваем счётчик. Выводим очередь на экран.

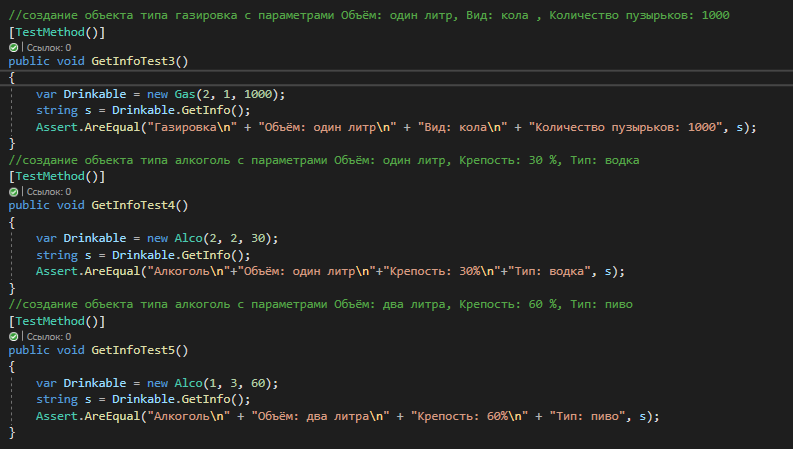
Таблица спецификаций метода ochered

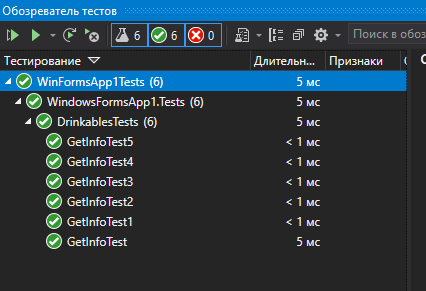
|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Назначение | Тип | Диапазон |
| count | Переменная для показа позиции каждого объекта в очереди | Int | >=1 |
| s | Строка, для хранения очереди напитков | string | - |

### Таблица тестов

| Номер теста | Что проверяем | Входные данные | Выходные данные |
| --- | --- | --- | --- |
| 1 | Работу программы | - | Случайная генерация |
| 2 | Работу программы | - | Случайная генерация |
| 3 | Работу программы | - | Случайная генерация |
| 4 | Работу программы | - | Случайная генерация |
| 5 | Работу программы | - | Случайная генерация |







### Код программы

Код main:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using WindowsFormsApp1;

//using CodeFile1;

namespace WinFormsApp1

{

public partial class Form1 : Form

{

List<Drinkables> DrinkablesList = new List<Drinkables>();

public Form1()

{

InitializeComponent();

ShowInfo();

}

private void button1\_Click(object sender, EventArgs e)

{

this.DrinkablesList.Clear();

// var r = new Random();

var rnd = new Random();

for (var i = 0; i < 10; ++i)

{

switch (rnd.Next() % 3)

{

case 0:

this.DrinkablesList.Add(Juice.Generate());

break;

case 1:

this.DrinkablesList.Add(Gas.Generate());

break;

case 2:

this.DrinkablesList.Add(Alco.Generate());

break;

}

}

ShowInfo();

}

private void ShowInfo()

{

int JuiceCount = 0;

int GasCount = 0;

int AlcoCount = 0;

foreach (var fruit in this.DrinkablesList)

{

if (fruit is Juice)

{

JuiceCount += 1;

}

else if (fruit is Gas)

{

GasCount += 1;

}

else if (fruit is Alco)

{

AlcoCount += 1;

}

}

richTextBox1.Text = "Сок\tГазировка\tАлкоголь";

richTextBox1.Text += "\n";

richTextBox1.Text += String.Format("{0}\t{1}\t\t{2}", JuiceCount, GasCount, AlcoCount);

}

private void button2\_Click(object sender, EventArgs e)

{

if (this.DrinkablesList.Count == 0)

{

txtOut.Text = "Конец очереди. Автомат пуст";

pictureBox1.Image = null;

return;

}

var Drinkable = this.DrinkablesList[0];

this.DrinkablesList.RemoveAt(0);

txtOut.Text = Drinkable.GetInfo();

switch (Drinkable.GetTYPE())

{

case "яблоко":

pictureBox1.Image = Properties.Resources.apple;

break;

case "виноград":

pictureBox1.Image = Properties.Resources.grape;

break;

case "апельсин":

pictureBox1.Image = Properties.Resources.orange;

break;

case "кола":

pictureBox1.Image = Properties.Resources.cola;

break;

case "спрайт":

pictureBox1.Image = Properties.Resources.sprite;

break;

case "фанта":

pictureBox1.Image = Properties.Resources.fanta;

break;

case "водка":

pictureBox1.Image = Properties.Resources.vodka;

break;

case "вино":

pictureBox1.Image = Properties.Resources.wine;

break;

case "пиво":

pictureBox1.Image = Properties.Resources.beer;

break;

}

ShowInfo();

}

}

}

Код класса:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace WindowsFormsApp1

{

public class Drinkables

{

public enum VolumeType { пол, один, два };

public static Random rnd = new Random();

public VolumeType Volume;

public virtual String GetInfo()

