

**федеральное государственное автономное образовательное  
учреждение высшего образования**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
(ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ)  
(Факультет информационных технологий)**

**(Институт Принтмедиа и информационных технологий)  
Кафедра Информатики и информационных технологий**

**направление подготовки  
09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № \_\_\_\_**

**Дисциплина:** Введение в программирование

**Тема:** Циклы и ветвления

**Выполнил(а): студент(ка) группы** 221-379

Паракин. Н. С.  
(Фамилия И.О.)

**Дата, подпись** 01.10.2022 [Подпись]  
(Дата) (Подпись)

**Проверил:** \_\_\_\_\_  
(Фамилия И.О., степень, звание) (Оценка)

**Дата, подпись** \_\_\_\_\_  
(Дата) (Подпись)

**Замечания:** \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Москва 2022**

## Лабораторная работа № 2

### Циклы и ветвления

**Цель:** Получить практические навыки работы с циклами «С++ и С» для создания программ.

#### Постановка задачи:

- Разработать функции выполняющие поставленные задачи, с использованием новых знаний.
- Оформить отчет
- Титульный лист
- Название и цель работы
- Постановка задачи
- Листинг исполняющего кода с комментариями
- Блок-схема программы

#### Задание №1

```
void x_pyramid()
{
    std::cout << "Задание #1\n";
    int number, const_number, x_count, space_print; // Инициализация переменных типа int
    std::cout << "Введите высоту пирамиды: "; // Вывож текста для получения ответа от пользователя
    bool allow_print;
    std::cin >> const_number; // Запрос на получение данных
    number = const_number - 1; // Получение количества " " на первой строке
    while (number >= 0)
    {
        x_count = const_number - number; // Вычисление количества X в строке
        space_print = 1; // Переменная для понятия сколько пробелов было напечатано
        allow_print = true; // Переменная отвечающая за написание пробелов и X на одной строке
        while (allow_print)
        {
            if (space_print <= number) // Пока количество пробелов меньше либо равно числу пробелов в строке
            {
                std::cout << ' '; // Печать пробела
                space_print += 1; // Добавление 1 к количеству напечатанных пробелов
            }
            else if (const_number >= space_print) // Пока количество символов в строке вообще больше количества пробелов
            {
                std::cout << "X"; // Выводим X
                space_print += 1; // Добавляем 1 к переменной отвечающей за количество написанных символов
            }
            else // В случае если предыдущие условия не могут быть выполнены
            {
                allow_print = false; // Запрещает продолжение внутреннего цикла
                std::cout << '\n'; // Осуществляет переход на новую строку
            }
        }
        number --; // Отнимает от количества пробелов 1
    }
}
```

## Задание №2

```
void factorial_finder()
{
    std::cout << "Задание #2\n";
    int factorial_number, number, iterations = 0;
    unsigned long long int factorial;
    bool count = true;
    while (count)
    {
        std::cout << "Укажите число для поиска факториала: ";
        std::cin >> number;

        while (number != 0 or number < 0)
        {
            if (iterations == 0 and number != 0 or number < 0)
            {
                factorial_number = number;
                factorial = factorial_number * (factorial_number - 1);
                factorial_number -= 2;
                iterations += 1;
            }
            else if (iterations != 0 and factorial_number >= 2)
            {
                factorial = factorial * factorial_number;
                factorial_number -= 1;
                iterations += 1;
            }
            else
            {
                std::cout << factorial << std::endl;
                iterations = 0;
                factorial = 0;
                break;
            }
        }
        if (number == 0)
        {
            std::cout << "Цикл завершен!!";
            count = false;
        }
    }
}
```

## Задание №3

```
void bank_offer()
{
    std::cout << "Задание #3\n";
    double money, precent;
    int years = 1, const_years;
    std::cout << "Пожалуйста введите сумму стартового капитала, " <<
        "длительность вклада и процентную ставку\n";
    std::cin >> money;
    std::cin >> const_years;
    std::cin >> precent;
    while (years <= const_years)
    {
        money = money + money * (precent / 100);
        std::cout << "В конце " << years << "го года у вас будет " << money << std::endl;
        years += 1;
    }
    std::cout << "Через " << const_years << " лет вы получите " << money << std::endl;
}
```

## Задание №4

```
void bank_mortgage()
{
    std::cout << "Задание #4\n";
    long double amount, initial_payment, yearly_payment, precent, years;
    std::cout << "Введите стоимость квартиры: ";
    std::cin >> amount;
    std::cout << "Введите сумму начального взноса: ";
    std::cin >> initial_payment;
    std::cout << "Введите срок выплат: ";
    std::cin >> years;
    std::cout << "Введите процентную ставку: ";
    std::cin >> precent;

    amount = amount - initial_payment;
    yearly_payment = amount / years;
    yearly_payment = yearly_payment + (yearly_payment * (precent / 100));

    std::cout << "Ежегодные выплаты составляют: " << yearly_payment << std::endl;
    std::cout << "Общая сумма выплат: " << yearly_payment * years << "\n";
}
```

