Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Институт информационных технологий и анализа данных

наименование института

ОТЧЕТ  
к лабораторной работе по дисциплине

Технология программирования

Наследование

Наименование темы

Выполнил студент группы ИСТб-19–2 Комогорцева Ю.В.

шифр Фамилия И.О.

Проверила Бахвалова З.А.

Фамилия И.О.

Содержание отчета на 34 стр.

Иркутск 2020 г.

### Постановка задачи

**Задача №6**

1. Должен быть 1 базовый класс, и 3 класса наследника. У базового класса должно быть, как минимум одно свойство передающиеся по наследству остальным, у каждого из классов наследников должно быть как минимум два уникальных свойства.
2. Подготовить мини отчет со UML-схемой иерархии классов, с подробным описанием свойств и методов.
3. Реализовать эмулятор торгового автомата по образу и подобию как описано [в](http://olive.tealeaf.su/inheritance.html) методичке
4. Если вы обычно делаете желтые задачки, то в эмуляторе надо реализовать отображение состояния очереди (чтобы было видно какие объекты в каком порядке сейчас находятся в автомате)

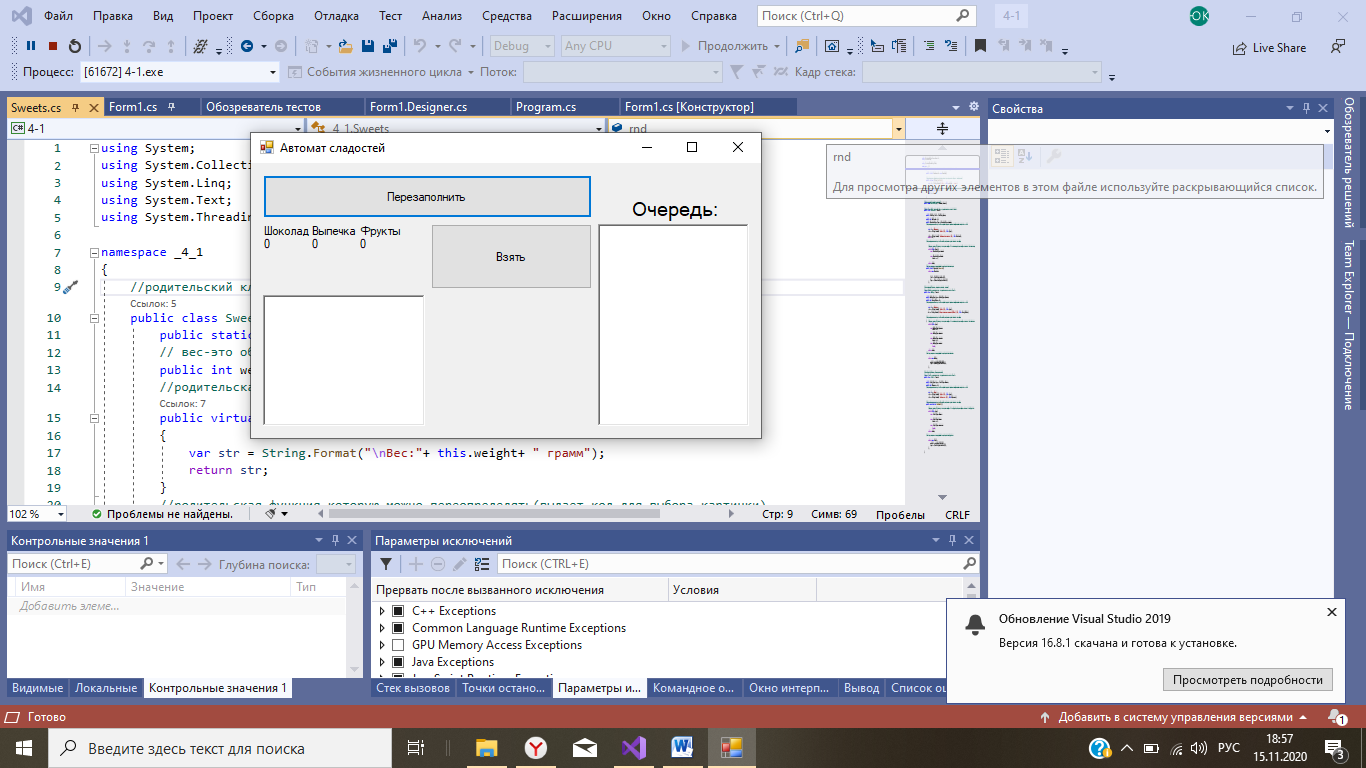
**Автомат для раздачи сладостей (вес)**

* Шоколадки (начинка, количество плиток, тип шоколада (темный, молочный))
* Выпечка (тип (булочка, пирожок, ватрушка…), энергетическая ценность (ккал))
* Фрукты (вид фрукта, спелость)

