

Ravesli [Ravesli](#)

- [Уроки по C++](#)
- [OpenGL](#)
- [SFML](#)
- [Qt5](#)
- [RegExr](#)
- [Ассемблер](#)
- [Купить .PDF](#)

Конфигурация компилятора: Уровни предупреждений и ошибки

👤 [Юрий](#) |

- [Уроки C++](#)

|

📅 Обновл. 17 Сен 2020 |

👁 9496

1  2

На этом уроке мы поговорим о том, как повысить уровень предупреждений в компиляторах и заставить их обрабатывать предупреждения так, как если бы это были ошибки.

Оглавление:

1. [Предупреждения в языке C++](#)
2. [Изменение уровня предупреждений](#)
 3. [Пользователям Visual Studio](#)
 4. [Пользователям Code::Blocks](#)
 5. [Пользователям GCC/G++](#)
6. [Обрабатывать предупреждения как ошибки](#)
 7. [Пользователям Visual Studio](#)
 8. [Пользователям Code::Blocks](#)
 9. [Пользователям GCC/G++](#)

Предупреждения в языке C++

На этапе компиляции компилятор проверяет, соответствует ли ваш код правилам языка C++. Если вы сделали что-то запрещенное, что нарушило синтаксис языка C++, то компилятор выдаст ошибку, предоставив вам как номер строки, содержащий ошибку, так и некоторый текст о содержании самой ошибки. Фактически, ошибка может находиться как в этой строке (которую сообщил вам компилятор), так и в строке перед ней. После того, как вы определили и исправили ошибочные строки кода, вы можете попробовать скомпилировать вашу программу еще раз.

Еще могут быть ситуации, когда компилятор видит ошибочный код, но не до конца в этом уверен (помните, что философия языка C++ заключается в выражении «Доверяй программисту!»). В таких случаях компилятор может выдать предупреждение. Предупреждения не останавливают процесс компиляции, но сообщают программисту, что что-то пошло не так.

Совет: Не позволяйте предупреждениям накапливаться. Решайте их по мере возникновения (так, как будто бы это были ошибки).

В большинстве случаев предупреждения могут быть устранены либо путем исправления ошибки, на которую указывает предупреждение, либо путем переписывания строки кода, генерирующей предупреждение, таким образом, чтобы предупреждение больше не генерировалось.

В редких случаях может потребоваться явно указать компилятору не генерировать конкретное предупреждение для рассматриваемой строки кода. Язык C++ не поддерживает такой способ решения предупреждений, но многие отдельные компиляторы (включая Visual Studio и GCC) предоставляют возможность (через *не портативные* директивы `#pragma`) временного отключения предупреждений.

