ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

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

Кафедра защиты информации

**

**ОТЧЁТ**

**по лабораторной работе № 8**

**«Многопоточность, конкурентность и параллелизм»**

**по дисциплине: «***Программирование***»**

Выполнил:Проверил:

Студент гр. «АБ-121», «АВТФ» *доцент кафедры ЗИ*

*Новиков Втюрин Александр Романович Архипова А. Б.*

«21» декабря 2022г«\_\_\_» \_\_\_\_\_\_ 2022 г.

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

(подпись) (подпись)

Новосибирск 2022

**Цели и задачи работы:**изучение принципов реализации многопоточности и практические применение возможностей языков высокого уровня (C++ или другой на выбор студента) при моделировании бизнес-процессов.

**Задание 1 (5 баллов)**

В рамках лабораторной работы необходимо выполнить анализ предметной области выделить бизнес-процессы, которые могут выполняться в режиме многопоточности. Построить 2 модели системы с топологиями звезда и кольцо. И выполнить программную реализацию в 2 вариантах, однопоточную и многопоточную и в рамках её провести сравнительный анализ детерминированности системы.

**Вариант 5:**

Продажа товаров продавцом

**С++**

**Main.cpp**

#include "WithoutThr.h"

#include "Ring.h"

#include "Star.h"

int main()

{

while (true)

{

system("cls");

cout << "What to do?" << endl << "1)Multithreading" << endl << "2)Single-threaded" << endl << "3)Exit" << endl;

int choice;

cin >> choice;

if (cin.fail())

{

exit(0);

}

else if (choice == 1)

{

system("cls");

cout << "What topology should I perform?" << endl << "1)Ring" << endl << "2)Star" << endl << "3)Exit" << endl;

int topology;

cin >> topology;

if (cin.fail())

{

exit(0);

}

else if (topology == 1)

{

system("cls");

SellRing sell;

sell.ring();

}

else if (topology == 2)

{

system("cls");

SellStar sell;

sell.star();

}

else continue;

}

else if (choice == 2)

{

system("cls");

Sell sell;

sell.withoutThr();

}

else if (choice == 3)

{

exit(0);

}

else continue;

}

}}

**Ring.h**

#pragma once

#include <iostream>

#include <thread>

#include <vector>

#include <random>

#include <mutex>

#include <ctime>

using namespace std;

class SellRing

{

public:

void sell();

void ring();

private:

int earnings = 0;

mutex mx;

};

**Star.h**

#pragma once

#include <iostream>

#include <thread>

#include <vector>

#include <random>

#include <ctime>

using namespace std;

class SellStar

{

public:

void sell();

void star();

private:

int earnings = 0;

};

**WithoutThr.h**

#pragma once

#include <iostream>

#include <random>

#include <vector>

#include <ctime>

using namespace std;

class Sell

{

public:

void sell();

void withoutThr();

private:

int earnings = 0;

};

**Ring.cpp**

#include "Ring.h"

void SellRing::sell()

{

mx.lock();

vector<pair<string, int>> products = { {"Water", 30}, {"Apple", 150}, {"Milk", 90}, {"Glue", 200}, {"Solvent", 150}, {"gamma-Hydroxybutyric acid", 1000} , {"2,3,7,8-Tetrachlorodibenzodioxin", 5000} , {"1,2-Dichloroethane", 110} };

srand(time(0));

int chosenProducts = rand() % 7;

cout << "Cash: ";

cout << "The product " << products[chosenProducts].first << " was bought" << endl;

earnings += products[chosenProducts].second;

cout << "The total earnings are " << earnings << "Rub" << endl << endl;

mx.unlock();

}

void SellRing::ring()

{

for (int i = 0; i < 2; i++)

{

for (int i = 0; i < 2; i++)

{

thread th1([&]() {

for (int i = 0; i < 5; i++)

{

sell();

}

});

thread th2([&]() {

for (int i = 0; i < 5; i++)

