Отчет по практической работе №22

Наименование работы: Паттерны проектирования.

Индивидуальные задания (вариант №9):

Паттерн строитель (Builder). Примененим паттерн на примере выпечки хлеба. Как известно, даже обычный хлеб включает множество компонентов. Можем использовать для представления хлеба и его компонентов следующие классы:

- Мука(Flour);

- Соль(Salt);

- Пищевые добавки(Additives);

- Сахар(Sugar);

- Дрожжи(Yeast);

- Хлеб(Bread).

Хлеб может иметь различную комбинацию компонентов: ржаной и пшеничной муки, соли, использованием дрожжи или сахара, пищевых добавок. И вам надо обеспечить выпечку разных сортов хлеба. Для разных сортов хлеба может варьироваться конкретный набор компонентов, не все компоненты могут использоваться. И для этой задачи применим паттерн. Понадобятся следующие классы:

- создатель хлеба(BreadBuilder);

- пекарь(Baker);

- ржаной хлеб(RyeBreadBuilder) – ржаная мука 1 сорт, соль;

- пшеничный хлеб(WheatBreadBuilder) - пшеничная мука высший сорт, соль, добавки: «улучшитель хлебопекарный», сахар;

- Хлеб на дрожжах(YeastBread) - пшеничная мука высший сорт, соль, сахар, сухие дрожжи.

Ход работы:

1. Разработал программу в соответствии с заданием №1. Были реализованы методы, условия, классы, патерны.

Листинг программы:

using System;

using System.Text;

namespace Lab22

{

class Program

{

static void Main(string[] args)

{

Baker baker = new Baker();

BreadBuilder builder = new RyeBreadBuilder();

Bread ryeBread = baker.Bake(builder);

Console.WriteLine(ryeBread.ToString());

builder = new WheatBreadBuilder();

Bread wheatBread = baker.Bake(builder);

Console.WriteLine(wheatBread.ToString());

builder = new YeastBread();

Bread yeastbread = baker.Bake(builder);

Console.WriteLine(yeastbread.ToString());

Console.Read();

}

}

abstract class BreadBuilder

{

public Bread Bread { get; private set; }

public void CreateBread()

{

Bread = new Bread();

}

public abstract void SetFlour();

public abstract void SetSalt();

public abstract void SetAdditives();

public abstract void SetSugar();

public abstract void SetYeast();

}

class Baker

{

public Bread Bake(BreadBuilder breadBuilder)

{

breadBuilder.CreateBread();

breadBuilder.SetFlour();

breadBuilder.SetSalt();

breadBuilder.SetAdditives();

breadBuilder.SetSugar();

breadBuilder.SetYeast();

return breadBuilder.Bread;

}

}

class RyeBreadBuilder : BreadBuilder

{

public override void SetFlour()

{

this.Bread.Flour = new Flour { Sort = "Ржаная мука 1 сорт" };

}

public override void SetSalt()

{

this.Bread.Salt = new Salt();

}

public override void SetAdditives()

{

// не используется

}

public override void SetSugar()

{

// не используется

}

public override void SetYeast()

{

// не используется

}

}

class WheatBreadBuilder : BreadBuilder

{

public override void SetFlour()

{

this.Bread.Flour = new Flour { Sort = "Пшеничная мука высший сорт" };

}

public override void SetSalt()

{

this.Bread.Salt = new Salt();

}

public override void SetAdditives()

{

this.Bread.Additives = new Additives { Name = "улучшитель хлебопекарный" };

}

public override void SetSugar()

{

this.Bread.Sugar = new Sugar();

}

public override void SetYeast()

{

// не используется

}

}

class YeastBread : BreadBuilder

{

public override void SetFlour()

{

this.Bread.Flour = new Flour { Sort = "Пшеничная мука высший сорт" };

}

public override void SetSalt()

{

this.Bread.Salt = new Salt();

}

public override void SetSugar()

{

this.Bread.Sugar = new Sugar();

}

public override void SetAdditives()

{

// не используется

}

public override void SetYeast()

{

this.Bread.Yeast = new Yeast { Name = "Сухие дрожжи" };

}

}

class Bread

{

public Flour Flour { get; set; }

public Salt Salt { get; set; }

public Additives Additives { get; set; }

public Sugar Sugar { get; set; }

public Yeast Yeast { get; set; }

public override string ToString()

