

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК ИМЕНИ
ПРОФЕССОРА Н.И.ЧЕРВЯКОВА**

ЛАБОРАТОРНАЯ РАБОТА №20

Алгоритмизация и программирование

Модули

Выполнил студент:

Сивко Иван Андреевич

студент 2 курса

группа ПМИ-б-о-23-2,

направление подготовки 01.03.02

Проверил:

Ассистент кафедры вычислительной
математики и кибернетики, к.ф.-м.н.,

Черкашина Анастасия Андреевна

Вариант 9

Цель:

- Совершенствование навыков разработки программ в среде программирования MS Visual C++
- Совершенствование навыков в программировании с использованием модульного подхода

Задание 1

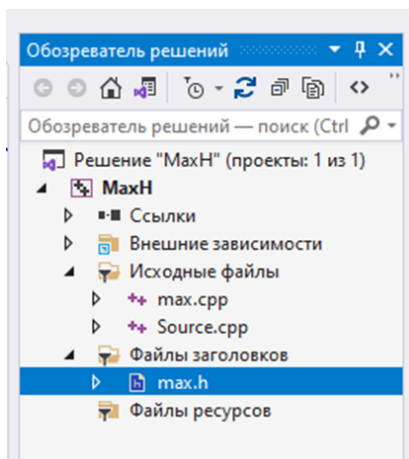
1 Условие:

При наличии прототипа вызываемые функции не обязаны размещаться в одном файле с вызывающей функцией. То есть, описание тел функций можно разместить в другом файле, который необходимо подключить к исходному с использованием директивы `#include`. Составить программу, содержащую обращение к функции вычисления максимума из двух чисел: Возможное решение данной задачи имеет вид:

1) Заголовочный файл `max.h`. Его текст:

```
#pragma once
int max(int a, int b);
```

Данный файл (`max.h`) добавляется в папку Файлы заголовков:



2) Файл `source.cpp` содержит исходный код проекта:

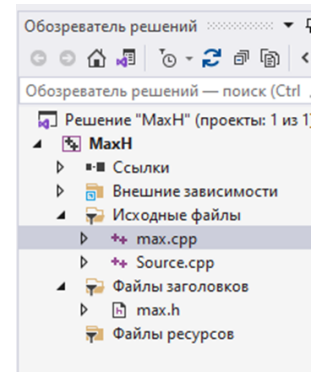
```
/* включение заголовочного файла max.h, содержащего
   прототип функции max */
#include <iostream>
#include "max.h"

using namespace std;

int main() {
    int x, y, z;
    cout << "\n поочередно введите x и y \n";
    cin >> x >> y;
    z = max(x, y);          /* вызов функции */
    cout << "z=" << z;
    return 0;
}
```

- 3) Текст слева от картинки. Этот текст будет расположен в одном столбце. Вы можете продолжить писать текст, и он будет оставаться в этом столбце. Например, описание функции или другие комментарии.

```
int max (int a, int b) {
    int c;           //
    c=(a>b)?a:b;      //   тело функции
    return c;         //
}
```



Вызов функции является выражением в правой части оператора присваивания:

```
z = max(x,y);
```

при выполнении которого значения аргументов x и y подставляются вместо параметров a и b соответственно (передача параметров в функцию по значению). После выполнения тела функции возвращаемое значение передается в место вызова функции и присваивается переменной z . Описание функции находится в файле `max.cpp`, поэтому для включения файла в программу необходимо в тексте программы указать директиву препроцессора `#include "max.h"`.

Описание функции может находиться в одном файле с главной программой. При этом директива `#include "max.h"` не указывается, а вместо нее помещается описание функции.