{

sell();

}

});

thread th3([&]() {

for (int i = 0; i < 5; i++)

{

sell();

}

});

th1.join();

th2.join();

th3.join();

}

}

cout << "Time: " << clock();

system("pause");

}

**Star.cpp**

#include "Star.h"

void SellStar::sell()

{

vector<pair<string, int>> products = { {"Water", 30}, {"Apple", 150}, {"Milk", 90}, {"Glue", 200}, {"Solvent", 150}, {"gamma-Hydroxybutyric acid", 1000} , {"2,3,7,8-Tetrachlorodibenzodioxin", 5000} , {"1,2-Dichloroethane", 110} };

srand(time(0));

int chosenProducts = rand() % 7;

cout << "The product " << products[chosenProducts].first << " was bought" << endl;

earnings += products[chosenProducts].second;

cout << "The total earnings are " << earnings << "Rub" << endl << endl;

}

void SellStar::star() {

for (int i = 0; i < 2; i++)

{

thread th1([&]() {

for (int i = 0; i < 5; i++)

{

cout << "Cash : ";

sell();

}

});

thread th2([&]() {

for (int i = 0; i < 5; i++)

{

cout << "Cash 2: ";

sell();

}

});

thread th3([&]() {

for (int i = 0; i < 5; i++)

{

cout << "Cash 3: ";

sell();

}

});

th1.join();

th2.join();

th3.join();

}

cout << "Time: " << clock();

system("pause");

}

**WithoutThr.cpp**

#include "WithoutThr.h"

void Sell::sell()

{

vector<pair<string, int>> products = { {"Water", 30}, {"Apple", 150}, {"Milk", 90}, {"Glue", 200}, {"Solvent", 150}, {"gamma-Hydroxybutyric acid", 1000} , {"2,3,7,8-Tetrachlorodibenzodioxin", 5000} , {"1,2-Dichloroethane", 110} };

srand(time(0));

int chosenProducts = rand() % 7;

cout << "The product " << products[chosenProducts].first << " was bought" << endl;

earnings += products[chosenProducts].second;

cout << "The total earnings are " << earnings << "Rub" << endl << endl;

}

void Sell::withoutThr()

{

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

{

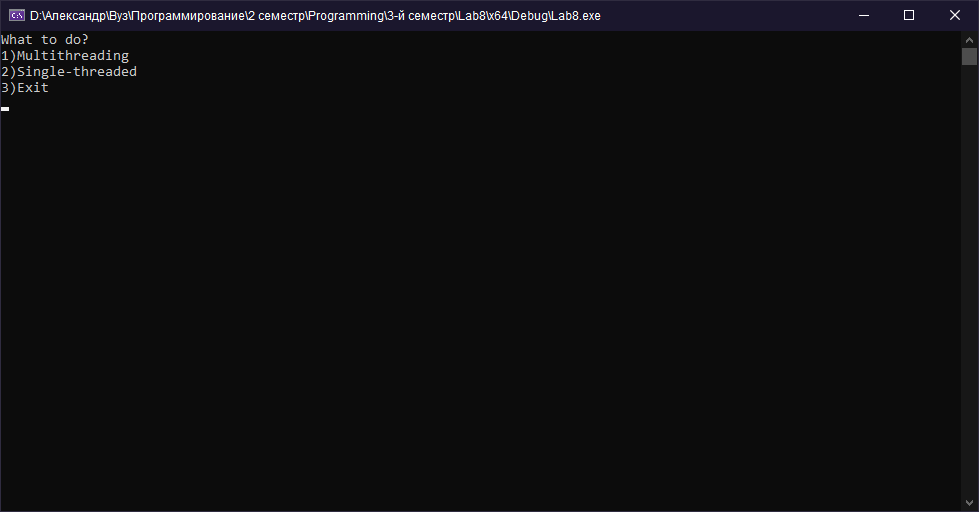
sell();