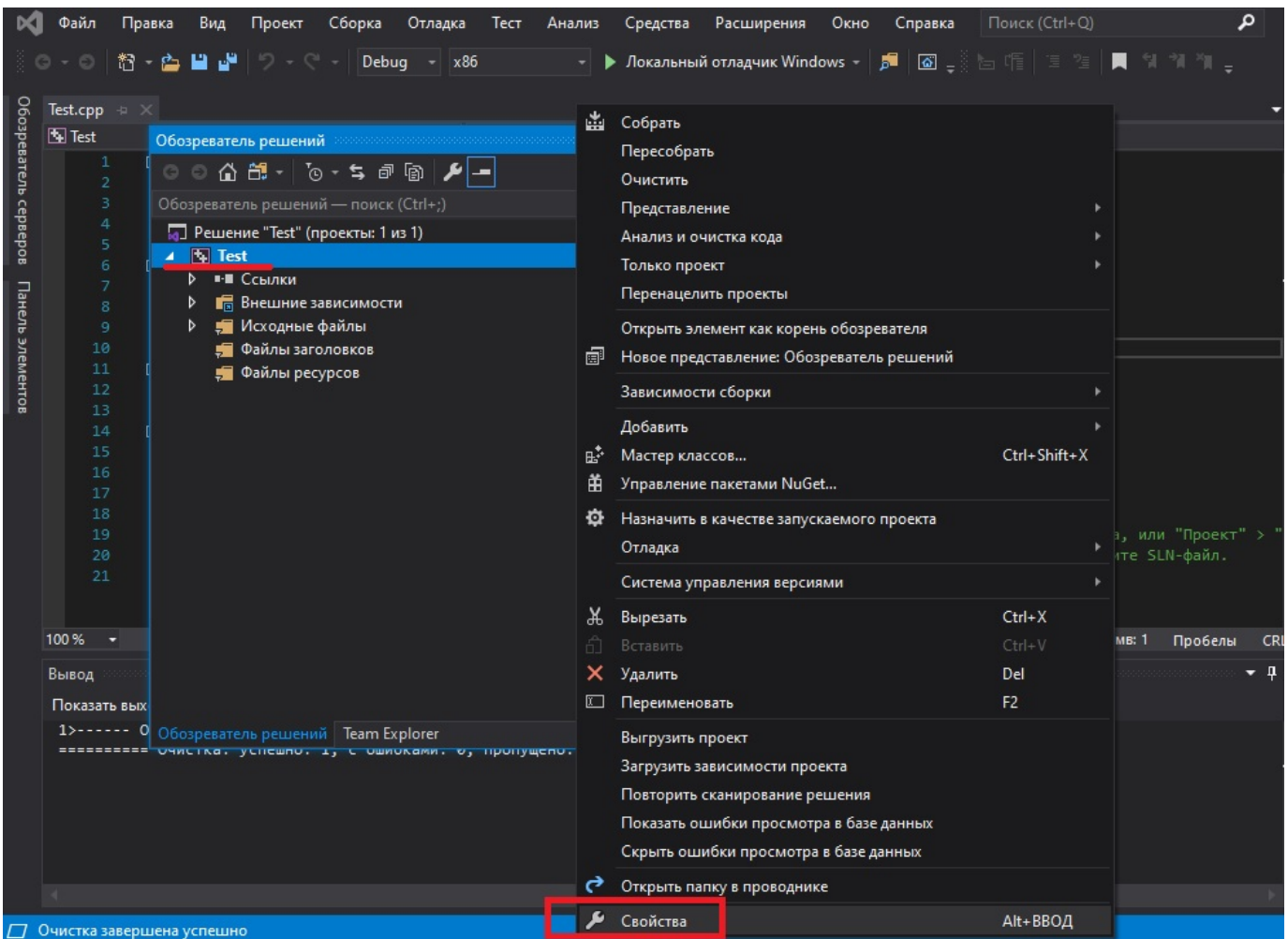
По умолчанию большинство компиляторов генерируют только предупреждения о наиболее очевидных проблемах. Однако вы можете попросить ваш компилятор быть более настойчивым в предоставлении предупреждений о вещах, которые он считает странными.

Совет: Сделайте максимальным уровень предупреждений от компилятора (особенно во время обучения). Это поможет вам определить возможные проблемы.