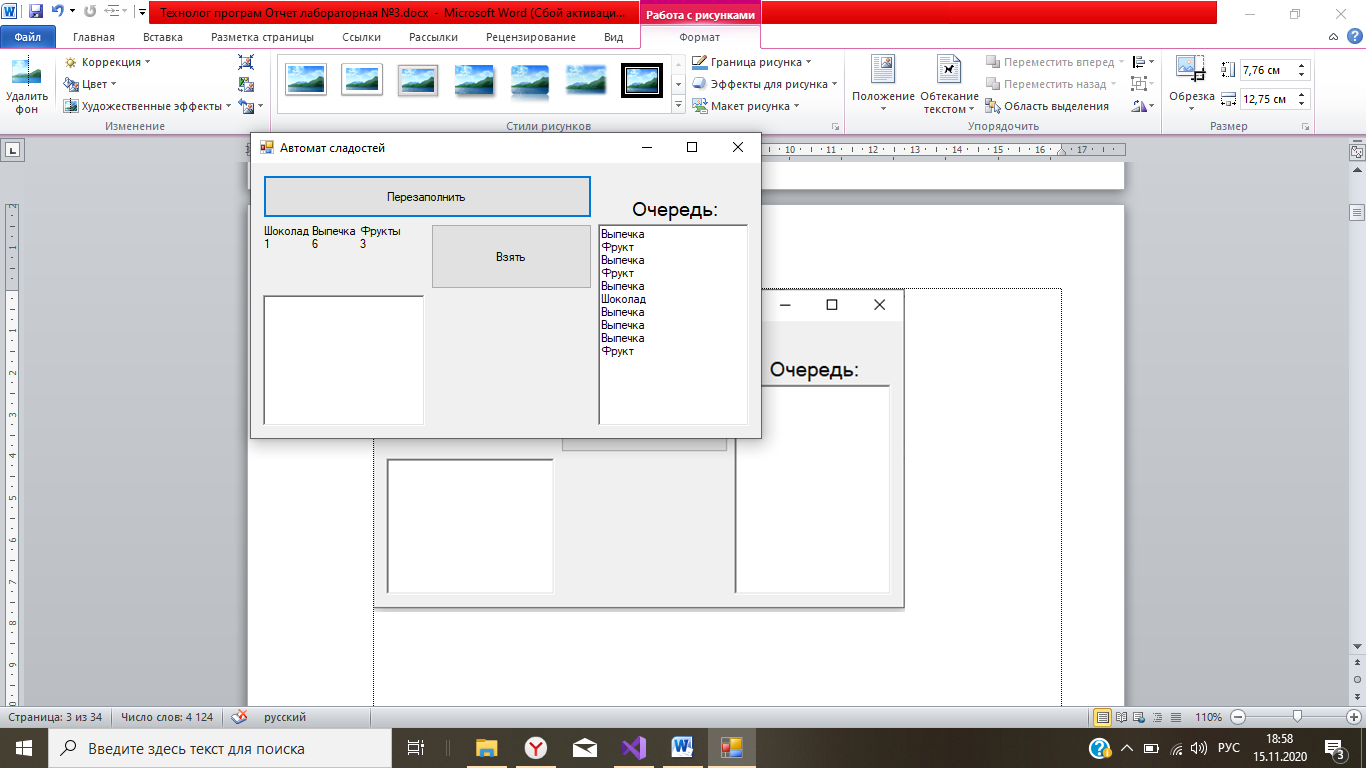
**В программе реализовано:**

* Sweets - 1 базовый класс, и Chocolate, Baking, Fruits - 3 класса наследника.
* У базового класса свойство передающиеся по наследству остальным это вес(weight)
* У каждого из классов наследников имеется от 2х до 3х уникальных свойств.
* Отображение состояния очереди (видно какие объекты в каком порядке сейчас находятся в автомате)
* Отображение в картинке предмета

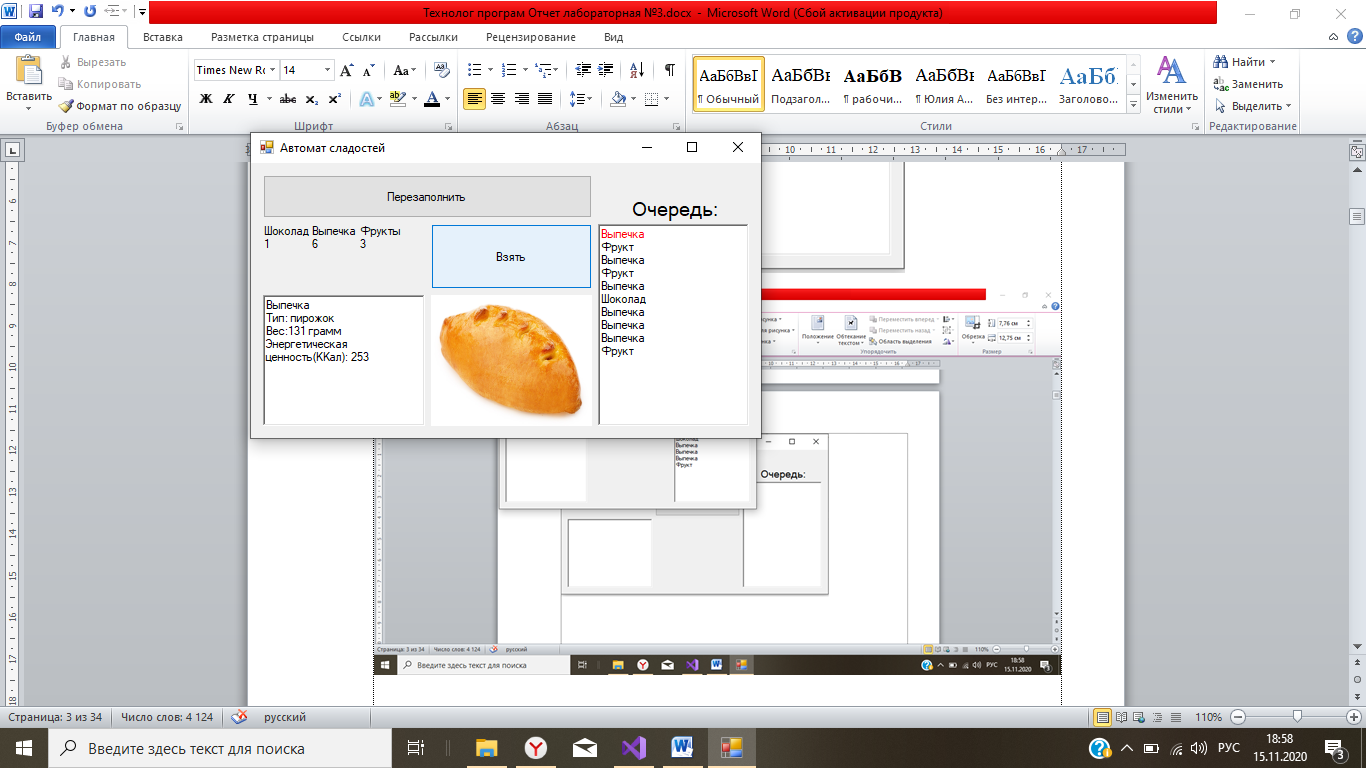
Ниже представлены скриншоты возможностей программы