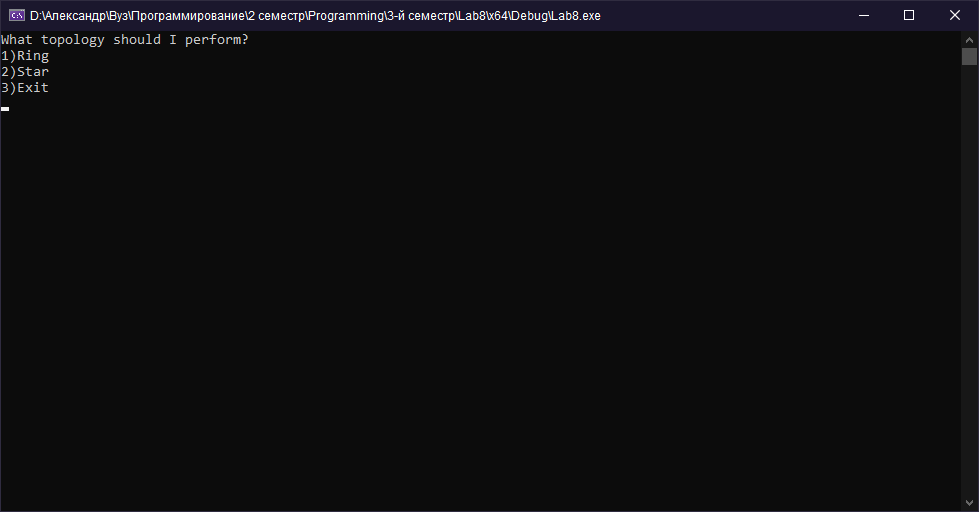
}

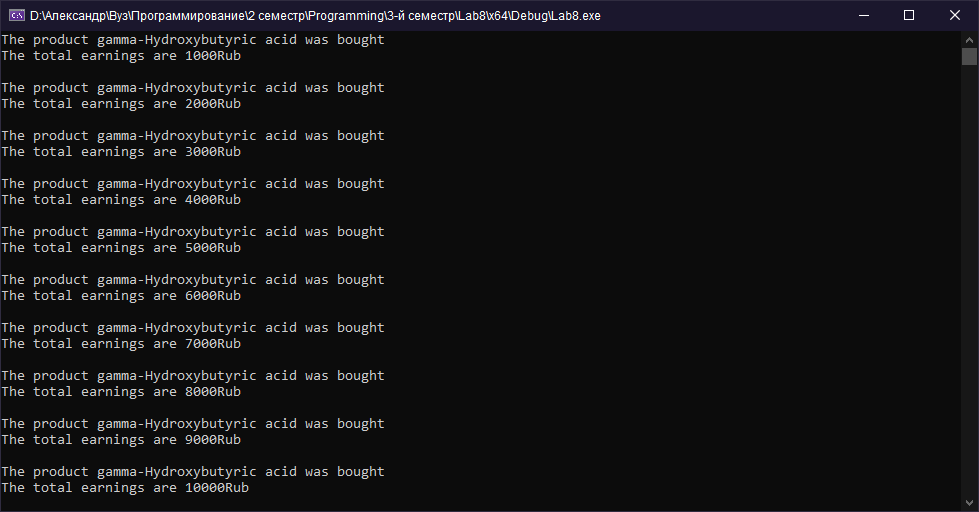
cout << "Time: " << clock();

