

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных
систем

Лабораторная работа №1

по дисциплине: Объектно-ориентированное программирование

тема: *«Знакомство с интегрированной средой разработки (ИСР) Microsoft
Visual Studio 2013 или QT»*

Выполнил: ст. группы ПВ-202
Нехаенко Владимир Сергеевич

Проверил:
Картамышев Сергей Владимирович

Белгород 2022 г.

Лабораторная работа №1

Знакомство с интегрированной средой разработки (ИСР) Microsoft Visual Studio 2013 или QT

Цель работы: изучение функциональных возможностей интегрированной среды разработки (ИСР) Visual Studio 2013 или QT

Задание:

1. Разработать и отладить консольное приложение в соответствии с вариантом задания (см.табл.1).
2. Разработать экранную форму, аналогичную по своему внешнему виду форме, заданной в варианте задания (см. табл.2). Обработчики событий экранной формы не реализовывать.

Вариант 18:

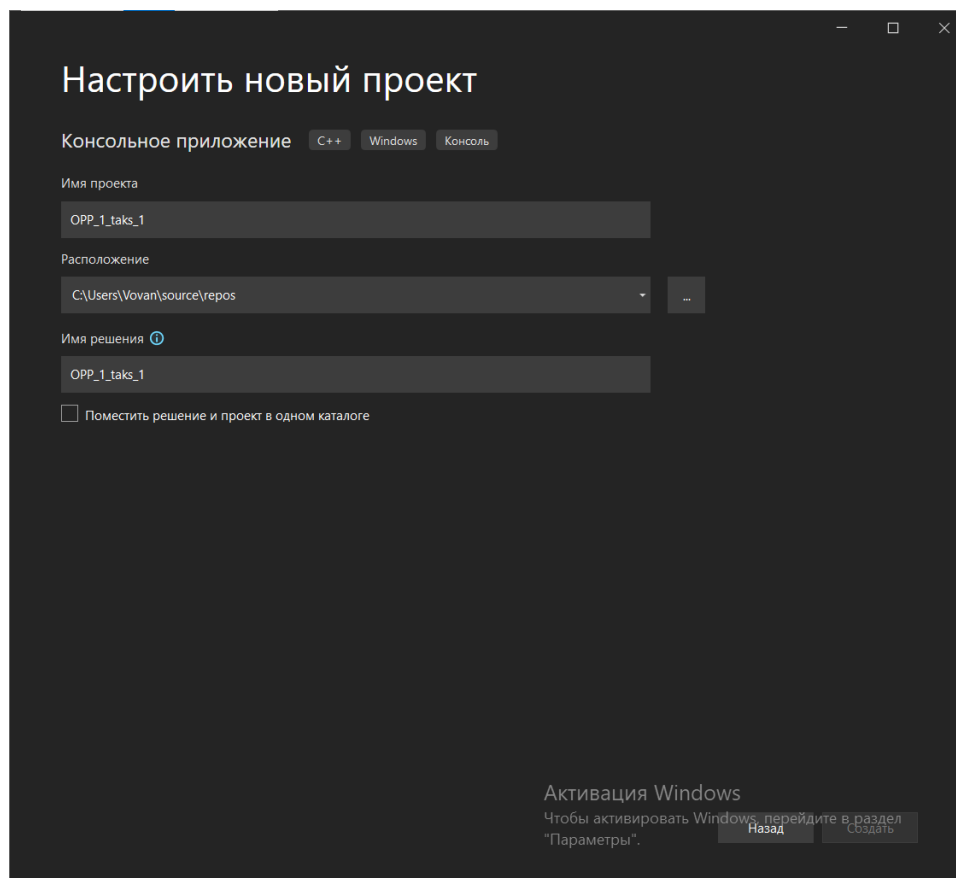
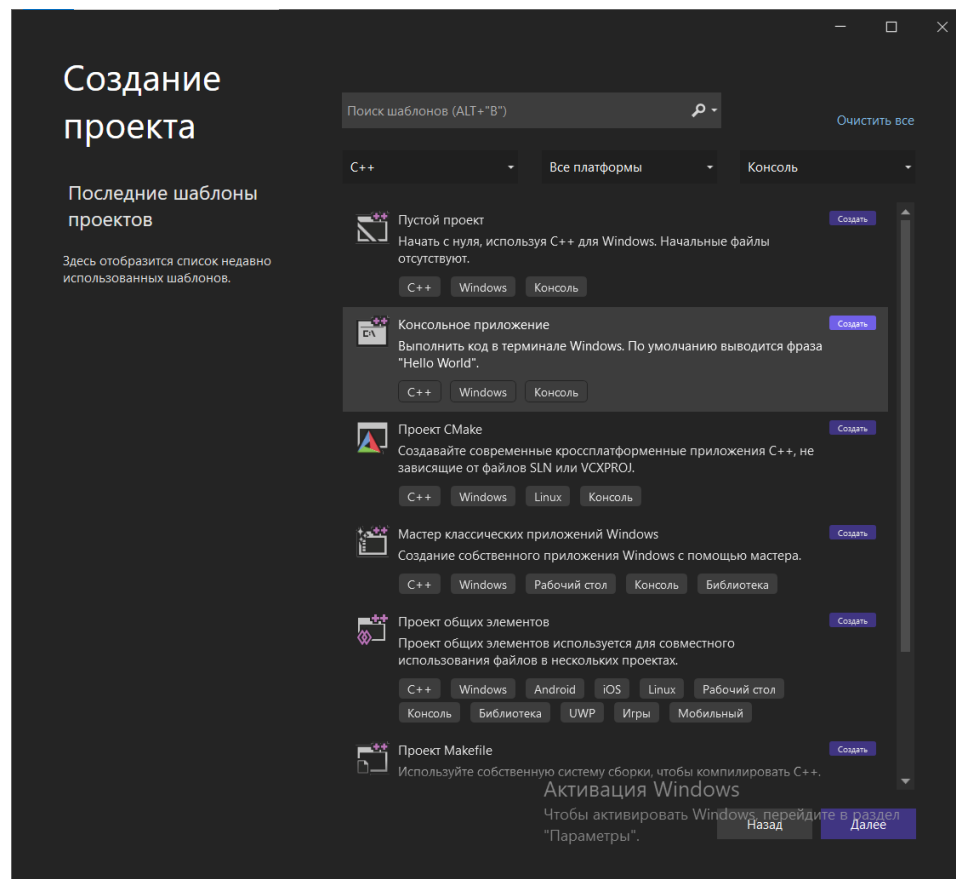
Задание 1:

- а. Дан текстовый файл. Вывести все согласные, которых нет ни в одном слове.
- б. Дан файл, компонентами которого являются n-мерные векторы ($n = \text{const}$). Векторы с наибольшим модулем перенести в конец файла

Задание 2:

Форма «Разметка» (контекстное меню рисунка – «размер и положение») MS Word, вкладки «Размер» и «Положение»

1. Пошаговое описание способа создания консольного приложения в Visual Studio



2. Исходный код задания 1.

```
#include <iostream>
#include <fstream>
#include <set>

using namespace std;

int main() {
    set<char> consonants = {'b', 'c', 'd', 'f', 'g', 'h', 'j', 'k', 'l',
                           'm', 'n', 'p', 'q', 'r', 's', 't', 'v', 'w', 'x',
                           'z'};

    string filename;
    cout << "Input file name:";
    cin >> filename;

    ifstream f(filename);

    if (f.is_open()) {
        while (!f.eof()) {
            char c;

            f >> c;

            if (isalpha(c))
                consonants.erase(tolower(c));
        }
        f.close();
    } else {
        cout << "Opening file error!";
        return 1;
    }

    cout << "Consonants that do not occur in any word: ";
    for (auto &letter : consonants) {
        cout << letter << " ";
    }
}
```

Исходный код задания 2

Файл “myVector.h”

```
#ifndef TESTCPP_MYVECTOR_H
#define TESTCPP_MYVECTOR_H

#include <fstream>
#include <cmath>

const size_t N = 3; // размер вектора

typedef double baseType; // Базовый тип вектора
typedef baseType vectorType[N]; // Тип - "вектор"

// Возвращает модуль вектора v
baseType getModulus(vectorType v){
    baseType sum = 0;

    for (size_t i = 0; i < N; ++i)
        sum += pow(v[i], 2);

    return sqrt(sum);
}
```

```

// Считывает вектор v с клавиатуры
void inputVector(vectorType v){
    for (size_t i = 0; i < N; ++i)
        std::cin >> v[i];
}

// Выводит вектор v на экран
void printVector(vectorType v){
    for (size_t i = 0; i < N; ++i)
        std::cout << v[i] << " ";

    std::cout << std::endl;
}

// Меняет местами векторы на позициях pos1 и pos2 в файле f
void swapVectorsInFile(size_t pos1, size_t pos2, std::fstream &f){
    vectorType temp1;
    vectorType temp2;

    // Чтение вектора с первой позиции
    f.seekg(pos1, std::ios::beg);
    f.read((char *)&temp1, sizeof temp1);

    // Чтение вектора со второй позиции
    f.seekg(pos2, std::ios::beg);
    f.read((char *)&temp2, sizeof temp2);

    // Запись первого вектора на место второго
    f.seekg(pos2, std::ios::beg);
    f.write((char *)&temp1, sizeof temp1);

    // Запись второго вектора на место первого
    f.seekg(pos1, std::ios::beg);
    f.write((char *)&temp2, sizeof temp2);
}

#endif //TESTCPP_MYVECTOR_H

```

Программа для создания бинарного файла с векторами:

```

#include <iostream>
#include "myVector.h"

using namespace std;

int main(){

    string filename;
    cout << "Input file name:";
    cin >> filename;

    size_t vectorsCnt;
    cout << "Input count of vectors:";
    cin >> vectorsCnt;

    ofstream out (filename, ios::binary|ios::out);

    cout << "Input vectors" << endl;
    for (size_t i = 0; i < vectorsCnt; ++i) {
        vectorType v;

```

```

        inputVector(v);
        out.write((char*)&v, sizeof v);
    }
    out.close();

    cout << "Vectors successfully written.";
}

```

Файл основной программы:

```

#include <iostream>
#include "myVector.h"

using namespace std;

struct afterProcessingFileInfo {
    baseType maxModulus = 0;
    size_t fileLen = 0;
};

// Выводит на экран все векторы из файла filename
void printVectorsInFile(const string &filename) {
    ifstream in(filename, ios::in | ios::binary);

    vectorType v;

    while (in.read((char *) &v, sizeof v))
        printVector(v);

    in.close();
}

/* Обработывает файл filename, возвращает структуру, которая хранит кол-во
 * векторов в файле и наибольший модуль вектора */
afterProcessingFileInfo findMaxModulusAndFileLen(const string& filename) {
    afterProcessingFileInfo res;

    ifstream f (filename, ios::binary|ios::in);
    vectorType v;

    while (f.read((char *) &v, sizeof v)) {
        res.fileLen++;

        baseType modulus = getModulus(v);
        if (modulus > res.maxModulus)
            res.maxModulus = modulus;
    }
    f.close();

    return res;
}

// Считывает вектор v, с позиции pos, из файла f
void readVectorFromPos(vectorType *v, size_t pos, fstream &f){
    f.seekg(pos, ios_base::beg);
    f.read((char *)v, sizeof (*v));
    f.seekg(pos, ios_base::beg);
}

// Перемещает векторы с максимальным модулем в файле filename в конец файла
void moveMaxModulusVectorsToEndFile(const string &filename) {

    afterProcessingFileInfo data = findMaxModulusAndFileLen(filename);

    fstream f(filename, fstream::in | fstream::out | fstream::binary);