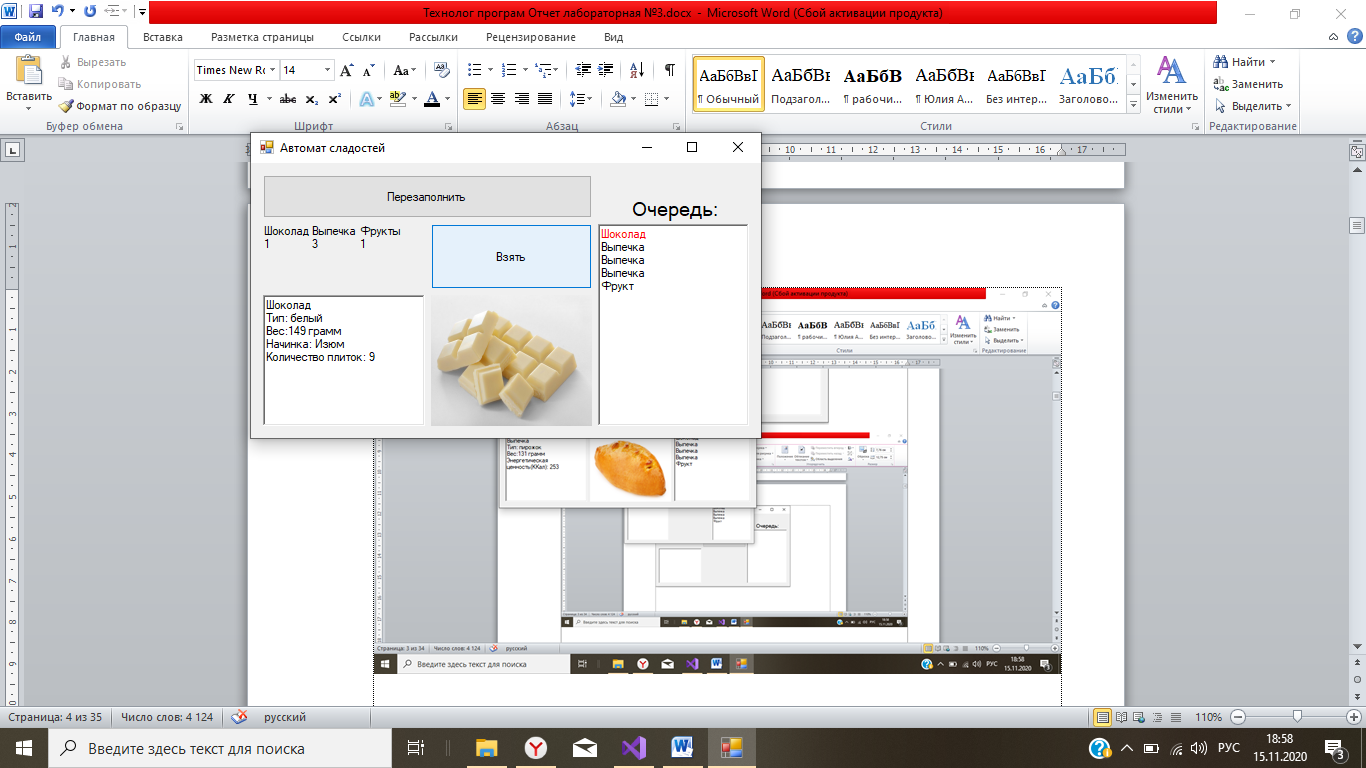
Начальное состояние



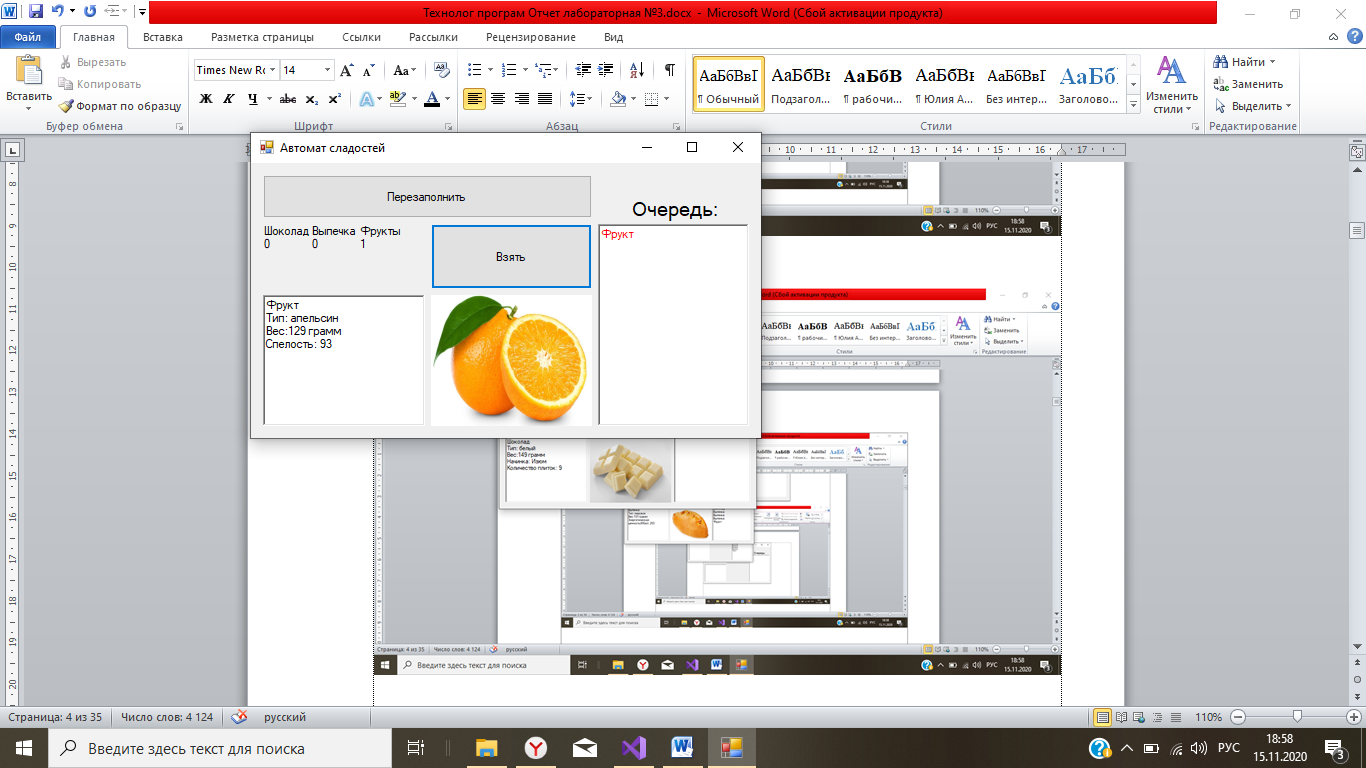