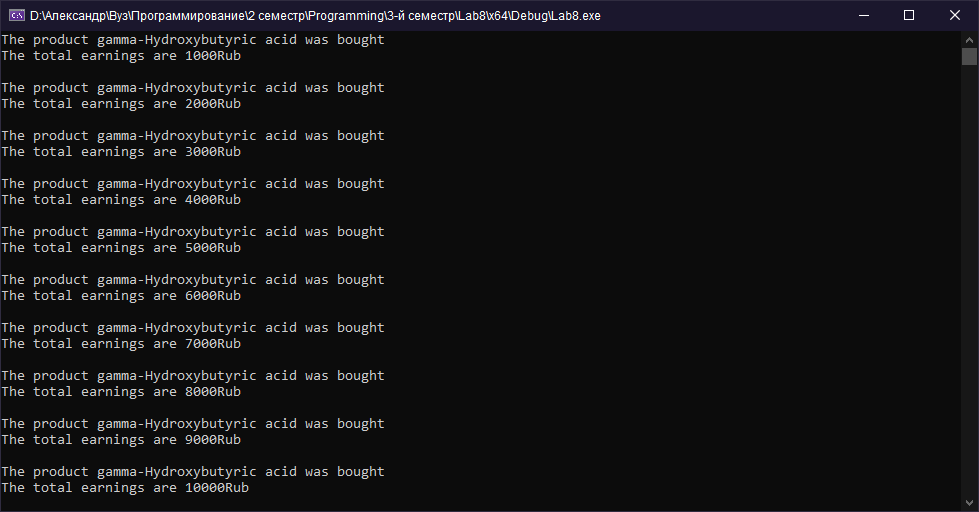
system("pause");

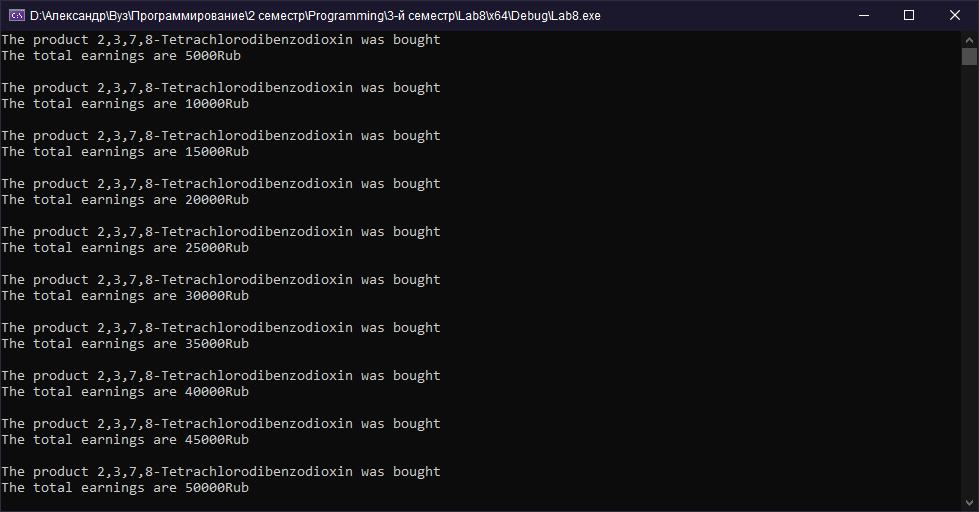
}

****

****

****

****

****

**C#**

**Main.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab8Sharp

{

internal class Programm

{

static void Main(string[] args)

{

while (true)

{

Console.Clear();

Console.WriteLine("What to do?");

Console.WriteLine("1)Multithreading");

Console.WriteLine("2)Single-threaded");

Console.WriteLine("3)Exit");

int choice = int.Parse(Console.ReadLine());

if (choice == 1)

{

Console.Clear();

Console.WriteLine("What topology should I perform?");

Console.WriteLine("1)Ring");

Console.WriteLine("2)Star");

Console.WriteLine("3)Exit");

int topology = int.Parse(Console.ReadLine());

if (topology == 1)

{

SellRing sell = new SellRing();

Console.Clear();

sell.ring();

}

else if (topology == 2)

{

SellStar sell = new SellStar();

Console.Clear();

sell.star();

}

else continue;

}

else if (choice == 2)

{

SellWithoutThr sell = new SellWithoutThr();

Console.Clear();

sell.withoutThr();

}

else if (choice == 3)

{

Environment.Exit(0);

