Министерство науки и высшего образования РФ

 Пензенский государственный университет Кафедра "Вычислительная техника"

Отчет по лабораторной работе №3

Работу выполнили студенты группы 23ВВВ3 бригады №9:

Полиневский Вадим

Кизым Иван

Проверили:

Кандидат т.н. Гудков А.А

Кандидат т.н. Евсеева И. Е

Пенза 2025

**Цель работы**: Изучить механизмы наследования в языке C++

**Задаие :**Создать иерархию классов Товар–Игрушка–Продукт–Молочный продукт. Класс Товар должен содержать атрибуты и методы, общие для производных классов. Основная программа должна создавать массивы объектов производных классов и выводить их на экран.

**Описание практической части**:

Для выполнения программы было использовано два модуля: **lab3.cpp** с функцией main и файл заголовка **header.h.**

В файле заголовка прописано 4 класса: **Product**, **Toy**, **Produce**, **Dairy\_Produce**.

Список ирерхии для логичности выполняемой лабораторной работы:

Product -> Toy

Product -> Produce -> Dairy\_Produce.

Дочерние классы на каждом уровне имеют дополнительные поля и методы, помимио методов и полей родительского класса.

Листинг:

**lab3.cpp:**

#include <iostream>

#include <vector>

#include "header.h"

using namespace std;

int main (){

cout << "lab\_3" << endl;

vector <Toy> Toy\_Vector;

vector <Produce> Produce\_Vector;

vector <Dairy\_Produce> Dairy\_Produce\_Vector;

Set\_Info(Toy\_Vector, Produce\_Vector, Dairy\_Produce\_Vector, 3);

Print\_Info(Toy\_Vector, Produce\_Vector, Dairy\_Produce\_Vector, 3);

}

**header.h:**

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Product {

private:

string Id;

string Name;

public:

void set\_id(string \_Id){Id = \_Id;}

void set\_name(string \_Name){Name = \_Name;}

string get\_id(){return Id;}

string get\_name(){return Name;}

};

class Toy : public Product{

private:

string Color;

public:

void set\_color(string \_Color){Color = \_Color;}

string get\_color(){return Color;}

};

class Produce : public Product{

private:

string Expiration\_Date;

public:

void set\_Expiration\_Date (string \_Exapiration\_Date) {Expiration\_Date = \_Exapiration\_Date;}

string get\_Expiration\_Date () {return Expiration\_Date;}

};

class Dairy\_Produce : public Produce {

private:

string Name\_Farm;

public:

void set\_Name\_Farm(string \_Name\_Farm){ Name\_Farm = \_Name\_Farm;}

string get\_Name\_Farm() {return Name\_Farm;}

};

void Enter\_Sting(string Input\_Info, string &Temp\_String) {

cout << Input\_Info << ": ";

cin >> Temp\_String;

}

void Set\_Info(vector <Toy>& temp\_Toy, vector <Produce>& temp\_Produce, vector <Dairy\_Produce>& temp\_Daire\_Produce, unsigned int Count\_Object){

string ID, Name, Expiration\_Date, Color, Name\_Farm;

for (int i = 0; i < Count\_Object; i++) {

cout << "ENTER TOY-OBJECT:" << endl;

Enter\_Sting("ENTER ID:", ID);

Enter\_Sting("ENTER Name\_Product: " , Name);

Enter\_Sting("ENTER Color: " , Color);

Toy temp\_Object;

temp\_Object.set\_name(Name);

temp\_Object.set\_id(ID);

temp\_Object.set\_color(Color);

temp\_Toy.push\_back(temp\_Object);

}

for (int i = 0; i < Count\_Object; i++) {

cout << "ENTER PRODUCE-OBJECT:" << endl;

Enter\_Sting("ENTER ID:", ID);

Enter\_Sting("ENTER Name\_Product: " , Name);

Enter\_Sting("ENTER Expiration\_Date: " , Expiration\_Date);

Produce temp\_Object;

temp\_Object.set\_id(ID);

temp\_Object.set\_name(Name);

temp\_Object.set\_Expiration\_Date(Expiration\_Date);

temp\_Produce.push\_back(temp\_Object);

}

for (int i = 0; i < Count\_Object; i++) {

cout << "ENTER DAIRE\_PRODUCE-OBJECT:" << endl;

Enter\_Sting("ENTER ID:", ID);

Enter\_Sting("ENTER Name\_Product: " , Name);

Enter\_Sting("ENTER Expiration\_Date: " , Expiration\_Date);

Enter\_Sting("ENTER Name\_Farm: " , Name\_Farm);

Dairy\_Produce temp\_Object;

temp\_Object.set\_id(ID);

temp\_Object.set\_name(Name);

temp\_Object.set\_Expiration\_Date(Expiration\_Date);

temp\_Object.set\_Name\_Farm(Name\_Farm);

temp\_Daire\_Produce.push\_back(temp\_Object);