После перезаполнения автомата



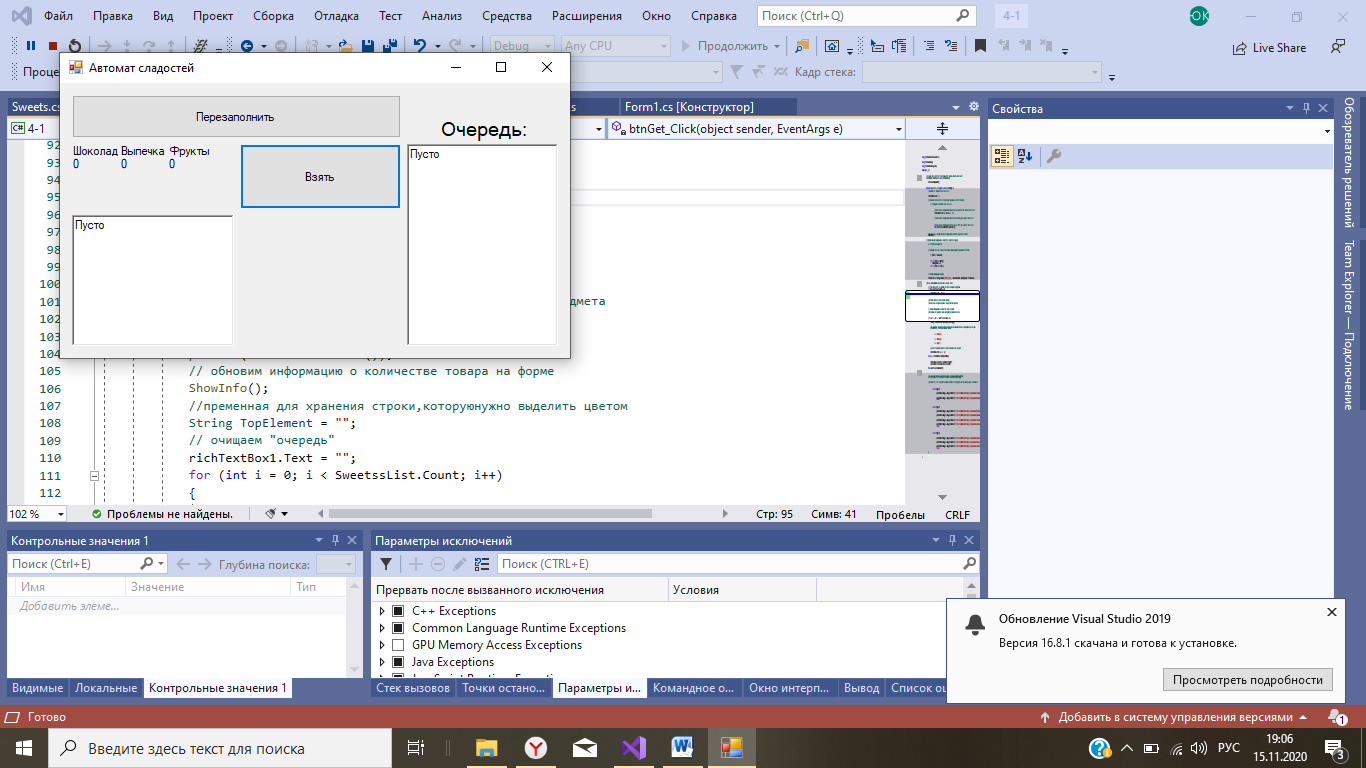
Первый раз нажали кнопку «взять»