}

else continue;

}

}

}

}

**Ring.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab8Sharp

{

public class SellRing

{

private Mutex mutexObj = new Mutex();

public void sell()

{

mutexObj.WaitOne();

List<string> product = new List<string> { "Water", "Apple", "Milk", "Glue", "Solvent", "gamma-Hydroxybutyric acid", "2,3,7,8-Tetrachlorodibenzodioxin", "1,2-Dichloroethane" };

List<int> money = new List<int> { 30, 150, 90, 200, 150, 1000, 5000 };

Random rnd = new Random();

int chosenProduct = rnd.Next(0, 7);

Console.Write("The product ");

Console.Write(product[chosenProduct]);

Console.Write(" was bought\n");

earnings += money[chosenProduct];

Console.Write("The total earnings are ");

Console.Write(earnings);

Console.Write("Rub\n\n");

mutexObj.ReleaseMutex();

}

private int earnings = 0;

public void ring()

{

SellRing sell = new SellRing();

for (int j = 0; j < 10; j++)

{

ThreadStart threadStart = new ThreadStart(sell.sell);

Thread thread = new Thread(threadStart);

thread.Start();

Thread.Sleep(10);

}

Console.ReadKey();

}

}

}

**Star.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab8Sharp

{

public class SellStar

{

public void sell()

{

List<string> product = new List<string> { "Water", "Apple", "Milk", "Glue", "Solvent", "gamma-Hydroxybutyric acid", "2,3,7,8-Tetrachlorodibenzodioxin", "1,2-Dichloroethane" };

List<int> money = new List<int> { 30, 150, 90, 200, 150, 1000, 5000 };

Random rnd = new Random();

int chosenProduct = rnd.Next(0, 7);

Console.Write("The product ");

Console.Write(product[chosenProduct]);

Console.Write(" was bought\n");

earnings += money[chosenProduct];

Console.Write("The total earnings are ");

Console.Write(earnings);

Console.Write("Rub\n\n");

}

private int earnings = 0;

public void star()

{

SellStar sell = new SellStar();

for (int j = 0; j < 10; j++)

{

ThreadStart threadStart = new ThreadStart(sell.sell);

Thread thread = new Thread(threadStart);

thread.Start();

Thread.Sleep(10);

}

Console.ReadKey();

}

}

}

**WithoutThr.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab8Sharp

{

public class SellWithoutThr

{

public void sell()

{

List<string> product = new List<string> { "Water", "Apple", "Milk", "Glue", "Solvent", "gamma-Hydroxybutyric acid", "2,3,7,8-Tetrachlorodibenzodioxin", "1,2-Dichloroethane" };

List<int> money = new List<int> { 30, 150, 90, 200, 150, 1000, 5000 };

Random rnd = new Random();

int chosenProduct = rnd.Next(0, 7);

Console.Write("The product ");

Console.Write(product[chosenProduct]);

Console.Write(" was bought\n");

earnings += money[chosenProduct];

Console.Write("The total earnings are ");

Console.Write(earnings);

Console.Write("Rub\n\n");

}

private int earnings = 0;

public void withoutThr()

{

SellWithoutThr sell = new SellWithoutThr();

for (int j = 0; j < 10; j++)

{

sell.sell();