}

}

void Print\_Info(vector <Toy> temp\_Toy, vector <Produce> temp\_Produce, vector <Dairy\_Produce> temp\_Daire\_Produce, unsigned int Count\_Object) {

cout << "TOYVECTOR: "<< endl;

for (int i = 0; i < Count\_Object; i++) {

cout << i << ": " << temp\_Toy[i].get\_id() << ", " << temp\_Toy[i].get\_name() << ", " << temp\_Toy[i].get\_color() << endl;

}

cout << "PRODUCE VECTOR:"<< endl;

for (int i = 0; i < Count\_Object; i++) {

cout << i << ": " << temp\_Produce[i].get\_id() <<", "<< temp\_Produce[i].get\_name() << ", " << temp\_Produce[i].get\_Expiration\_Date() << endl;

}

cout << "DAIRE PRODUCE VECTOR:" << endl;

for (int i = 0; i < Count\_Object; i++) {

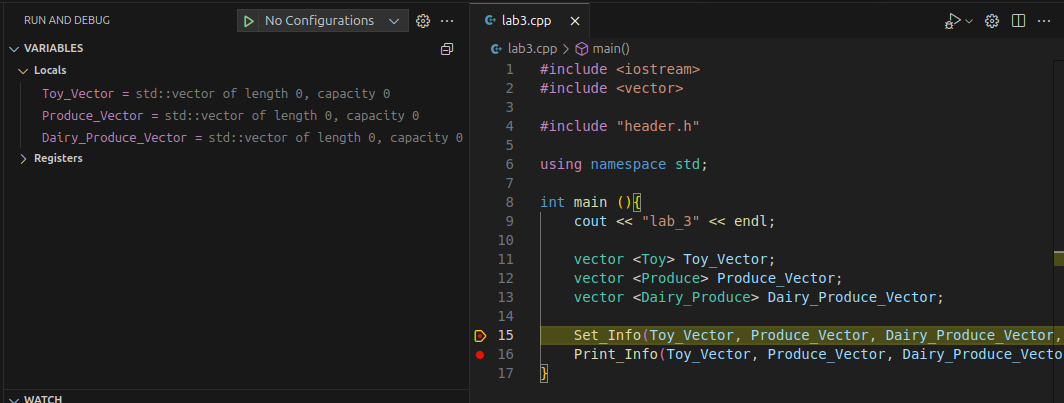
cout << i << ": "<< temp\_Daire\_Produce[i].get\_id()<< ", " << temp\_Daire\_Produce[i].get\_name() << ", "<< temp\_Daire\_Produce[i].get\_Expiration\_Date() << ", "<< temp\_Daire\_Produce[i].get\_Name\_Farm()<< endl;

}

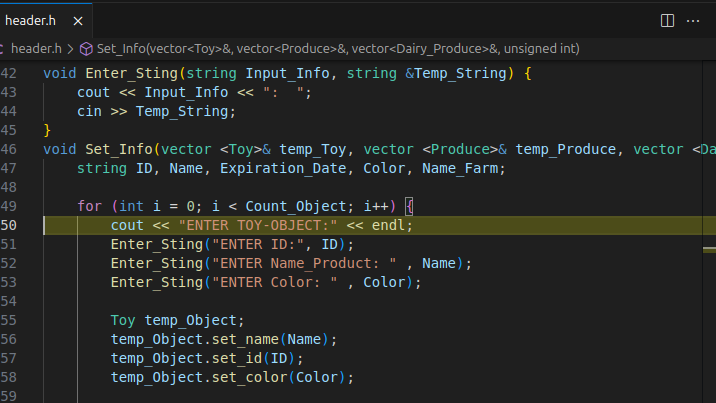
}

**Отладка:**

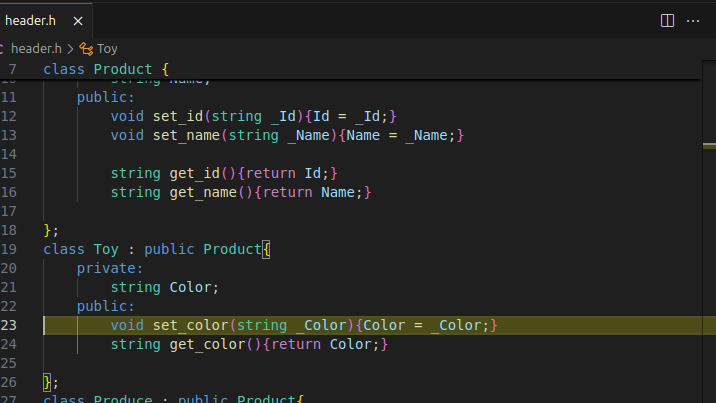
Первая точка остановки, векторы с объектами Дочерних классов.