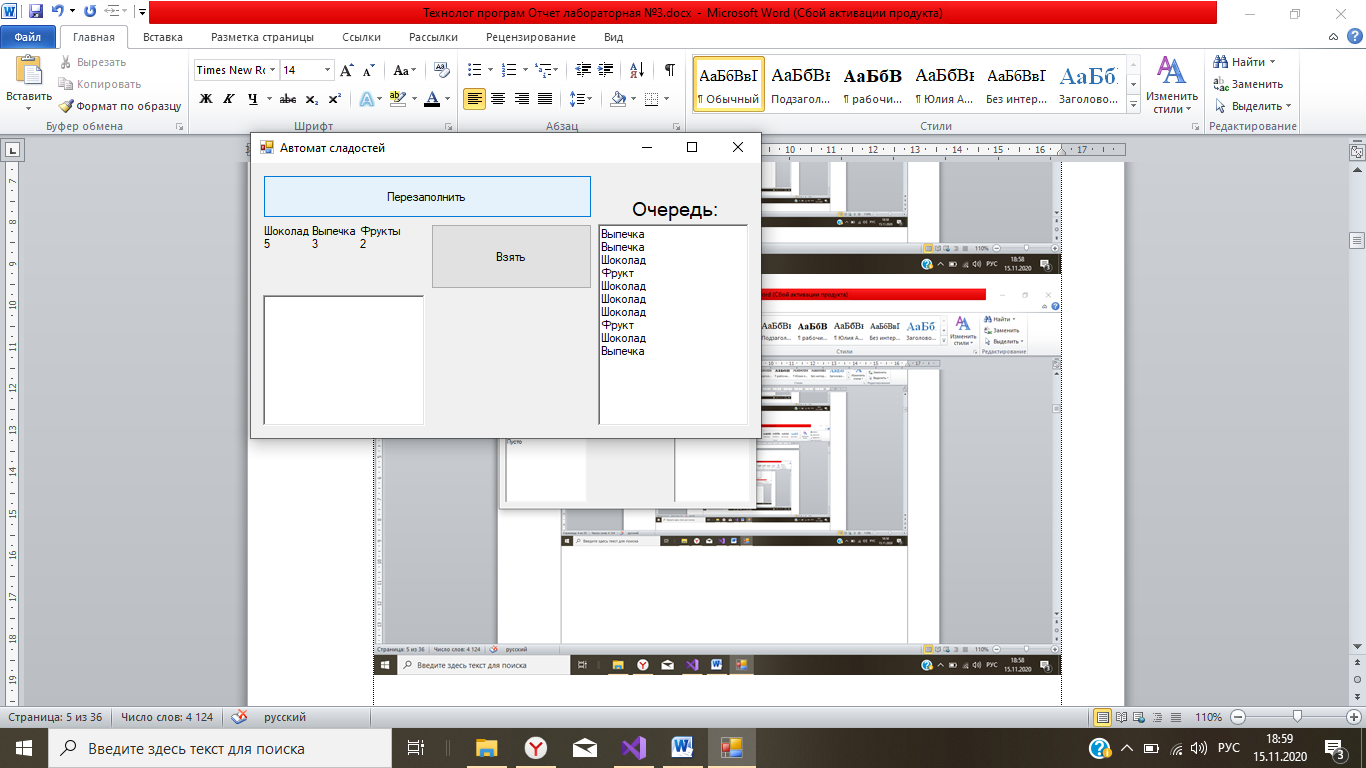
Нажали кнопку «взять» несколько раз



Нажали кнопку «взять» еще несколько раз

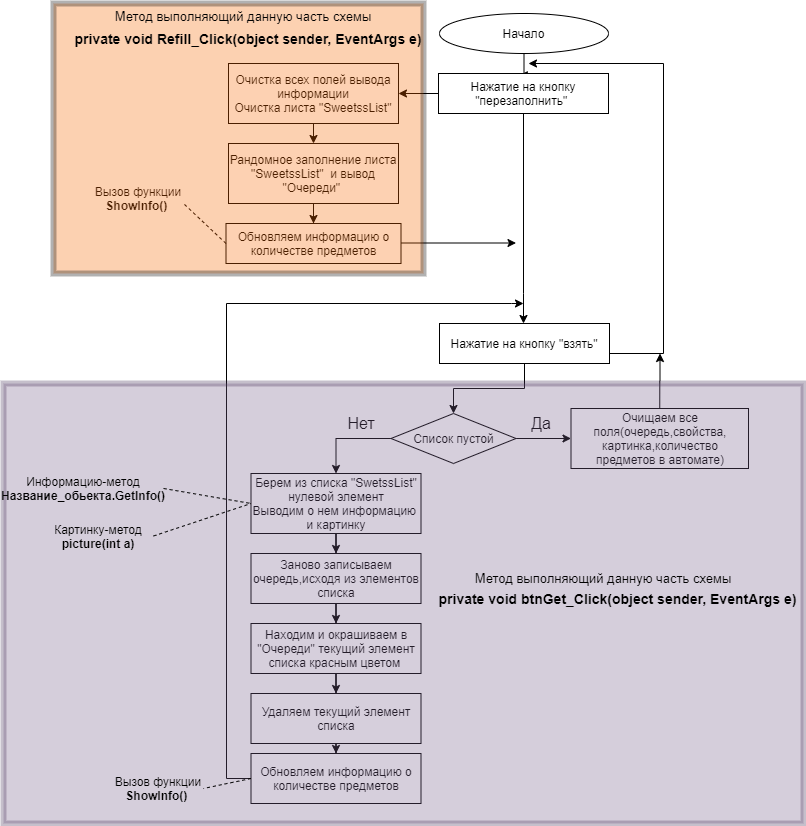


После того,как взяли последний предмет - автомат пуст,очередь пуста



Заново нажимаем перезаполнить-автомат заполняется

**Схематическое описание работы программы**



### Таблицы описания классов,методов,конструкторов

### Тип enum

**public enum ChocolateType { темный, белый };** тип шоколада(темный ,молочный)

**public enum Stuffing {Орехи, Изюм };** наличие начинки(Орехи,изюм)

**public enum BakingType {булочка, пирожок,чизкейк, пончик };** тип выпечки(булочка, пирожок,чизкейк, пончик )

**public enum FruitsType {яблоко, банан,апельсин };** тип фрукта(яблоко, банан,апельсин)

### Класс Sweets-Родитель

**weight = 0**-Локальная переменная, обозначающая вес

public static Random rnd = new Random(); -что бы «дети» этого класса могли «рандомить»

***public virtual String GetInfo()-***родительская функция, которую можно переопределять(выдает информацию об объекте класса)

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| Выходные величины | | |
| str | Строка с информацией об объекте класса | string |

***public virtual int GetNumPic()****–*родительская функция, которую можно переопределять(выдает код для выбора картинки)

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| *Выходные величины* | | |
| index | Код для выбора картинки | int |

### Класс Chocolate-потомок класса Sweets

***public override String GetInfo()*** *-* переопределили метод ,что бы каждый обьект класса Chocolate мог выдавать информацию конкретно о себе

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| *Выходные величины* | | |
| str | Данные об обьекте класса Chocolate | *String* |

***public override int GetNumPic()-***переопределили метод что бы выбор картинки осуществлялся по цифре

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| *Выходные величины* | | |
| index | Код для выбора картинки | int |

***public static Chocolate Generate()-***метод рандомно генерирующий характеристики обьекта класса Chocolate

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| *Выходные величины* | | |
|  | Новый обьект класса Chocolate,который был создан с помощью генерирования случайных значений | *Chocolate* |

### Класс Baking -потомок класса Sweets

***public override String GetInfo()*** *-* переопределили метод ,что бы каждый объект класса мог выдавать информацию конкретно о себе

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| *Выходные величины* | | |
| str | Данные об объекте класса Baking | *String* |

***public override int GetNumPic()-***переопределили метод что бы выбор картинки осуществлялся по цифре

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| *Выходные величины* | | |
| index | Код для выбора картинки | int |

***public static Baking Generate()-***метод рандомно генерирующий характеристики объекта класса Baking

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| *Выходные величины* | | |
|  | Новый объект класса Baking, который был создан с помощью генерирования случайных значений | *Baking* |