```

```

if (!f.is_open())
    throw runtime_error("error opening file!");

vectorType v;

// позиция первого вектора с конца файла с не максимальным модулем
size_t notMaxPos = (data.fileLen - 1) * sizeof(v);
// позиция первого вектора с начала файла с максимальным модулем
size_t maxPos = 0;

while (true){

    // Поиск первого вектора с максимальным модулем начиная с позиции maxPos
    findFirstMax:
    readVectorFromPos(&v, maxPos, f);
    if (getModulus(v) != data.maxModulus){
        maxPos += sizeof v;
        goto findFirstMax;
    }

    // Поиск первого вектора с не максимальным модулем начиная с позиции notMaxPos
    findFirstNotMax:
    readVectorFromPos(&v, notMaxPos, f);
    if (getModulus(v) == data.maxModulus){
        notMaxPos -= sizeof v;
        goto findFirstNotMax;
    }

    // Все векторы с максимальным модулем уже в конце
    if (maxPos >= notMaxPos)
        break;

    swapVectorsInFile(maxPos, notMaxPos, f);
}

f.close();
}

int main() {
    string filename;
    cout << "Input name of file with vectors:";
    cin >> filename;

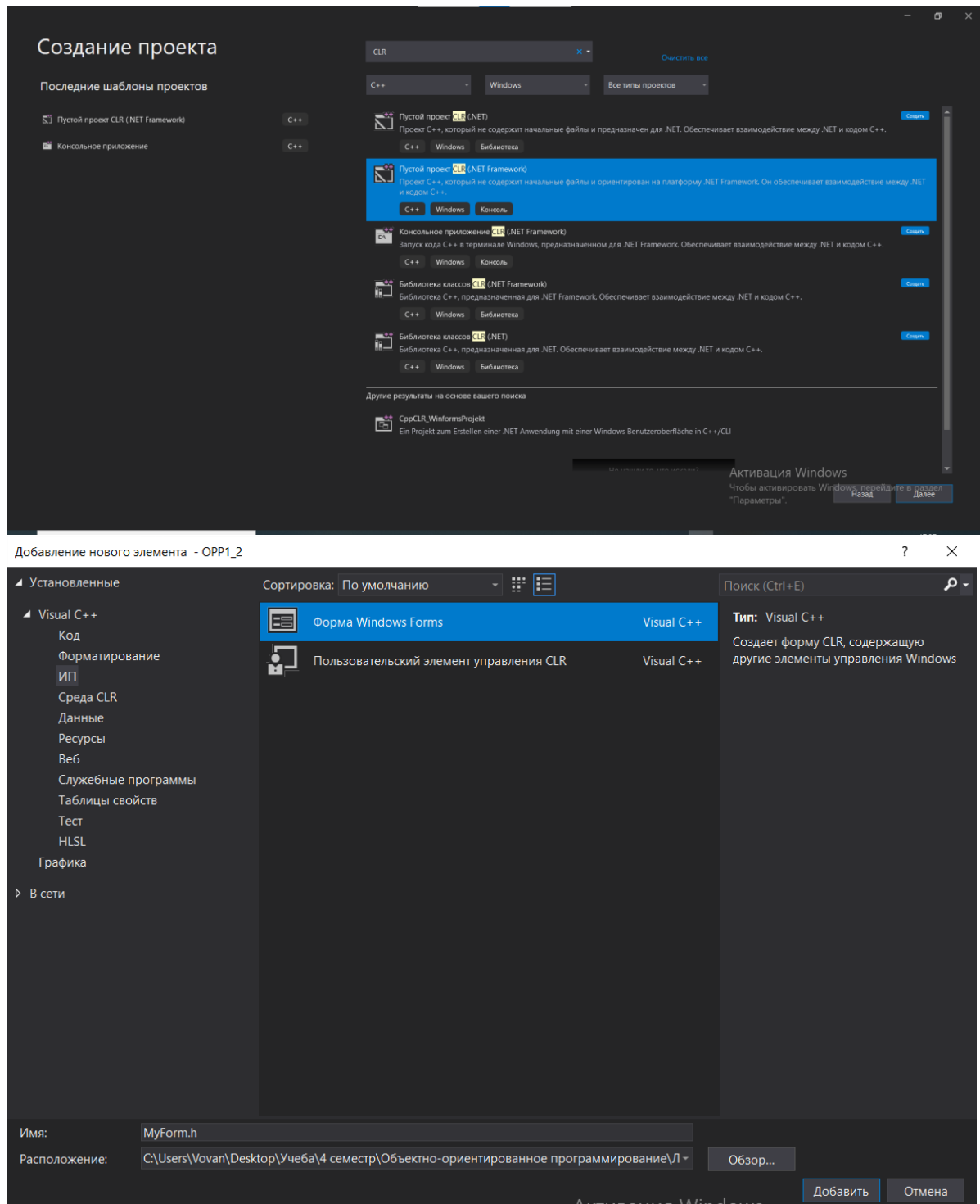
    cout << "Content of file before processing:" << endl;
    printVectorsInFile(filename);

    moveMaxModulusVectorsToEndFile(filename);

    cout << "Content of file after processing:" << endl;
    printVectorsInFile(filename);
}