{

string str="",s="";

switch (Volume)

{

case VolumeType.пол:

s = " литра";

break;

case VolumeType.один:

s = " литр";

break;

case VolumeType.два:

s = " литра";

break;

}

str += "\nОбъём: " + this.Volume + s;

return str;

}

public virtual String GetTYPE()

{

string str="";

return str;

}

}

//сок

// public enum JuicesType { яблоко, виноград, апельсин };

// public enum PulpType { есть, нет };

// public enum VolumeType { пол, один, два };

public class Juice : Drinkables

{

public enum JuicesType { яблоко, виноград, апельсин };

public enum PulpType { есть, нет };

// public VolumeType Volume = VolumeType.один;

public JuicesType type;

public PulpType Pulp;

// public Juice(VolumeType v,PulpType p, JuicesType j)

public Juice(int v, int p,int j)

{

switch (v)

{

case 1:

Volume = VolumeType.два;

break;

case 2:

Volume = VolumeType.один;

break;

case 3:

Volume = VolumeType.пол;

break;

}

switch (j)

{

case 1:

type=JuicesType.апельсин;

break;

case 2:

type = JuicesType.виноград;

break;

case 3:

type = JuicesType.яблоко;

break;

}

switch (p)

{

case 1:

Pulp = PulpType.есть;

break;

case 2:

Pulp = PulpType.нет;

break;

}

// Volume =v;

// Pulp = p;

// type = j;

}

Juice()

{

}

public override String GetInfo()

{

var str = "Сок";

str += base.GetInfo();

str += String.Format("\nФрукт: {0}",this.type);

str += String.Format("\nМякоть: {0}", this.Pulp);

return str;

}

public static Juice Generate()

{

return new Juice

{

Volume = (VolumeType)rnd.Next(3),

Pulp = (PulpType)rnd.Next(2),

type = (JuicesType)rnd.Next(3),

};

}

public override String GetTYPE()

{

var str = "";

switch (this.type)

{

case JuicesType.яблоко:

str = "яблоко";

break;

case JuicesType.виноград:

str = "виноград";

break;

case JuicesType.апельсин:

str = "апельсин";

break;

}

return str;

}

}

//газировка

// public enum GasType { кола, спрайт,фанта };

public class Gas : Drinkables

{

// public VolumeType Volume = VolumeType.два;

public enum GasType { кола, спрайт, фанта };

public int Count = 0;

public GasType type; // тип

public Gas(int v, int p, int j)

{

switch (v)

{

case 1:

Volume = VolumeType.два;

break;

case 2:

Volume = VolumeType.один;

break;

case 3:

Volume = VolumeType.пол;

break;

}

switch (p)

{

case 1:

type = GasType.кола;

break;

case 2:

type = GasType.спрайт;

break;

case 3:

type = GasType.фанта;

break;

}

Count = j;

// Volume =v;

// Pulp = p;

// type = j;

}

Gas()

{

}

public override String GetInfo()

{

var str = "Газировка";

str += base.GetInfo();

str += String.Format("\nВид: {0}", this.type);

str += String.Format("\nКоличество пузырьков: {0}", this.Count);

return str;

}

public static Gas Generate()

{

return new Gas

{

Volume = (VolumeType)rnd.Next(3),

Count = rnd.Next(500, 1500),

type = (GasType)rnd.Next(3),

};

}

public override String GetTYPE()

{

var str = "";

switch (this.type)

{

case GasType.кола:

str = "кола";

break;

case GasType.спрайт:

str = "спрайт";

break;

case GasType.фанта:

str = "фанта";

break;

}

return str;

}

}

//алкоголь

// public enum AlcoType { водка, вино, пиво };

public class Alco : Drinkables

{

// public VolumeType Volume = VolumeType.пол;

//public AlcoType type = AlcoType.вино;

public AlcoType type;

public int Number = 0;

public enum AlcoType { водка, вино, пиво };

public Alco(int v, int p, int j)

{

switch (v)

{

case 1:

Volume = VolumeType.два;

break;

case 2:

Volume = VolumeType.один;

break;

case 3:

Volume = VolumeType.пол;

break;

}

switch (p)

{

case 1:

type = AlcoType.вино;

break;

case 2:

type = AlcoType.водка;

break;

case 3:

type = AlcoType.пиво;

break;

}

Number = j;

// Volume =v;

// Pulp = p;

// type = j;

}

Alco()

{

}

public override String GetInfo()

{

string str = "Алкоголь";

str += base.GetInfo();

str += "\nКрепость: "+this.Number+"%";

str += String.Format("\nТип: {0}", this.type);

return str;

}

public static Alco Generate()

{

return new Alco

{

Volume = (VolumeType)rnd.Next(3),

Number = rnd.Next(101),

type = (AlcoType)rnd.Next(3),

};

}

public override String GetTYPE()

{

var str = "";

switch (this.type)

{

case AlcoType.водка:

str = "водка";

break;

case AlcoType.вино:

str = "вино";

break;

case AlcoType.пиво:

str = "пиво";

break;

}

return str;

}

}

}

## Результаты тестов

|  |  |
| --- | --- |
| №Теста | Скриншот результата |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |

## Ссылка на Git

https://github.com/Bert-10/4\_laba