### Класс Fruits -потомок класса Sweets

***public override String GetInfo()*** *-* переопределили метод ,что бы каждый объект класса мог выдавать информацию конкретно о себе

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| *Выходные величины* | | |
| str | Данные об объекте класса Fruits | *String* |

***public override int GetNumPic()-***переопределили метод что бы выбор картинки осуществлялся по цифре

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| *Выходные величины* | | |
| index | Код для выбора картинки | int |

***public static Baking Generate()-***метод рандомно генерирующий характеристики обьекта класса Fruits

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| *Выходные величины* | | |
|  | Новый объект класса Fruits,который был создан с помощью генерирования случайных значений | *Fruits* |

### Form1.cs

**List<Sweets> SweetssList = new List<Sweets>()-** создание листа SweetssList,который может хранить обьекты типа Sweets(Наследование)

**private void Refill\_Click(object sender, EventArgs e)-**метод для кнопки «перезаполнить»(очищает все поля, рандомно заполняет лист обьектами )

**private void ShowInfo()-**функция выводит информацию о количестве сладостей на форму(проходит через весь список и считает сколько осталось и выводит)

**private void btnGet\_Click(object sender, EventArgs e)** -метод ,выполняющийся при нажатии на кнопку "взять"(если список пуст то очищаем все поля, иначе берем из списка первый эелемент, выводим информацию о нем и картинку, выводит очередь и выделяет то ,что взяли)

**public void picture(int a)**- метод для вывода картинок(передается двухзначное число, первая цифра- это шоколад/выпечка/фрукты, вторая цифра это тип каждой из сладостей,находим каждую цифру)

### Код программы

**Класс Sweets**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_4\_1

{

//родительский класс,все что в нем написано наследуют его "дети"

public class Sweets {

public static Random rnd = new Random();

// вес-это общая характеристика

public int weight = 0;

//родительская функция,которую можно переопределять(выдает информацию)

public virtual String GetInfo()

{

var str = String.Format("\nВес:"+ this.weight+ " грамм");

return str;

}

//родительская функция,которую можно переопределять(выдает код для выбора картинки)

public virtual int GetNumPic()

{

var index=0 ;

return index;

}

}

//тип шоколада(темный ,молочный)

public enum ChocolateType { темный, белый };

//наличие начинки(Орехи,изюм)

public enum Stuffing {Орехи, Изюм };

//класс chocolate ,наследует всё от родительского класса Sweets

public class Chocolate : Sweets

{

//наличие начинки

public Stuffing Stuf = Stuffing.Орехи;

// количество плиток

public int BarCount = 0;

// тип шоколада

public ChocolateType Type = ChocolateType.темный;

//переопределили метод что бы каждый предмет выдавал информацию конкретно о себе

public override String GetInfo()

{

var str = "Шоколад";

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

str += base.GetInfo();

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

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

return str;

}

//переопределили метод что бы выбор картинки осуществлялся по цифре

public override int GetNumPic()