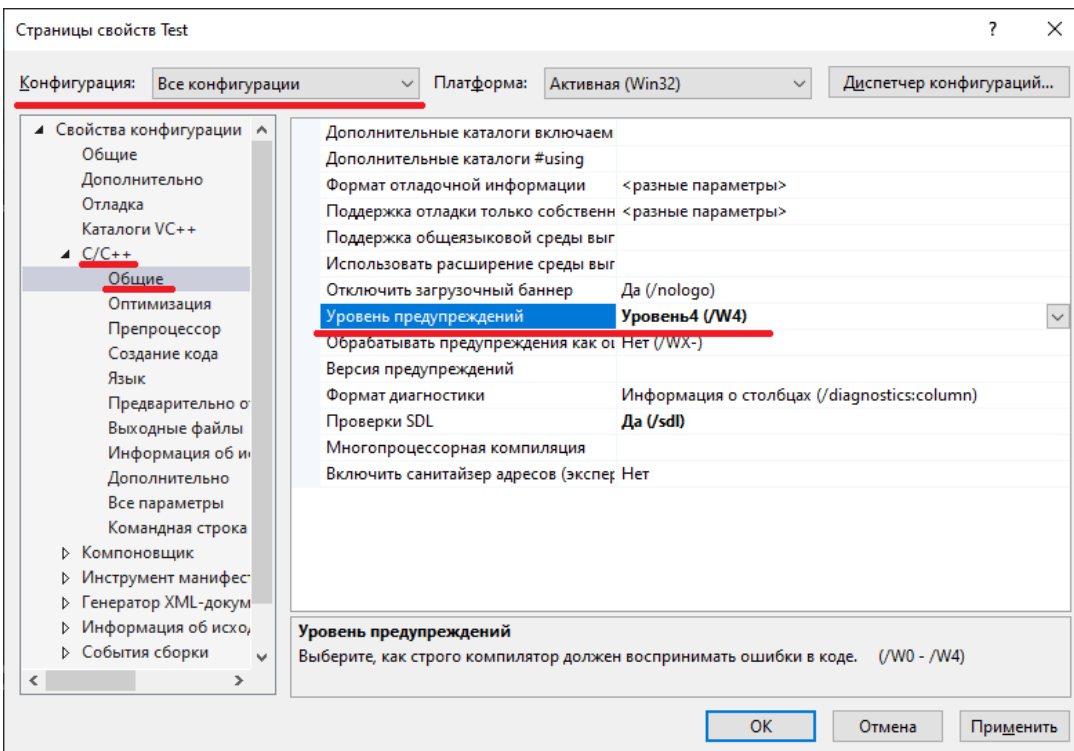
Изменение уровня предупреждений

Пользователям Visual Studio

Чтобы повысить уровень предупреждений в Visual Studio, щелкните правой кнопкой мышки по названию вашего проекта в меню "Обозреватель решений" > "Свойства":



В диалоговом окне вашего проекта убедитесь, что в пункте "Конфигурация" установлено значение "Все конфигурации". Затем перейдите на вкладку "C/C++" > "Общие" и в пункте "Уровень предупреждений" выберите значение "Уровень4 (/W4)":

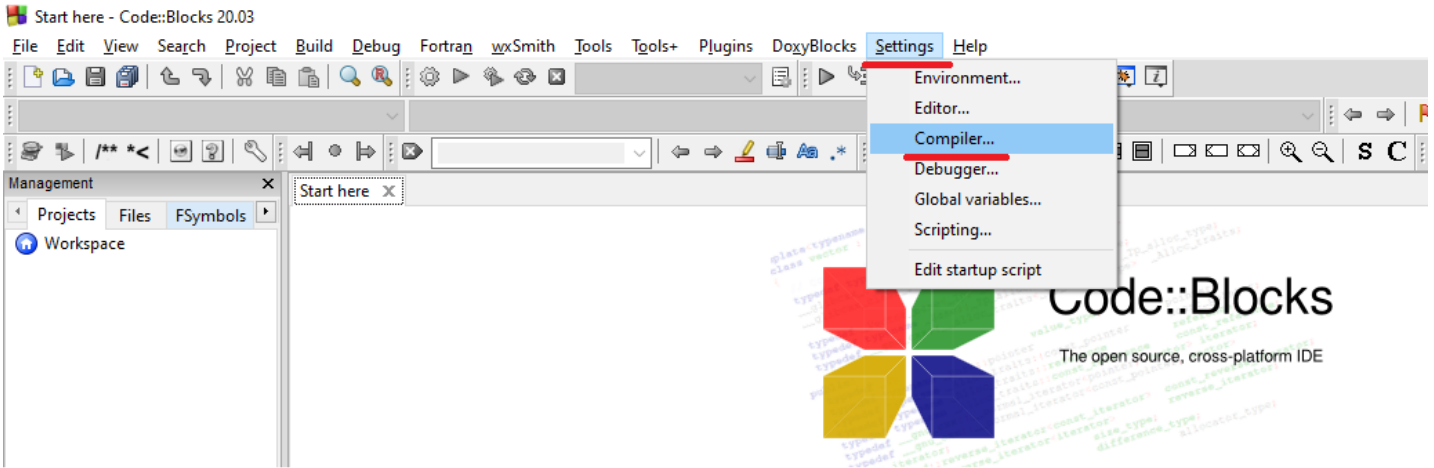


Затем нажмите "Применить" и "OK".

