# Директивы препроцессора

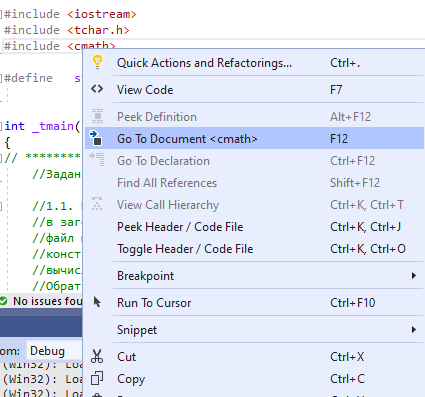
## Задание 1.1. Использование "математических" макросов стандартной библиотеки:

в заголовочном файле <cmath> (который в свою очередь включает старый файл math.h) определены часто используемые математические константы, например M\_PI. Воспользуйтесь этой константой для вычисления длины окружности.

Обратите внимание: определения констант в свою очередь тоже заключен в директивы условной трансляции. Что еще, кроме подключения заголовочного файла, требуется сделать?

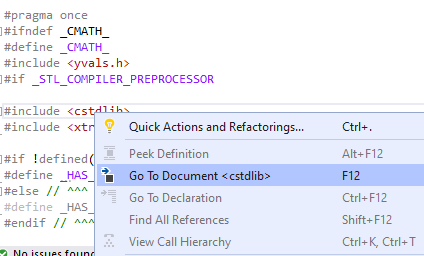
Подсказки:

1. Для того, чтобы увидеть содержимое файла <cmath>, можно по вызвать контекстное меню по правой кнопке мыши. И выбрать команду «Go To Document <cmath>».

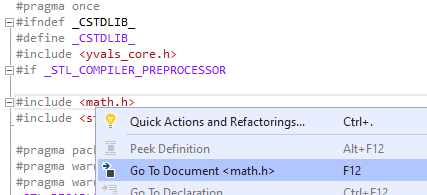
Icon

Description automatically generated

При этом откроется файл <cmath>:

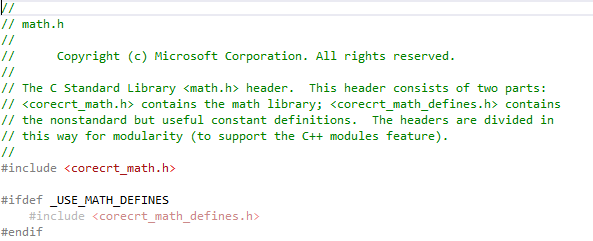


Но до файла <math.h> мы пока не добрались. Далее надо открыть файл <cstdlib> , а для этого выбрать команду «Go To Document < cstdlib >».

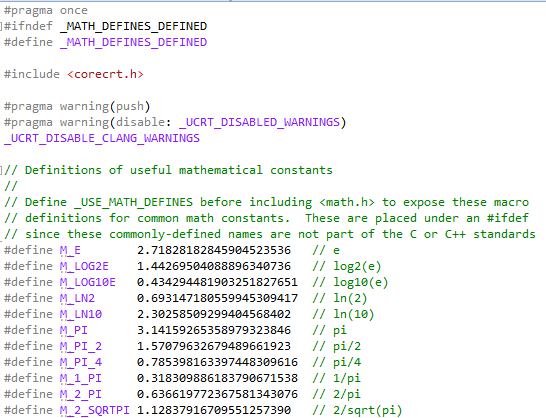




Теперь можно открыть файл <math.h> , а для этого выбрать команду «Go To Document <math.h>».



Нужная нам константа находится в файле <corecrt\_math\_defines.h>, но сам файл не подключен, т.к. препроцессору не известно имя **\_USE\_MATH\_DEFINES.**



1. Можно сделать имя **\_USE\_MATH\_DEFINES**  известным для препроцессора двумя способами:

* С помощью макроса в коде своей программы
* В настройках проекта

### **Первый** вариант.

Предположим, имеем такой код:

#include <iostream>

#include <tchar.h>

#include <cmath>

Если объявить макрос ДО #include <cmath>, но ПОСЛЕ #include <iostream>, то желаемого эффекта не удастся добиться. Это происходит потому, что при подключении iostream на одном из уровней вложенности уже происходит подключение <math>.