2 Код + Результат работы программы:

```
$ cat max.hpp
template <typename T>
T max(T, T);
$ cat max.cpp
#include <cstdlib>
#include "max.hpp"

template <> // template specialization
int32_t max<int32_t>(int32_t a, int32_t b) {
    return a > b ? a : b;
}
$ cat main.cpp
#include <cstdlib>
#include <iostream>

#include "max.hpp"

int32_t main(int32_t, const char**) {
    std::cout << "Последовательно введите два числа для сравнения: ";
    int32_t x, y;
    std::cin >> x >> y;
    std::cout << "Максимальное значение между этими числами: " << max(x, y) << '\n';
    return EXIT_SUCCESS;
}
$ echo $CXXFLAGS
-Wall -Wextra -Wconversion -Wshadow -Weffc++ -pedantic-errors -std=c++23
$ g++ $CXXFLAGS main.cpp max.cpp
$ ./a.out
Последовательно введите два числа для сравнения: 1 2
Максимальное значение между этими числами: 2
$ ./a.out
Последовательно введите два числа для сравнения: 2 3
Максимальное значение между этими числами: 3
```

src: max.hpp src: max.cpp src: main.cpp

Задание 2

В соответствии со своим вариантом лабораторной работы №16 Задания 2, оформить функции, реализующие основные алгоритмы работы с динамическими структурами данных, в виде модуля. Организовать вызов модуля программой.

3 Код:

- func.hpp

```
#include <list>
#include <string>
#include <iostream>

#pragma once

enum class PublicationType { Newspaper, Magazine };

using Publication = std::tuple<PublicationType, std::string, double>;

std::istream& operator>>(std::istream& is, PublicationType& type);
std::ostream& operator<<(std::ostream& out, PublicationType type);
std::ostream& operator<<(std::ostream& out, const std::list<Publication>& pubs);
void addPublication(std::list<Publication>& pubs, const Publication& pub);
bool loadPublications(std::list<Publication>& pubs, const std::string& filename);
bool savePublications(const std::list<Publication>& pubs, const std::string& filename);
```

- func.cpp

```
#include <sstream>
#include <fstream>
#include "func.hpp"

std::istream& operator>>(std::istream& is, PublicationType& type) {
    std::string str;
    is >> str;
    if (str == "Газета")
        type = PublicationType::Newspaper;
    else if (str == "Журнал")
        type = PublicationType::Magazine;
    else {
        std::cerr << "Ошибка: Неверный тип публикации: " << str << '\n';
        return is;
    }
    return is;
}

std::ostream& operator<<(std::ostream& out, PublicationType type) {
    return out << (type == PublicationType::Newspaper ? "Газета" : "Журнал");
}

std::ostream& operator<<(std::ostream& out, const std::list<Publication>& pubs) {
    if (pubs.empty()) {
        out << "Список пуст.\n";
        return out;
    }
    for (const auto& pub : pubs) {
        out << std::get<0>(pub) << ": " << std::get<1>(pub)
            << ", Цена: " << std::get<2>(pub) << " монеты.\n";
    }
}
```

```

    }
    return out;
}

void addPublication(std::list<Publication>& pubs, const Publication& pub) {
    if (std::get<0>(pub) == PublicationType::Magazine)
        pubs.push_front(pub);
    else
        pubs.push_back(pub);
}

bool loadPublications(std::list<Publication>& pubs, const std::string& filename) {
    std::ifstream file{filename};
    if (!file.is_open()) {
        std::cerr << "Ошибка: Не удалось открыть файл " << filename << '\n';
        return false;
    }

    std::string line, pubType, name;
    double price;
    while (std::getline(file >> std::ws, line)) {
        std::istringstream iss{line};
        std::getline(iss >> std::ws, pubType, ',');
        std::getline(iss >> std::ws, name, ',');
        if (!(iss >> price)) {
            std::cerr << "Ошибка: Некорректный формат строки: " << line << '\n';
            return false;
        }
        PublicationType type = pubType == "Журнал" ? PublicationType::Magazine : Publi
        addPublication(pubs, {type, name, price});
    }
    return true;
}

bool savePublications(const std::list<Publication>& pubs, const std::string& filename) {
    std::ofstream file{filename};
    if (!file.is_open()) {
        std::cerr << "Ошибка: Не удалось открыть файл для записи: " << filename << '\n';
        return false;
    }

    for (const auto& pub : pubs) {
        file << (std::get<0>(pub) == PublicationType::Newspaper ? "Газета" : "Журнал")
            << ", " << std::get<1>(pub) << ", " << std::get<2>(pub) << '\n';
    }
    return true;
}