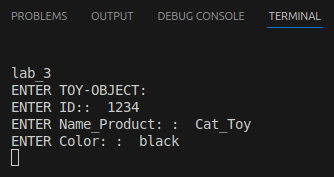
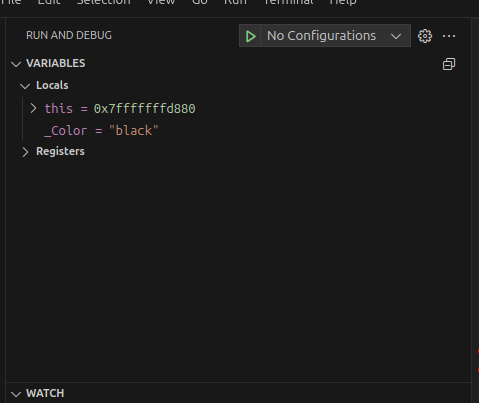


Вторая точка остановки, до пользовательского ввода для полей класса Toy:

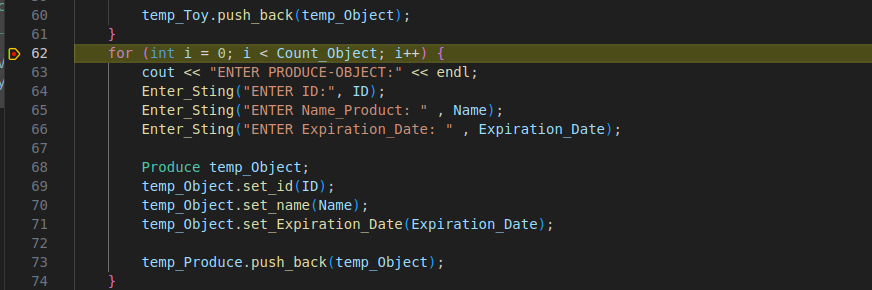


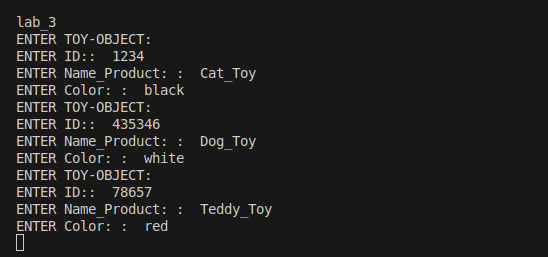
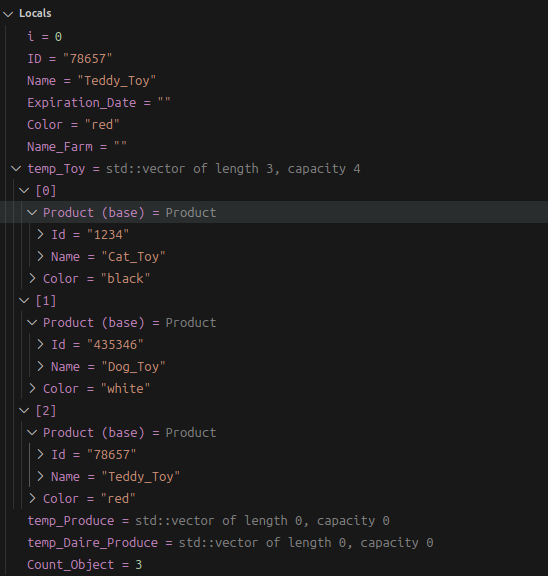
Третья точка остановки, передача в неунаследованное поле:





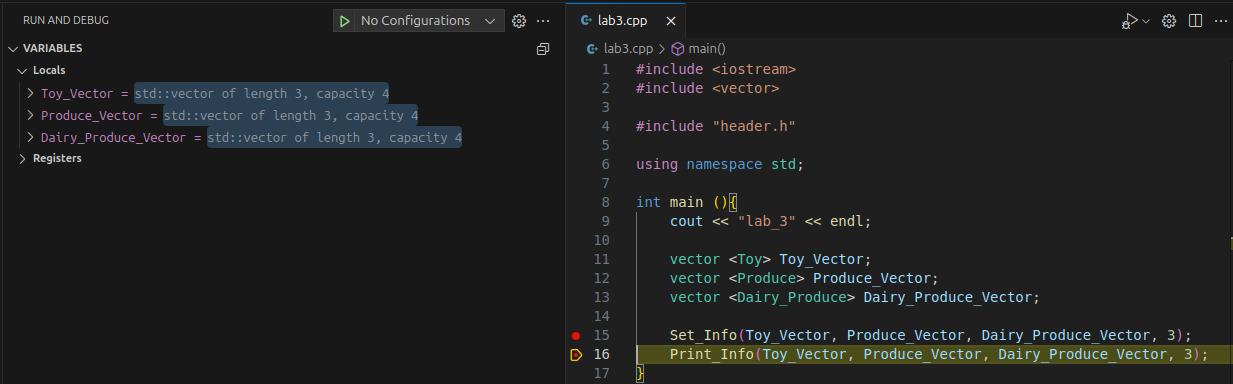
Четвертая точка остановки: конец пользовательского ввода для Toy, на данном шаге Count\_Object повторений:



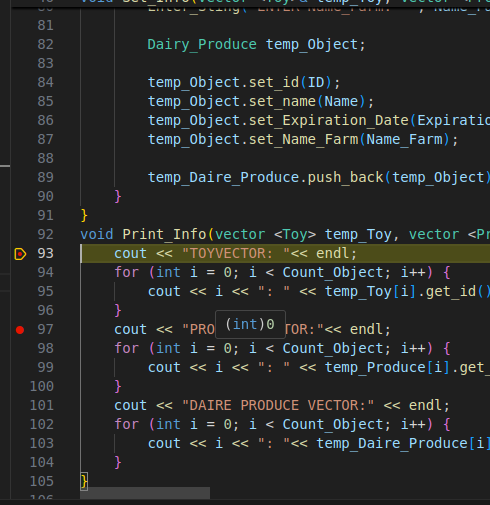
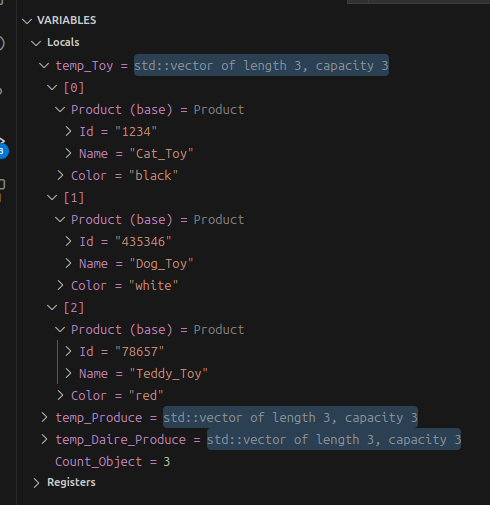


Аналагично для других векторов.

Завершение заполнения векторов:



Начало вывода для вектор с объектами типа класс Toy:



**Terminal-Log:**

lab\_3

ENTER TOY-OBJECT:

ENTER ID:: 1234

ENTER Name\_Product: : Cat\_Toy

ENTER Color: : black

ENTER TOY-OBJECT:

ENTER ID:: 435346

ENTER Name\_Product: : Dog\_Toy

ENTER Color: : white

ENTER TOY-OBJECT:

ENTER ID:: 78657

ENTER Name\_Product: : Teddy\_Toy

ENTER Color: : red

ENTER PRODUCE-OBJECT:

ENTER ID:: 78567

ENTER Name\_Product: : meat

ENTER Expiration\_Date: : 13.05.24

ENTER PRODUCE-OBJECT:

ENTER ID:: 89097

ENTER Name\_Product: : chips

ENTER Expiration\_Date: : 12.09.25

ENTER PRODUCE-OBJECT:

ENTER ID:: 78234

ENTER Name\_Product: : beer

ENTER Expiration\_Date: : 12.12.23

ENTER DAIRE\_PRODUCE-OBJECT:

ENTER ID:: 34123

ENTER Name\_Product: : curd

ENTER Expiration\_Date: : 13.04.25

ENTER Name\_Farm: : ooo123

ENTER DAIRE\_PRODUCE-OBJECT:

ENTER ID:: Milk

ENTER Name\_Product: : Milk

ENTER Expiration\_Date: : jjj1

ENTER Name\_Farm: : jjj123123

ENTER DAIRE\_PRODUCE-OBJECT:

ENTER ID:: 891428

ENTER Name\_Product: : yogurt

ENTER Expiration\_Date: : 13/04.12

ENTER Name\_Farm: : Q190

TOYVECTOR:

0: 1234, Cat\_Toy, black

1: 435346, Dog\_Toy, white

2: 78657, Teddy\_Toy, red

PRODUCE VECTOR:

0: 78567, meat, 13.05.24

1: 89097, chips, 12.09.25

2: 78234, beer, 12.12.23

DAIRE PRODUCE VECTOR:

0: 34123, curd, 13.04.25, ooo123

1: Milk, Milk, jjj1, jjj123123

2: 891428, yogurt, 13/04.12, Q190

[1] + Done "/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<"/tmp/Microsoft-MIEngine-In-zsoe52rj.wlm" 1>"/tmp/Microsoft-MIEngine-Out-bwbfpyub.uuj"