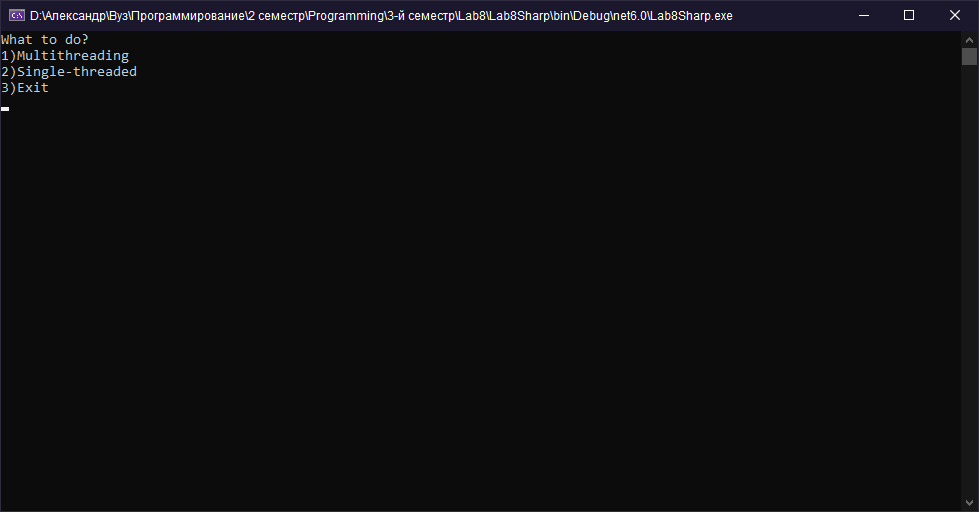
}

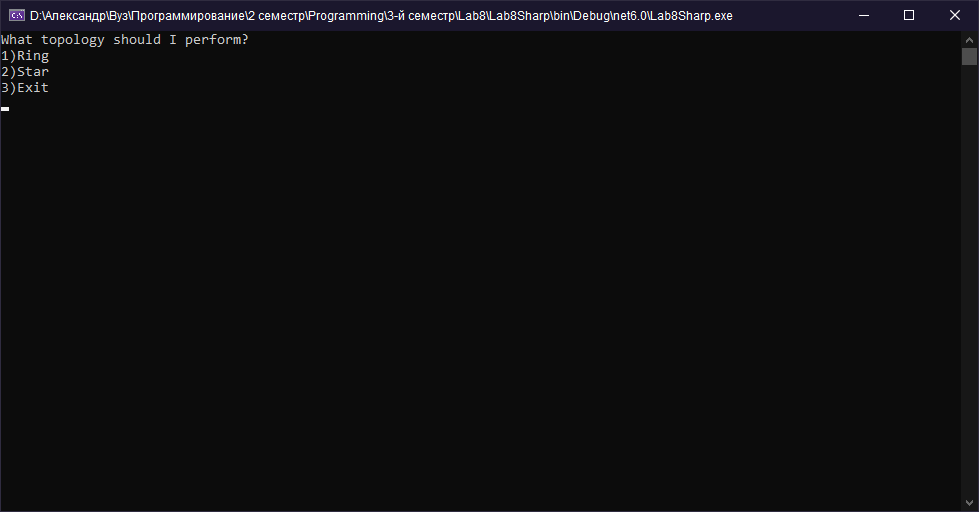
Console.ReadKey();