# Задание №5

```
void old_pounds_summ()
{
    std::cout << "Задание #5\n";
    int old_pound, shilling, penny, old_pound2, shilling2, penny2, old_pound3, shilling3, penny3;
    char next = 'y';
    while (next != 'n')
    {
        std::cout << "Введите первое число: ";
        std::cin >> old_pound >> shilling >> penny;
        std::cout << "Введите второе число: ";
        std::cin >> old_pound2 >> shilling2 >> penny2;
        penny3 = penny + penny2;
        shilling3 = shilling + shilling2;
        old_pound3 = old_pound + old_pound2;

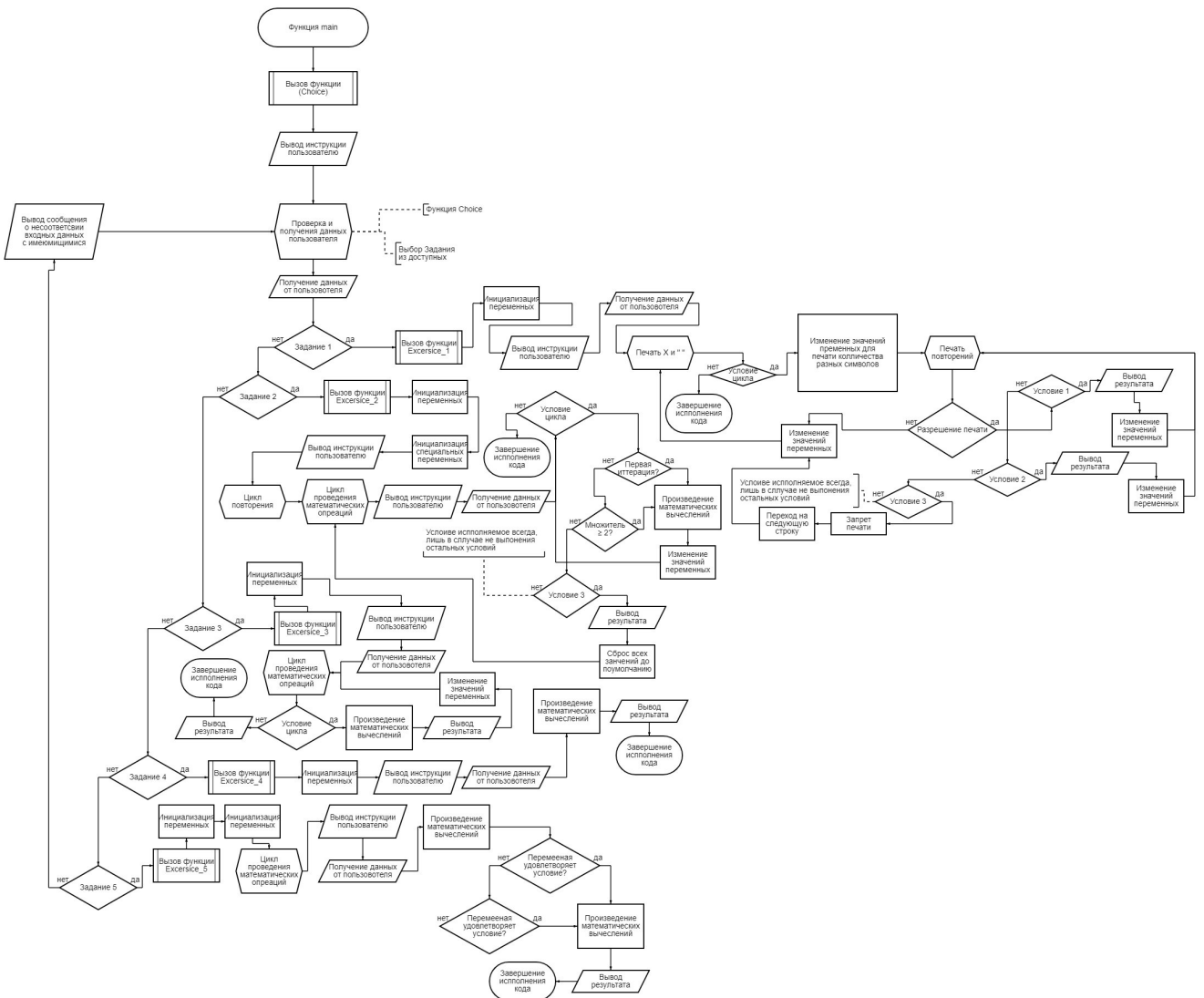
        if (penny3 >= 12)
        {
            penny3 = penny3 - 12;
            shilling3 += 1;
        }

        if (shilling3 >= 20)
        {
            shilling3 = shilling3 - 20;
            old_pound3 += 1;
        }

        std::cout << old_pound3 << " " << shilling3 << " " << penny3 << std::endl;

        std::cout << "Продолжить (y/n): ";
        std::cin >> next;
    }
}
```

# Блок-схема





## Остальные функции

```
void choice()
{
    int x;
    bool loop = true;
    while (loop)
    {
        std::cout << "Введите номер задания от 1-5: ";
        std::cin >> x;
        if (x == 1)
        {
            x_pyramid();
            loop = false;
        }
        else if (x == 2)
        {
            factorial_finder();
            loop = false;
        }
        else if (x == 3)
        {
            bank_offer();
            loop = false;
        }
        else if (x == 4)
        {
            bank_mortgage();
            loop = false;
        }
        else if (x == 5)
        {
            old_pounds_summ();
            loop = false;
        }
        else
        {
            std::cout << "Выбранного задания не существует. Выберите другое!\n";
        }
    }
}

int main()
{
    setlocale(LC_ALL, "Russian");
    choice();
}
```