**Министерство общего и профессионального образования Свердловской области**

**Государственное автономное профессиональное образовательное учреждение Свердловской области «Уральский политехнический колледж – Межрегиональный центр компетенций»**

**(ГАПОУ СО «Уральский политехнический колледж - МЦК»)**

**Практическая работа № 3**

**по дисциплине МДК.01.01 «Разработка программных модулей»**

**на тему:**

**«Перегрузка методов**

**Преподаватель: Морозов В.В.**

**Студент: Ветошкин Н.О.**

**Группа: ИП-391к**

Екатеринбург

2025

**Вариант 3**

**Текст задания:**

Создать класс с произвольным именем, содержащий набор математических операций над целыми или вещественными числами, но с перегруженными версиями методов для разных типов данных и разного количества аргументов.

**Текст программы:**

#include <iostream>

#include <vector>

#include <string>

#include <windows.h>

#include <fstream>

using namespace std;

int add(int a, int b) {

return a + b;

}

double add(double a, double b) {

return a + b;

}

int add(int a, int b, int c) {

return a + b + c;

}

double average(const vector<double>& values) {

double flag1 = 0;

double flag2 = 0;

for (int i = 0; i < values.size(); ++i) {

flag1++;

flag2 += values[i];

}

return flag2 / flag1;

}

string concat(const string& a, const string& b) {

return a + b;

}

string concat(const vector<string>& strs) {

string flag3;

for (int i = 0; i < strs.size(); ++i) {

flag3 += strs[i];

}

return flag3;

}

int main()

{

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

ofstream MyFile("Перегрузка\_методов.txt", ios::app);

cout << "Выберите действие:\n1 - Сложение двух целых чисел; 2 - Сложение двух вещественных чисел; 3 - Сумма трёх целых;\n4 - Среднее арифметическое любого количества вещественных значений (передаётся вектор);\n5 - Конкатенация двух строк; 6 - Конкатенация произвольного количества строк (передаётся вектор): ";

int flag;

cin >> flag;

cin.ignore();

switch (flag) {

case(1): {

int a, b;

MyFile << "Операция: Сложение двух целых чисел\n";

cout << "Введите два целых числа: ";

cin >> a >> b;

MyFile << "Аргументы: " << a << "; " << b << "\n";

cin.ignore();

cout << "Результат сложения двух целых чисел: " << add(a, b);

MyFile << "Результат: " << add(a, b) << b << "\n";

MyFile << "-----------------------------------\n";

}

break;

case(2): {

double a, b;

MyFile << "Операция: Сложение двух вещественных чисел\n";

cout << "Введите два вещественных числа: ";

cin >> a >> b;

MyFile << "Аргументы: " << a << "; " << b << "\n";

cin.ignore();

cout << "Результат сложения двух вещественных чисел: " << add(a, b);

MyFile << "Результат: " << add(a, b) << "\n";

MyFile << "-----------------------------------\n";

}

break;

case(3): {

int a, b, c;

MyFile << "Операция: Сложение трех целых чисел\n";

cout << "Введите три целых числа: ";

cin >> a >> b >> c;

cin.ignore();

MyFile << "Аргументы: " << a << "; " << b << "; " << c << "\n";

cout << "Результат сложения трех целых чисел: " << add(a, b, c);

MyFile << "Результат: " << add(a, b, c) << "\n";

MyFile << "-----------------------------------\n";

}

break;

case(4): {

int razmer;

MyFile << "Операция: Среднее арифметическое\n";

cout << "Введите размер массива: ";

cin >> razmer;

cin.ignore();

vector<double> values(razmer);

string x;

cout << "Введите вещественные числа: ";

MyFile << "Аргументы: ";

for (int i = 0; i < razmer; ++i) {

cin >> values[i];

cin.ignore();

MyFile << values[i] << " ";

}

cout << "Среднее арифметическое всех вещественных значений: " << average(values);

MyFile << "\nРезультат: " << average(values) << "\n";

MyFile << "-----------------------------------\n";

}

break;

case(5): {

string a, b;

MyFile << "Операция: Конкатенация двух строк\n";

cout << "Введите две строки\n";

cout << "Первая строка: ";

getline(cin, a);

cout << "Вторая строка: ";

getline(cin, b);