{

//индекс равен 10,потому что первая цифра 1-это шоколад,вторая цифра означает тип шоколада

var index = 10;

switch (this.Type) {

case ChocolateType.темный:

index+=1;

break;

case ChocolateType.белый:

index += 2;

break;

}

return index;

}

//метод рандомно генерирующий характеристики шоколада

public static Chocolate Generate()

{

return new Chocolate

{

weight = rnd.Next(10, 200),

Stuf = (Stuffing)rnd.Next(2),

BarCount = rnd.Next() % 10,

Type = (ChocolateType)rnd.Next(2)

};

}

}

//тип выпечки(булочка, пирожок,чизкейк, пончик )

public enum BakingType {булочка, пирожок,чизкейк, пончик };

//класс Baking ,наследует всё от родительского класса Sweets

public class Baking : Sweets

{

//тип выпечки

public BakingType type = BakingType.булочка;

// энергетическая ценность

public int EnergyValue = 0;

//переопределили метод что бы каждый предмет выдавал информацию конкретно о себе

public override String GetInfo()

{

var str = "Выпечка";

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

str += base.GetInfo();

str += String.Format("\nЭнергетическая ценность(ККал): {0}", this.EnergyValue);

return str;

}

//переопределили метод что бы выбор картинки осуществлялся по цифре

public override int GetNumPic()

{ //индекс равен 20,потому что первая цифра 2-это выпечка,вторая цифра означает тип выпечки

var index = 20;

switch (this.type)

{

case BakingType.булочка:

index += 1;

break;

case BakingType.пирожок:

index += 2;

break;

case BakingType.чизкейк:

index += 3;

break;

case BakingType.пончик:

index += 4;

break;

}

return index;

}

//метод рандомно генерирующий характеристики выпечки

public static Baking Generate()

{

return new Baking

{

weight = rnd.Next(100,300) ,

type = (BakingType)rnd.Next(4),

EnergyValue = rnd.Next() % 600

};

}

}

//тип фрукта(яблоко, банан,апельсин)

public enum FruitsType {яблоко, банан,апельсин };

//класс Fruits ,наследует всё от родительского класса Sweets

public class Fruits : Sweets

{

// тип

public FruitsType type = FruitsType.яблоко;

//спелость

public int Ripeness = 0;

//переопределили метод что бы каждый предмет выдавал информацию конкретно о себе

public override String GetInfo()

{

var str = "Фрукт";

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

str += base.GetInfo();

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

return str;

}

//переопределили метод что бы выбор картинки осуществлялся по цифре

public override int GetNumPic()

{

//индекс равен 30,потому что первая цифра 3-это фрукты,вторая цифра означает тип фруктов

var index = 30;

switch (this.type)

{

case FruitsType.яблоко:

index += 1;

break;

case FruitsType.банан:

index += 2;

break;

case FruitsType.апельсин:

index += 3;

break;

}

return index;

}

//метод рандомно генерирующий характеристики фруктов

public static Fruits Generate()

{

return new Fruits

{

weight = rnd.Next(50,200) ,

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

Ripeness = rnd.Next(10,100) ,

};

}

}

}

**Код Form1.cs**

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;

namespace \_4\_1

{

public partial class Form1 : Form

{

//создание листа SweetssList,который может хранить обьекты типа Sweets

List<Sweets> SweetssList = new List<Sweets>();

public Form1()

{

InitializeComponent();

ShowInfo();

}

//метод для кнопки "перезаполнить"

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

{

//очищаем все элементы листа SweetssList

this.SweetssList.Clear();

//очищаем все поля

richTextBox1.Text = "";

txtOut.Text = "";

pictureBox1.Image = null;

var rnd = new Random();

//заполнение листа SweetssList происходит рандомно, всего 10 обьектов

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

{

// генерируются случайные числа от 0 до 2

switch (rnd.Next() % 3)

{

case 0:

//вызов метода генерирования обьекта класса Chocolate для записи в SweetssList

this.SweetssList.Add(Chocolate.Generate());

richTextBox1.Text += "Шоколад" + "\n";

break;

case 1:

//вызов метода генерирования обьекта класса Baking для записи в SweetssList

this.SweetssList.Add(Baking.Generate());

richTextBox1.Text += "Выпечка" + "\n";

break;

case 2:

//вызов метода генерирования обьекта класса Fruits для записи в SweetssList

this.SweetssList.Add(Fruits.Generate());

richTextBox1.Text += "Фрукт" + "\n";

break;

}

}

//вызов метода, который показывает информацию о наличие сладостей в автомате

ShowInfo();

}

// функция выводит информацию о количестве сладостей на форму

private void ShowInfo()

{

// счетчики под каждый тип

int ChocolateCount = 0;

int BakingCount = 0;

int FruitsCount = 0;

// проходим через весь список,если находим нужные предметы,увеличиваем счетчики

foreach (var fruit in this.SweetssList)

{

if (fruit is Chocolate)

{

ChocolateCount += 1;

}

else if (fruit is Baking)

{

BakingCount += 1;

}

else if (fruit is Fruits)

{

FruitsCount += 1;

}

}

// выводим информацию на форму

txtInfo.Text = "Шоколад\tВыпечка\tФрукты";

txtInfo.Text += "\n";

txtInfo.Text += String.Format("{0}\t{1}\t{2}", ChocolateCount, BakingCount, FruitsCount);

}

//метод ,выполняющийся при нажатии на кнопку "взять"

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

{

// если список пуст, то напишем что пусто и выйдем из функции

if (this.SweetssList.Count == 0)

{

txtOut.Text = "Пусто";

richTextBox1.Text = "Пусто";

pictureBox1.Image= null;

textBox1.Text = "";