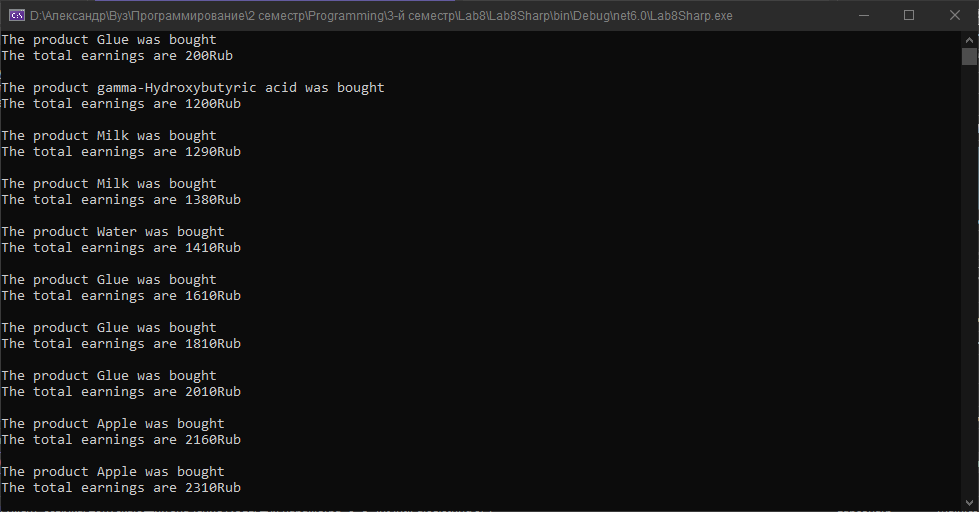
}

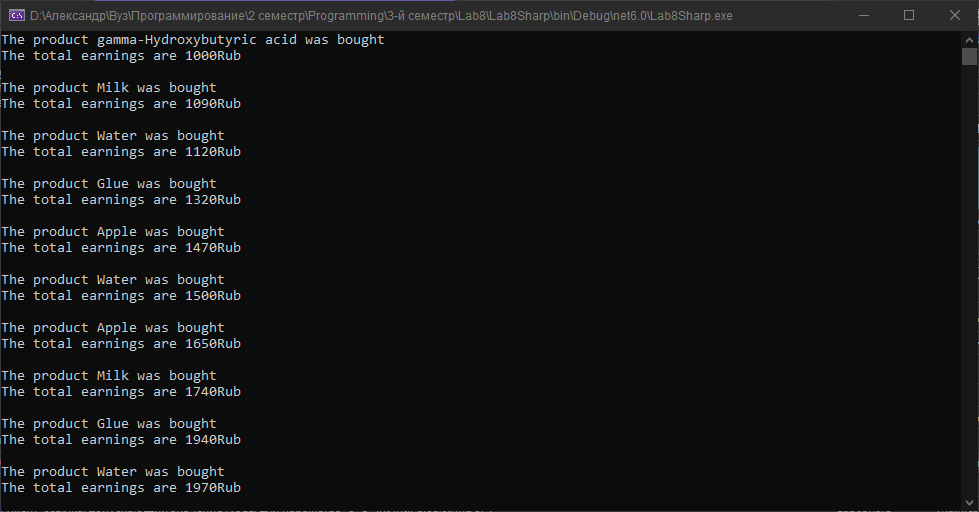
}

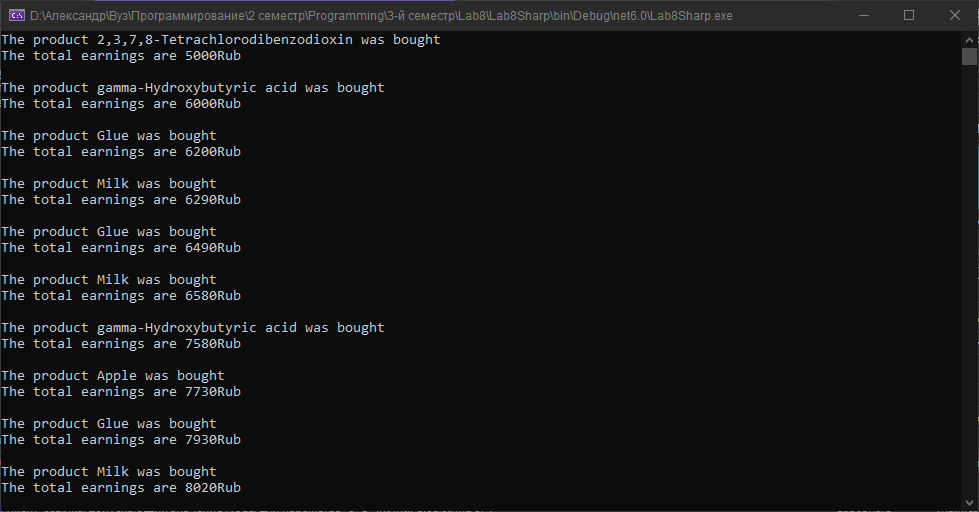
}

****

****

****

****

****