```

- main.cpp

```

#include <cstdlib>
#include <cstdint>

```

```

#include <limits>
#include <iostream>

#include "func.hpp"

int main() {
    std::setlocale(LC_ALL, "ru_RU.UTF-8");
    std::list<Publication> pubs;
    uint32_t choice = 0;

    while (choice != 5) {
        std::cout << "\nМеню:\n"
            << "1. Добавить публикацию\n"
            << "2. Показать публикации\n"
            << "3. Загрузить из файла\n"
            << "4. Сохранить в файл\n"
            << "5. Выход\n"
            << "Ваш выбор: ";
        if (!(std::cin >> choice)) {
            std::cerr << "Ошибка: Некорректный ввод. Ожидается число.\n";
            std::cin.clear();
            std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
            continue;
        }

        switch (choice) {
            case 1: {
                PublicationType type;
                std::string name;
                double price;

                std::cout << "Введите тип (Газета или Журнал): ";
                if (!(std::cin >> type)) {
                    std::cerr << "Ошибка: Неверный тип публикации.\n";
                    std::cin.clear();
                    std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
                    break;
                }

                std::cout << "Введите название: ";
                std::getline(std::cin >> std::ws, name);

                std::cout << "Введите цену: ";
                if (!(std::cin >> price)) {
                    std::cerr << "Ошибка: Неверный ввод цены.\n";
                    std::cin.clear();
                    std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
                    break;
                }

                addPublication(pubs, {type, name, price});
                std::cout << "Публикация добавлена.\n";
            }
        }
    }
}

```

```

        break;
    }
    case 2:
        std::cout << pubs;
        break;

    case 3: {
        std::string filename;
        std::cout << "Введите имя файла: ";
        std::getline(std::cin >> std::ws, filename);

        if (loadPublications(pubs, filename))
            std::cout << "Публикации успешно загружены.\n";
        break;
    }

    case 4: {
        std::string filename;
        std::cout << "Введите имя файла: ";
        std::getline(std::cin >> std::ws, filename);

        if (savePublications(pubs, filename))
            std::cout << "Публикации успешно сохранены.\n";
        break;
    }

    case 5:
        std::cout << "Выход из программы.\n";
        break;

    default:
        std::cerr << "Ошибка: Неверный выбор.\n";
        break;
    }
}

return EXIT_SUCCESS;
}

```

src: func.hpp src: func.cpp src: main.cpp

4 Результат работы программы:

Меню:

1. Добавить публикацию
2. Показать публикации
3. Загрузить из файла
4. Сохранить в файл
5. Выход

Ваш выбор: 1

Введите тип (Газета или Журнал): Газета

Введите название: Новая

Введите цену: 100

Публикация добавлена.

Меню:

1. Добавить публикацию
2. Показать публикации
3. Загрузить из файла
4. Сохранить в файл
5. Выход

Ваш выбор: 1

Введите тип (Газета или Журнал): Журнал

Введите название: Зеленый

Введите цену: 102

Публикация добавлена.

Меню:

1. Добавить публикацию
2. Показать публикации
3. Загрузить из файла
4. Сохранить в файл
5. Выход

Ваш выбор: 1

Введите тип (Газета или Журнал): Газета

Введите название: Фиолетовая

Введите цену: 20

Публикация добавлена.

Меню:

1. Добавить публикацию
2. Показать публикации
3. Загрузить из файла
4. Сохранить в файл
5. Выход

Ваш выбор: 2

Журнал: Зеленый, Цена: 102 монеты.

Газета: Новая, Цена: 100 монеты.

Газета: Фиолетовая, Цена: 20 монеты.

Меню:

1. Добавить публикацию
2. Показать публикации
3. Загрузить из файла
4. Сохранить в файл
5. Выход

Ваш выбор: 5

Выход из программы.