MyFile << "Аргументы: первая строка: '" << a << "'; вторая строка: '" << b << "'\n";

cout << "Результат конкатенации двух строк: " << concat(a, b);

MyFile << "Результат: " << concat(a, b) << "\n";

MyFile << "-----------------------------------\n";

}

break;

case(6): {

int razmer;

MyFile << "Операция: Конкатенация произвольного числа строк строк\n";

cout << "Введите размер массива: ";

cin >> razmer;

cin.ignore();

vector<string> strs(razmer);

string x;

cout << "Введите все элементы массива: ";

for (int i = 0; i < razmer; ++i) {

getline(cin, strs[i]);

x += '"' + strs[i] + '"' + " ";

}

MyFile << "Аргументы: " << x << "\n";

cout << "Результат конкатенации произвольного числа строк: " << concat(strs);

MyFile << "Результат: " << concat(strs) << "\n";

MyFile << "-----------------------------------\n";

}

break;

}

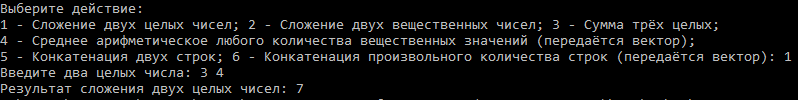
return 0;

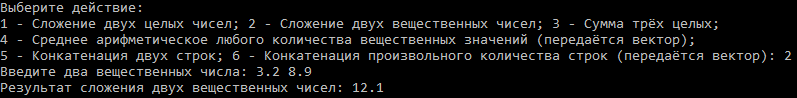
}

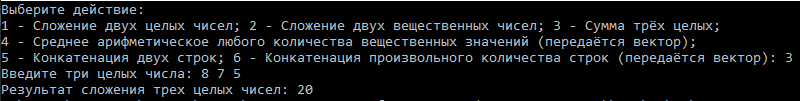
**Ссылка на репозиторий в GitHub:**

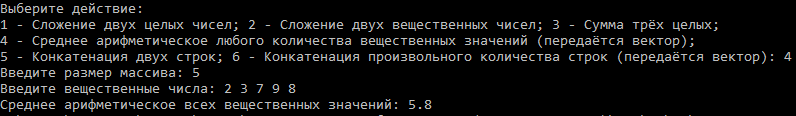
<https://github.com/RikkiTikkki/System-programming>

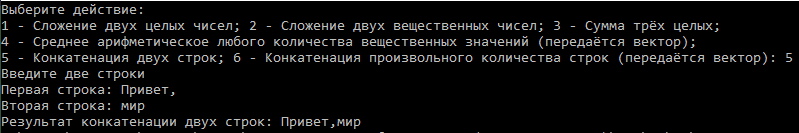
**Скриншоты работы программы:**

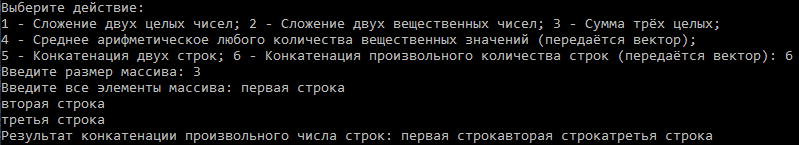
****

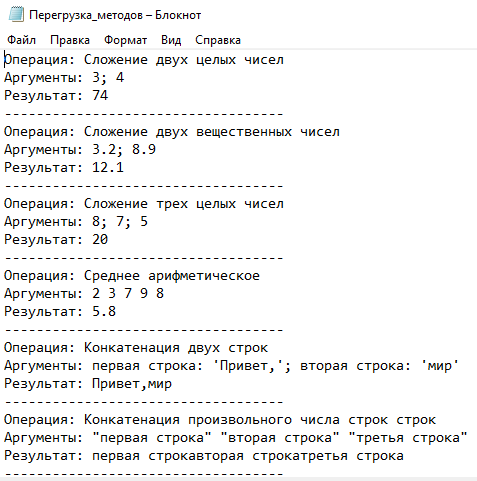
****

****

****

****

****

****

**Вывод:**

В процессе выполнения практической работы я изучил механизм перегрузки функций/методов в C++ и научился применять его для повышения гибкости и удобства интерфейсов классов.