# САОД. Задание 1

## Темы

* Входной тест
* Знакомство с C++, особенностями Visual Studio.
* Простейший ввод-вывод.
* Функции и дополнительные модули в проекте (cpp и h).

## Задания

1. Пройти тест [https://lms.uni-dubna.ru/my/](https://sdo2.uni-dubna.ru/my/) в “Структуры и алгоритмы обработки данных”. Входной тест… (не оценивается в баллах, просто для справки …).
2. Запустить Visual Studio (*описано для VS 2019*) и “Создать новый проект”:
   1. Выбираем шаблон “Console App| C++”
   2. Называем проект, например, “Hello”.
   3. Выбираем для сохранения Рабочий стол.
   4. Изменяем имя решения на Ex1
   5. Нажимаем “Создать”.
3. Для знакомства со средой:
   1. В cpp-файл после   
      #include <iostream>   
      добавить:

using namespace std;

* 1. В main убрать std:: перед cout  
     cout << “Hellow World!” << endl;   
       
     и выполнить проект (Ctrl + F5).

Приложение будет выполнено, окно консоли останется открытым и будет ожидать нажатия любой клавиши.

* 1. Заценить объем на диске. Объем занимает более 40 Мб на диске.  
       
     Это происходит за счет большого объема вспомогательных файлов. Поэтому проект **должен быть на локальном диске** (не на флеш или в сети). Иначе построение будет занимать слишком много времени…

1. Учимся сохранять и продолжать проект.
   1. Выйти из студии.
   2. Удалить скрытый каталог .vs и каталог Debug.
   3. Архивировать каталог с решением через контекстное меню “Отправить->Сжатая (zip) папка”. Архив займет всего несколько килобайт.
   4. Удалить исходный каталог проекта.

Достигнута промежуточная цель – проект архивирован. Результат можно отправить на сетевой диск или на флешку.

* 1. Восстановить каталог Ex1 из архива.
  2. Открыть файл с расширением sln.
  3. Запустить проект Ctrl+F5.

Убедились, что смогли восстановить проект из нашего архива и что он работоспособен.

1. Для знакомства с вводом-выводом и студией.
   1. Добавить в Решение новый проект. На имени Решения (Ex1) из контекстного меню “Добавить новый проект” “Console App| C++”.

Назвать проект можно “in-out”.

* 1. Уберите из main строку std::cout<< … Попробуйте запустить новый проект (Ctrl+F5). Выполнится старый Hello.
  2. Для запуска нового проекта на его имени через контекстное меню “Назначить запускаемым”. После этого Ctrl+F5 запустит именно назначенный.

Несколько проектов в одном решении иметь удобно. Каждый из них прост, а имена говорят о содержании. Частично текст можно будет брать из предыдущих проектов.

* 1. Добавить строку   
     using namespace std;

для упрощения текста программы.

* 1. Добавить в main строчки и выполнить проект:  
     int x, y;

cout << "x = ? "; cin >> x; // Подсказка и ввод х

cout << "y = ? "; cin >> y; // Подсказка и ввод y

cout << x << " + " << y << " = " << x + y << endl;

* 1. Проверить, что работает (или добиться, чтобы работало ☺).
  2. Рассмотрите текст и поймите его. В итоге мы испробовали консольный вывод cout << и ввод cin >>.

1. Доработать упражнение, копируя последнюю строку и применяя ее для операций -,\*,/,%,|,&,^, <<, >> и любых других…

Научились создавать проект, объединять в решении несколько проектов. Подключать консольный ввод-вывод.

**Изучаем проект с несколькими cpp- и h-файлами.**

1. Добавить новый проект и назвать его “modules”.
   1. В main   
      int x = 3, y = 5;  
      cout<< x+y << endl;
   2. Проверить, что проект компилируется и выполняется (Ctrl+F5).
   3. После main добавить функцию: int Plus(int a, int b) {return a + b;}
   4. Вызывать ее в main   
      cout<< Plus (x, y) << endl;

*Компилятор “ругается”, не знает про Plus.*

* 1. Перенести Plus выше main и снова выполнить проект. Должен успешно выполниться.
  2. Вернуть Plus на старое место (после main), а перед main поместить прототип:  
     int Plus(int a, int b);

**Прототип** – заголовок функции без ее реализации.

* 1. Убедиться, что все работает.

Компилятору достаточно знать прототип, чтобы скомпилировать вызов функции.

1. Выносим функцию в отдельный модуль (*модуль компиляции в C++ состоит из заголовочного h-файла и реализации – cpp-файла*).
   1. Добавляем utils.cpp и utils.h. На имени проекта в контекстном меню “Добавить файл”.
   2. В utils.cpp перенести функцию.
   3. В utils.h перенести прототип.
   4. В модуле где main добавляем ниже #include <iostream> строку   
      #include "utils.h"
   5. Выполняем проект, убеждаемся, что работает.
2. Выходим из VS, удаляем ненужные файлы, архивируем решение и сохраняем на сетевом диске или на флешке.

Научились:

* создавать консольные приложения и подключать ввод-вывод.
* Пользоваться вводом-выводом;
* Добавлять новые модули в проект (utils) и размещать в них части программы (прототип в h, а реализацию в cpp);
* Знаем какие файлы из рабочего каталога не нужны, т.к. носят временный характер.

# Отчет

Подготовьте в Word краткий (*не более 1 стр.*) отчет о выполнении задания.

В отчете опишите: как в одном решении создать несколько проектов, как выполняется в C++ ввод-вывод на консоль (*какие используются объекты и операции*), из чего состоит модуль компиляции в C++, что содержится в заголовочном файле модуля, а что в реализации.

Отчет и архив решения поместите как ответ на задание.