```

Задание 2:



Необходимо добавить следующие настройки для корректной работы проекта

Страницы свойств OPP1_2

Конфигурация: Активная (Debug) Платформа: Активная (Win32) Диспетчер конфигураций...

Свойства конфигурации

Общие

Дополнительно

Отладка

Каталоги VC++

C/C++

Компоновщик

Общие

Ввод

Файл манифеста

Отладка

Система

Оптимизация

Внедренный IDL

Метаданные Window

Дополнительно

Все параметры

Командная строка

Инструмент манифеста

Генератор XML-документов

Информация об исходных файлах

События сборки

Настраиваемый этап сборки

Подсистема

Windows (/SUBSYSTEM:WINDOWS)

Минимальная требуемая версия	
Резервируемый размер кучи	
Фиксируемый размер кучи	
Резервируемый размер стека	
Фиксируемый размер стека	
Включить большие адреса	
Сервер терминалов	
Запускать с компакт-диска с помощью	Нет
Запускать из сети с помощью файла	Нет
Драйвер	Не задано

Подсистема

Параметр /SUBSYSTEM предписывает операционной системе порядок выполнения EXE-файла. Выбор подсистемы влияет на символ точки входа (или функцию точки входа), который выберет ...

ОКОтменаПрименить

Страницы свойств OPP1_2

Конфигурация: Активная (Debug) Платформа: Активная (Win32) Диспетчер конфигураций...

Свойства конфигурации

Общие

Дополнительно

Отладка

Каталоги VC++

C/C++

Компоновщик

Общие

Ввод

Файл манифеста

Отладка

Система

Оптимизация

Внедренный IDL

Метаданные Window

Дополнительно

Все параметры

Командная строка

Инструмент манифеста

Генератор XML-документов

Информация об исходных файлах

События сборки

Настраиваемый этап сборки

Точка входа

main

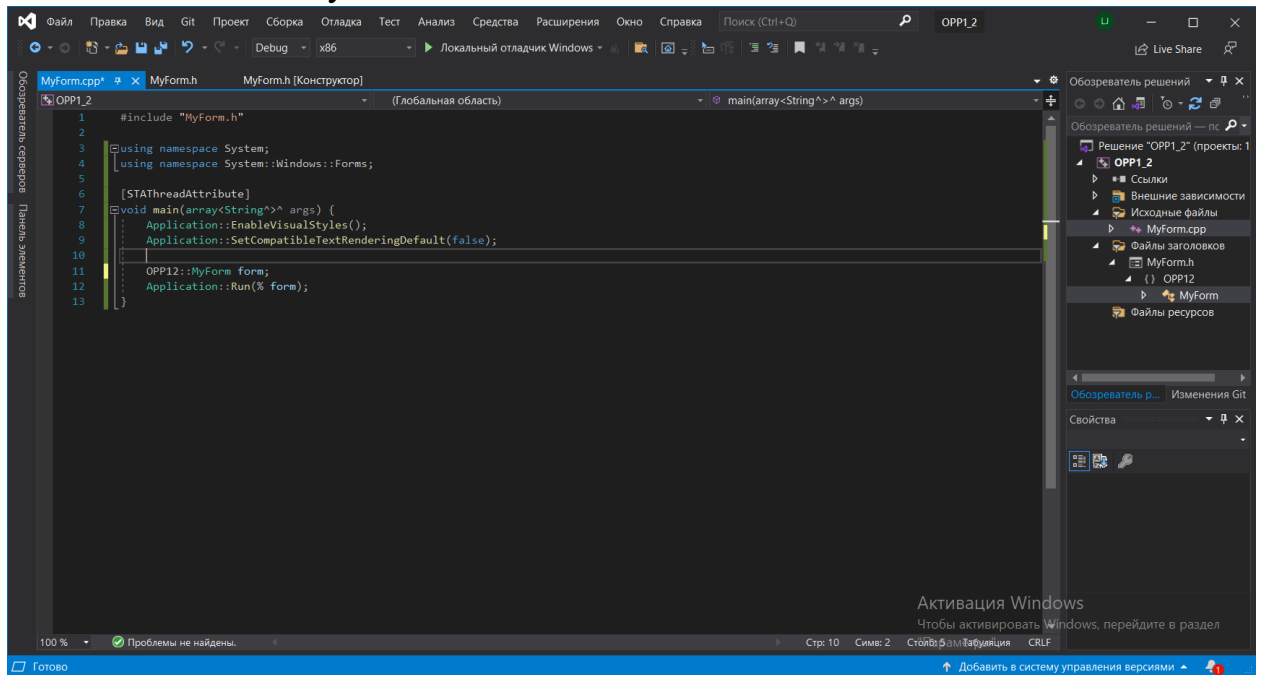
Без точки входа	Нет
Установить контрольную сумму	Нет
Базовый адрес	
Внесение случайности в базовый а	Да (/DYNAMICBASE)
Фиксированный базовый адрес	Нет (/FIXED:NO)
Предотвращение исполнения данн	Да (/NXCOMPAT)
Отключить создание сборки	Нет
Выгрузить отложено загружаемые [
Не включать отложено загружаемы	
Библиотека импорта	
Объединить разделы	
Конечный компьютер	MachineX86 (/MACHINE:X86)
Профиль	Нет
Атрибут потока CLR	
Тип CLR-образа	Тип образа по умолчанию
Файл ключа	
Контейнер ключей	
Отложенная подпись	

