

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
Институт математики и информационных систем  
Факультет автоматики и вычислительной техники  
Кафедра электронных вычислительных машин

Отчет по лабораторной работе №1  
по дисциплине  
«Объектно-ориентированное программирование»

Выполнил студент гр. ИВТб-2301-05-00 \_\_\_\_\_ /Макаров С.А./  
Руководитель преподаватель \_\_\_\_\_ /Шмакова Н.А./

Киров 2026

## **Цель работы**

Цель работы: изучить и освоить принципы объектно-ориентированного программирования, включая инкапсуляцию, наследование и полиморфизм, путем создания иерархии классов в выбранной предметной области.

## **Задание**

1. Создать иерархию классов состоящую не менее чем из одного родительского и двух дочерних классов.
2. В каждом классе определить не менее двух член данных, не менее двух собственных, а для дочерних не менее двух унаследованных и двух перекрытых член функций.
3. Разработать приложение демонстрирующее принципы инкапсуляции, наследования и полиморфизма.

## **Решение**

Предметная область описывает управление ассортиментом блюд в заведении общественного питания. Система позволяет хранить информацию о различных блюдах, учитывать их специфические характеристики, рассчитывать итоговую цену для клиента в зависимости от этих характеристик и выводить информацию о блюде в удобном виде.

Схема структуры иерархии классов представлена в виде диаграммы классов, представленная на рисунке 1.