{

StringBuilder sb = new StringBuilder();

if (Flour != null)

sb.Append(Flour.Sort + "\n");

if (Yeast != null)

sb.Append(Yeast.Name + "\n");

if (Salt != null)

sb.Append("Соль \n");

if (Sugar != null)

sb.Append("Сахар \n");

if (Additives != null)

sb.Append("Добавки: " + Additives.Name + " \n");

return sb.ToString();

}

}

class Flour { public string Sort { get; set; } }

class Salt { }

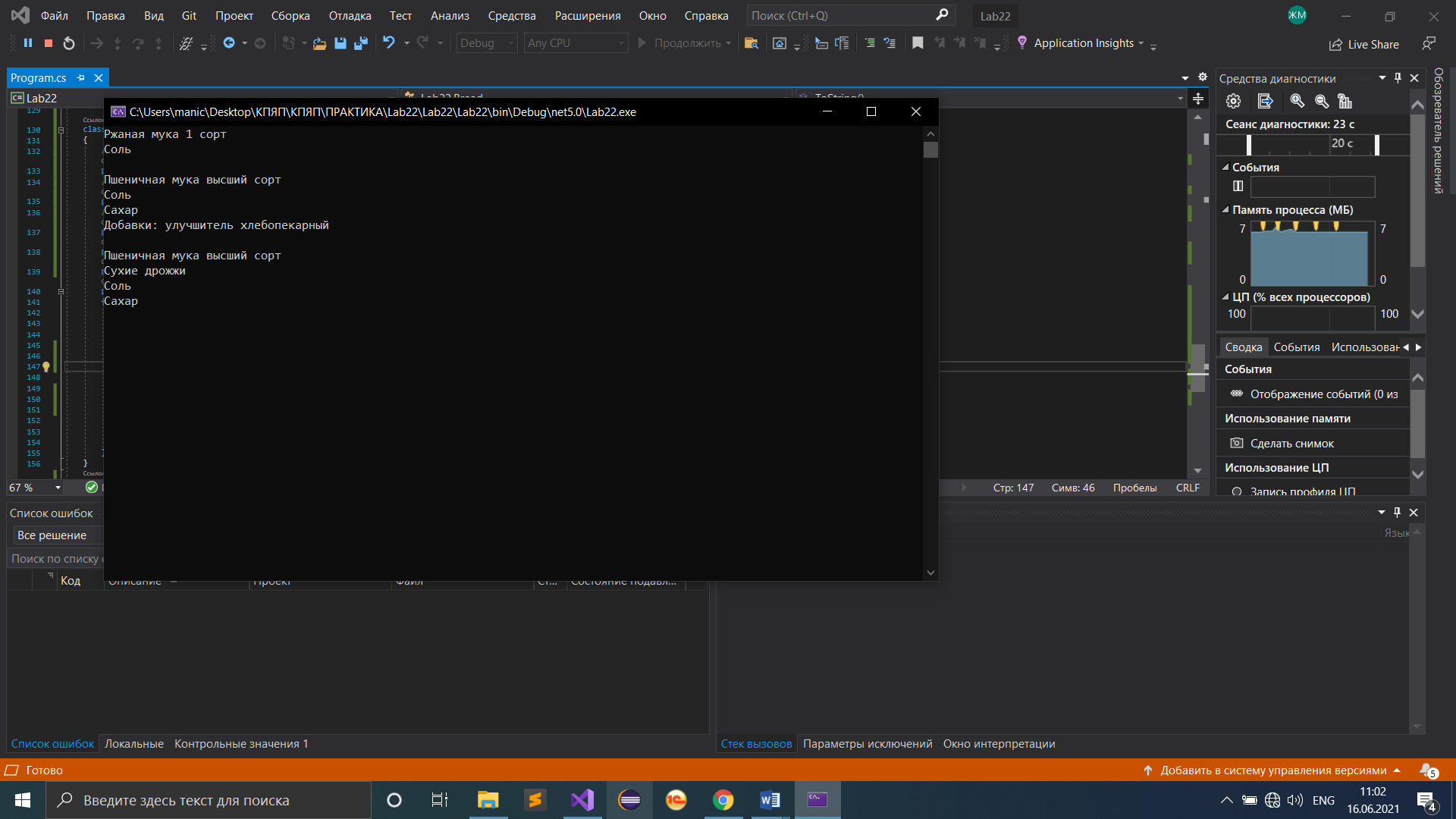
class Sugar { }

class Yeast { public string Name { get; set; } }

class Additives { public string Name { get; set; } }

}

Результат выполнения программы Рис 1.1.



**Рисунок 1.1 – Пример работы программы c потернами**

Отметка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись преподавателя\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

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

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