Точка входа

Параметр /ENTRY задает функцию точки входа в качестве начального адреса для EXE-файла или DLL.

ОКОтменаПрименить

Затем вставить следующий код.



Форма для повторения:

Макет ? X

Положение Обтекание текстом **Размер**

Высота

☒ абсолютная 11,58 см

☐ относительная относительно Страницы

Ширина

☒ абсолютная 12,65 см

☐ относительная относительно Страницы

Поворот

Доворот: 0°

Масштаб

Высота: 42 % Ширина: 42 %

☒ сохранить пропорции

☒ относительно исходного размера

Исходный размер

Высота: 27,26 см Ширина: 29,77 см

Сброс

OK Отмена

Макет ? X

Положение Обтекание текстом Размер

По горизонтали

☐ выравнивание По левому краю относительно Колонны

☐ в книге Внутри относительно Поля

☒ положение 0,01 см правее Колонны

☐ относительное положение относительно Страницы

По вертикали

☐ выравнивание По верхнему краю относительно Страницы

☒ положение 0,01 см ниже Абзаца

☐ относительное положение относительно Страницы

Параметры

☒ перемещать вместе с текстом ☒ разрешить перекрытие

☐ установить привязку ☒ разметка в ячейке таблицы

OK Отмена

Результат:

Макет

Положение Обтекание текстом **Размер**

Высота

☒ абсолютная 0

☐ относительная 0 относительно Страницы

Ширина

☒ абсолютная 0

☐ относительная 0 относительно Страницы

Поворот

Поворот 0

Масштаб

Высота 0 Ширина 0

☒ сохранить пропорции

☒ относительно исходного размера

Исходный размер

Высота: 27,26 см Ширина: 29,77 см

Сброс

Ок Отмена

Макет

Положение Обтекание текстом **Размер**

По горизонтали

☐ выравнивание По левому краю относительно Колонны

☐ в книге Внутри относительно Поля

☒ положение 0 правее Колонны

☐ относительное положение 0 относительно Страницы

По вертикали

☐ выравнивание По левому краю относительно Страницы

☒ положение 0 ниже Абзаца

☐ относительное положение 0 относительно Страницы

Параметры

☒ перемещать вместе с текстом ☒ разрешить перекрытие

☐ установить привязку ☒ разметка в ячейке таблицы

Ок Отмена

Исходный код формы:
См. репозиторий на [GitHub](#)