ShowInfo();

return;

}

//дословно-берем из списка первый предмет

var sweets = this.SweetssList[0];

//вызов метода который возвращает характеристики предмета

txtOut.Text = sweets.GetInfo();

//выводим картинку этого предмета

picture(sweets.GetNumPic());

// очищаем "очередь"

richTextBox1.Text = "";

foreach (var sweet in this.SweetssList)

{

if (sweet is Chocolate)

{

richTextBox1.Text += "Шоколад" + "\n";

}

else if (sweet is Baking)

{

richTextBox1.Text += "Выпечка" + "\n";

}

else if (sweet is Fruits)

{

richTextBox1.Text += "Фрукт" + "\n";

}

}

String s = null;

//находим 0 элемент в листе и исходя из его значений присваиваем строке слово

var fruit = SweetssList[0];

if (fruit is Chocolate)

{s = "Шоколад";}

else if (fruit is Baking)

{s = "Выпечка";}

else if (fruit is Fruits)

{s = "Фрукт";}

//проверяем наличие строки(нашли ее выше) которую нужно окрасить

int pos = richTextBox1.Find(s);

//если искомая строка найдена

if (pos != -1) {

//то окрашиваем эту строку в красный цвет

richTextBox1.Select(0, s.Length);

richTextBox1.SelectionColor= Color.Red;

}

//удаляем нулевой элемент списка

this.SweetssList.RemoveAt(0);

// обновим информацию о количестве товара на форме

ShowInfo();

}

//метод для вывода картинок

public void picture(int a)

{

try

{

//на входе двухначное число,первая цифра это шоколад/выпечка/фрукты,

//вторая цифра это тип каждой из сладостей,находим каждую цифру

int type1 = a / 10;

int type2 = a % 10;

textBox1.Visible = false;

//находим что у нас за предмет,находим картинку этого предмета по указанному адресу и выводим ее

switch (type1)

{

//шоколад

case 1:

switch (type2)

{

case 1:

pictureBox1.Image = Image.FromFile("C:\\Юлия\\Институт\\2 курс\\Технология програмирования\\Лабы\\4 лаба\\Картинки\\темный шоколад.jpg");

break;

case 2:

pictureBox1.Image = Image.FromFile("C:\\Юлия\\Институт\\2 курс\\Технология програмирования\\Лабы\\4 лаба\\Картинки\\белый шоколад.jpg");

break;

}

break;

//выпечка

case 2:

switch (type2)

{

case 1:

pictureBox1.Image = Image.FromFile("C:\\Юлия\\Институт\\2 курс\\Технология програмирования\\Лабы\\4 лаба\\Картинки\\булочка.jpg");

break;

case 2:

pictureBox1.Image = Image.FromFile("C:\\Юлия\\Институт\\2 курс\\Технология програмирования\\Лабы\\4 лаба\\Картинки\\пирожок.jpg");

break;

case 3:

pictureBox1.Image = Image.FromFile("C:\\Юлия\\Институт\\2 курс\\Технология програмирования\\Лабы\\4 лаба\\Картинки\\чизкейк.jpg");

break;

case 4:

pictureBox1.Image = Image.FromFile("C:\\Юлия\\Институт\\2 курс\\Технология програмирования\\Лабы\\4 лаба\\Картинки\\пончик.jpg");

break;

}

break;

//фрукты

case 3:

switch (type2)

{

case 1:

pictureBox1.Image = Image.FromFile("C:\\Юлия\\Институт\\2 курс\\Технология програмирования\\Лабы\\4 лаба\\Картинки\\яблоко.jpg");

break;

case 2:

pictureBox1.Image = Image.FromFile("C:\\Юлия\\Институт\\2 курс\\Технология програмирования\\Лабы\\4 лаба\\Картинки\\банан.png");

break;

case 3:

pictureBox1.Image = Image.FromFile("C:\\Юлия\\Институт\\2 курс\\Технология програмирования\\Лабы\\4 лаба\\Картинки\\апельсин.jpg");

break;

}

break;

}

}

catch(System.IO.FileNotFoundException ex) {

//на входе двухначное число,первая цифра это шоколад/выпечка/фрукты,

//вторая цифра это тип каждой из сладостей,находим каждую цифру

int type1 = a / 10;

int type2 = a % 10;

textBox1.Visible = true;

//находим что у нас за предмет,находим картинку этого предмета по указанному адресу и выводим ее

switch (type1)

{

//шоколад

case 1:

switch (type2)

{

case 1:

textBox1.Text = "Тут должно было быть изображение темного шоколада,но его не удалось загрузить.";

break;

case 2:

textBox1.Text = "Тут должно было быть изображение белого шоколада,но его не удалось загрузить.";

break;

}

break;

//выпечка

case 2:

switch (type2)

{

case 1:

textBox1.Text = "Тут должно было быть изображение булочки,но его не удалось загрузить.";

break;

case 2:

textBox1.Text = "Тут должно было быть изображение пирожка,но его не удалось загрузить.";

break;

case 3:

textBox1.Text = "Тут должно было быть изображение чизкейка,но его не удалось загрузить.";

break;

case 4:

textBox1.Text = "Тут должно было быть изображение пончика,но его не удалось загрузить.";

break;

}

break;

//фрукты

case 3:

switch (type2)

{

case 1:

textBox1.Text = "Тут должно было быть изображение яблока,но его не удалось загрузить.";

break;

case 2:

textBox1.Text = "Тут должно было быть изображение банана,но его не удалось загрузить.";

break;

case 3:

textBox1.Text = "Тут должно было быть изображение апельсина,но его не удалось загрузить.";

break;

}

break;

}

}

}

}

}