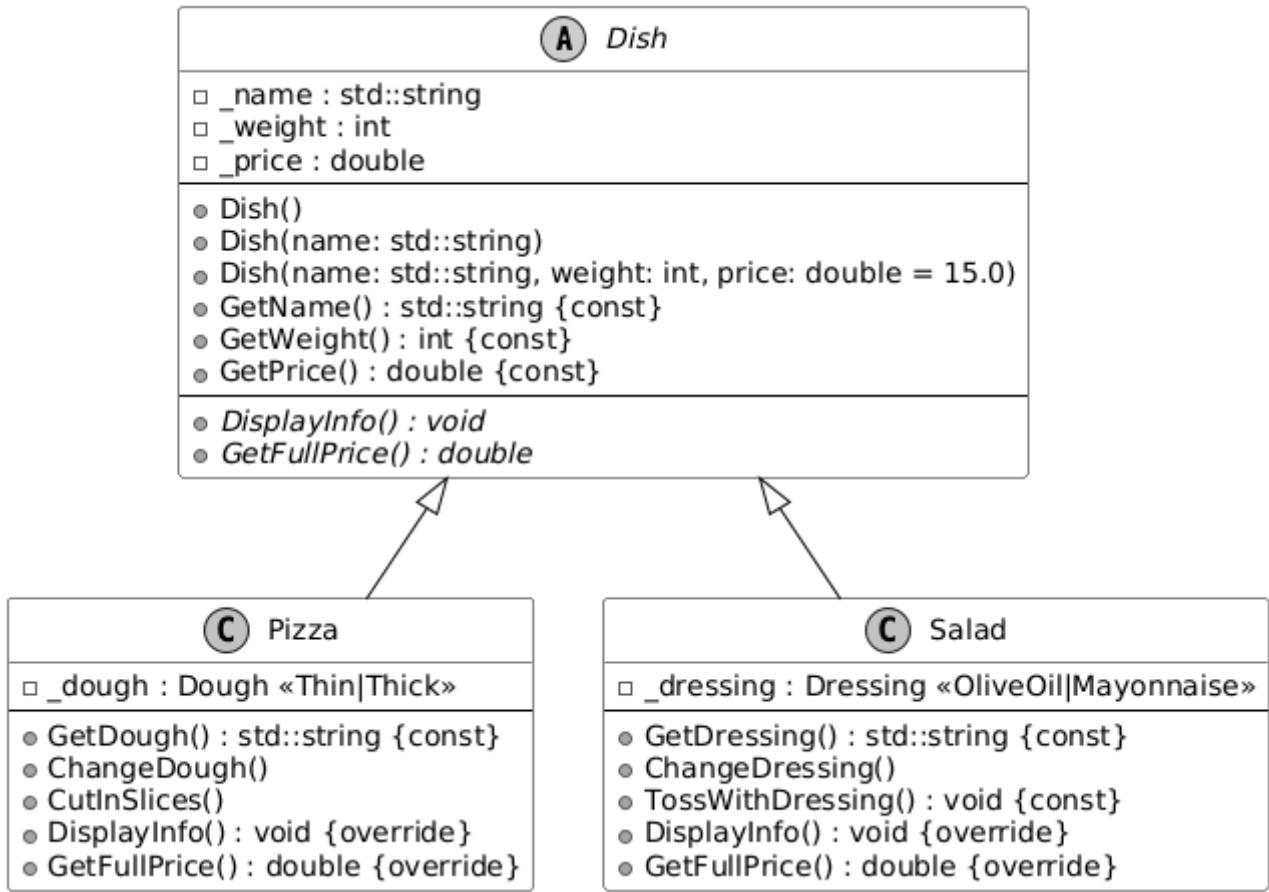


Рисунок 1 – Диаграмма классов

Описание методов классов:

- Класс **Dish**:
  - `Dish()` – конструктор, инициализирует название блюда, вес в граммах и базовую цену;
  - `Dish(string name)` – конструктор, инициализирует название блюда, вес в граммах и базовую цену;
  - `Dish(string name, int weight, double price = 15.0)` – конструктор, инициализирует название блюда, вес в граммах и базовую цену;
  - `string GetName() const` – возвращает название блюда;
  - `int GetWeight() const` – возвращает вес блюда в граммах;
  - `double GetPrice() const` – возвращает базовую цену блюда;
  - `virtual void DisplayInfo() = 0` – виртуальный метод отображения информации;

`virtual double GetFullPrice() = 0` – виртуальный метод, возвращающий полную цену;

– Класс `Pizza` наследуется от `Dish`:

- `Pizza()` – конструктор, передаёт параметры в базовый класс `Dish`, тип теста по умолчанию — `Thick`;
- `Pizza(string name)` – конструктор, передаёт параметры в базовый класс `Dish`, тип теста по умолчанию — `Thick`;
- `Pizza(string name, int weight, double price)` – конструктор, передаёт параметры в базовый класс `Dish`, тип теста по умолчанию — `Thick`;
- `string GetDough() const` – возвращает текстовое описание текущего типа теста;
- `void ChangeDough()` – переключает тип теста между `Thin` и `Thick`;
- `void CutInSlices()` – имитирует действие «нарезать пиццу на куски», выводит соответствующее сообщение;
- `void DisplayInfo() override` – переопределённый метод, выводит информацию о пицце: название, тип теста, вес, итоговая цена;
- `double GetFullPrice() override` – возвращает итоговую цену с учётом типа теста;

– Класс `Salad` наследуется от `Dish`:

- `Salad()` – конструктор, передаёт параметры в базовый класс `Dish`, заправка по умолчанию — `OliveOil`;
- `Salad(string name)` – конструктор, передаёт параметры в базовый класс `Dish`, заправка по умолчанию — `OliveOil`;
- `Salad(string name, int weight, double price)` – конструктор, передаёт параметры в базовый класс `Dish`, заправка по умолчанию — `OliveOil`;
- `string GetDressing() const` – возвращает текстовое описание текущей заправки;

- void ChangeDressing() – переключает вид заправки между OliveOil и Mayonnaise;
- void TossWithDressing() const – имитирует действие «перемешать салат с заправкой», выводит соответствующее сообщение;
- void DisplayInfo() override – переопределённый метод, выводит информацию о салате: название, вид заправки, вес, цена;
- double GetFullPrice() override – возвращает итоговую цену с учётом заправки.

Исходный код программы находится в репозитории GitHub:

<https://github.com/mkstas/VyatSu/tree/master/oop/labaratories/labaratory-work-1>

## Вывод

В ходе выполнения лабораторной работы была разработана и реализована иерархия классов, моделирующая управление ассортиментом блюд в заведении общественного питания. Изучены основные принципы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм. Разработано приложение, демонстрирующее применение данных принципов.