Graphical user interface, text, application

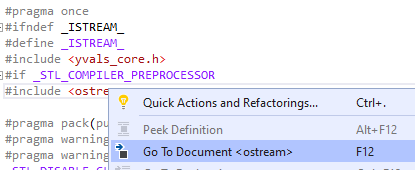
Description automatically generated

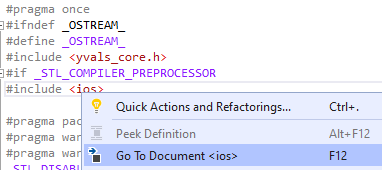
Graphical user interface, text, application

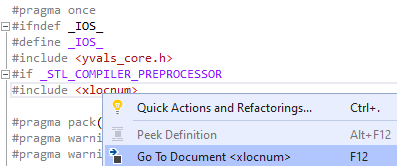
Description automatically generated

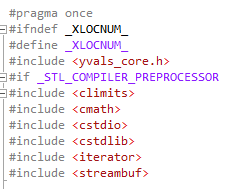
Graphical user interface, text, application

Description automatically generated









Поэтому, определять макрос следует ДО включения <iostream>. А еще лучше переставить строки кода так:

#include <cmath>

#include <iostream>

#include <tchar.h>

и определить макрос ДО включения <cmath>.

### **Второй** вариант.

В этом варианте настройка препроцессора осуществляется не в коде, а с помощью ключей компилятора.

Откройте окно **Property Pages.**  Это можно сделать выбрав команду “**Properties**” из контекстного меню проекта или выбрав команду “Properties” на вкладке “Project”.

Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application, email

Description automatically generated

При этом будет открыто окно Property Pages. Нас будет интересовать раздел

«C/C++» -> «Preprocessor» - > «Preprocessor Definitions».

Graphical user interface, text, application, email

Description automatically generated

В этом разделе можно увидеть, какие имена известны препроцессоруи , кроме того, добавить имя **\_USE\_MATH\_DEFINES**.

Graphical user interface, text, application, email

Description automatically generated

А после нажатия кнопки «Применить» и выбора раздела «C/C++» -> «Command Line» можно увидеть /D “\_USE\_MATH\_DEFINES” в настройках командной строки компилятора.

Graphical user interface, text, application, email

Description automatically generated

## Задание 1.4.

Загляните в Project-> Properties -> диалоговая панель “Property Pages” -> “C\C++” -> “Preprocessor” -> “Preprocessor definitions”.

В опциях командной строки в зависимости от мишени сборки проекта (Debug или Release) может быть определено с помощью ключа /D имя **\_DEBUG.**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Если изменить конфигурацию сборки, то получим:

Graphical user interface, text, email

Description automatically generated

Graphical user interface, text

Description automatically generated

## Задание 1.5

Управляем определенностью идентификатора **\_UNICODE.** У

Это необходимо делать НЕ в коде, а используя настройки проекта, т.к. было бы не корректно в одном файле использовать кодировку SBCS, а в другом - UNICODE.

Настройка осуществляется в разделе «Configuration Properties» -> «Advanced» -> « Character Set».

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

## Задание 1.6

Создаем файл предкомпиляции.

Предкомпиляция заголовочных файлов используется для уменьшения времени получения загрузочного модуля, если используется большое количество кода, который **не изменяется** (системныезаголовочные файлы - <windows.h>, библиотечные – <tchar.h>), а также, если во многих файлах-клиентах подключаются одни и те же пользовательские интерфейсы (”my.h”), которые **изменяются редко**.

В случае нашей лабораторной работы можно поместить строки, включающие библиотечные файлы в отдельный заголовочный файл и создать файл предкомпиляции.

По умолчанию при создании нового проекта в Visual Studio в проекте указывается имя *файл предкомпилированного заголовка -* stdafx.h ( VS 2019*)*. Имя этого файла может быть изменено.

Graphical user interface, text, email

Description automatically generated

Надо выполнить несколько шагов:

1. Создать файл stdafx.h и поместить в него заголовки библиотечных файлов, которые будут использованы в работе, например:

#pragma once

#define \_USE\_MATH\_DEFINES

#include <iostream>

#include <tchar.h>

#include <cmath>

1. Создать файл реализации stdafx.cpp и в него поместить единственную строку

#include "stdafx.h"

1. В клиентском файле с исходным кодом подключить заголовочный файл stdafx.h
2. Задать ключи для файлов реализации:

для stdafx.cpp -/Yc

для lab3.cpp -/Yu

Graphical user interface, text, email

Description automatically generatedGraphical user interface, text, email

Description automatically generated

В результате будет сформирован файл предкомпиляции **<имя\_проекта>.pch**.