Примечание: Не выбирайте пункт "Включить все предупреждения (/Wall)", иначе вы будете погребены в предупреждениях, генерируемых Стандартной библиотекой C++.

Пользователям Code::Blocks

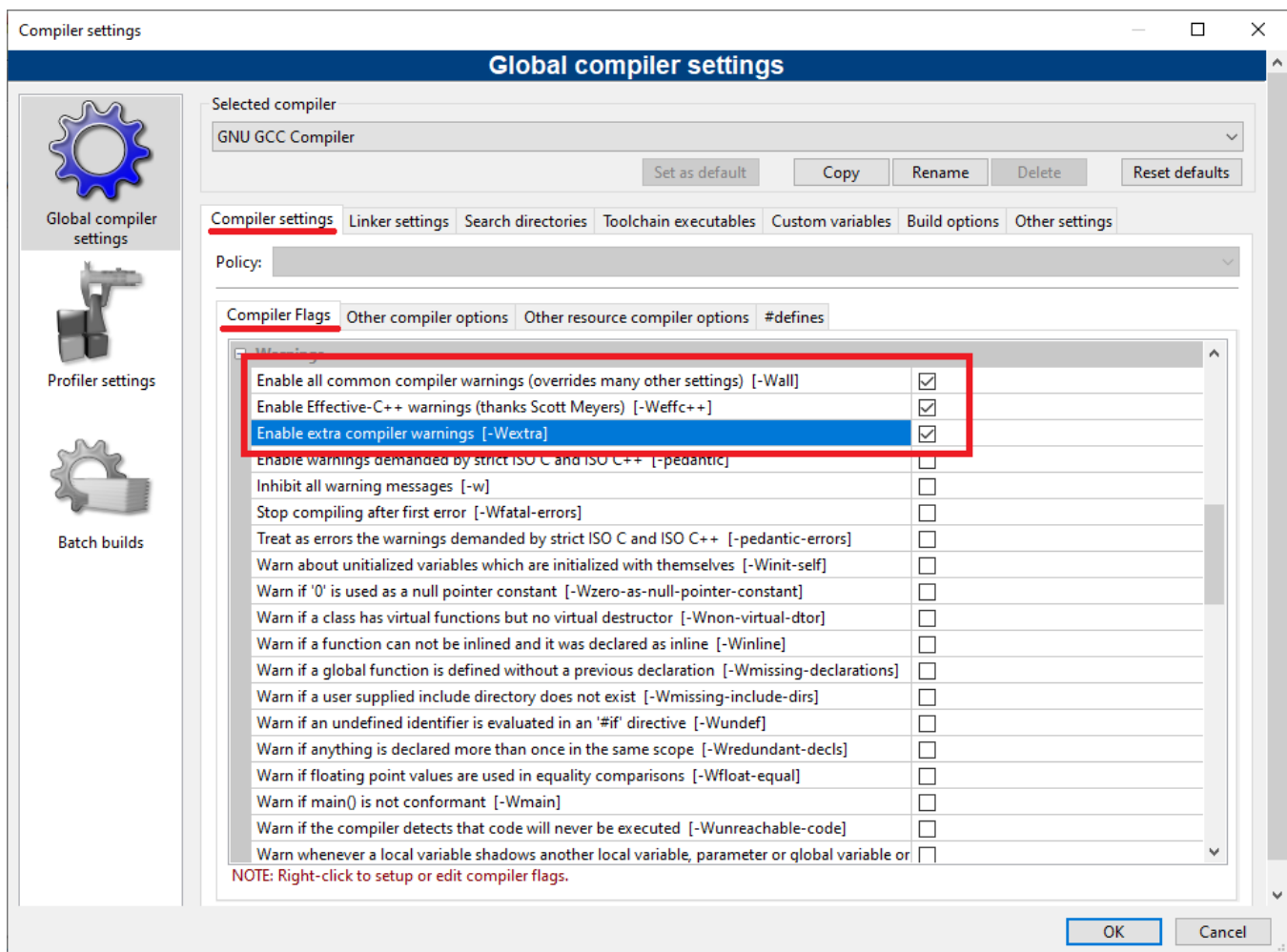
Перейдите в меню "Settings" > "Compiler":



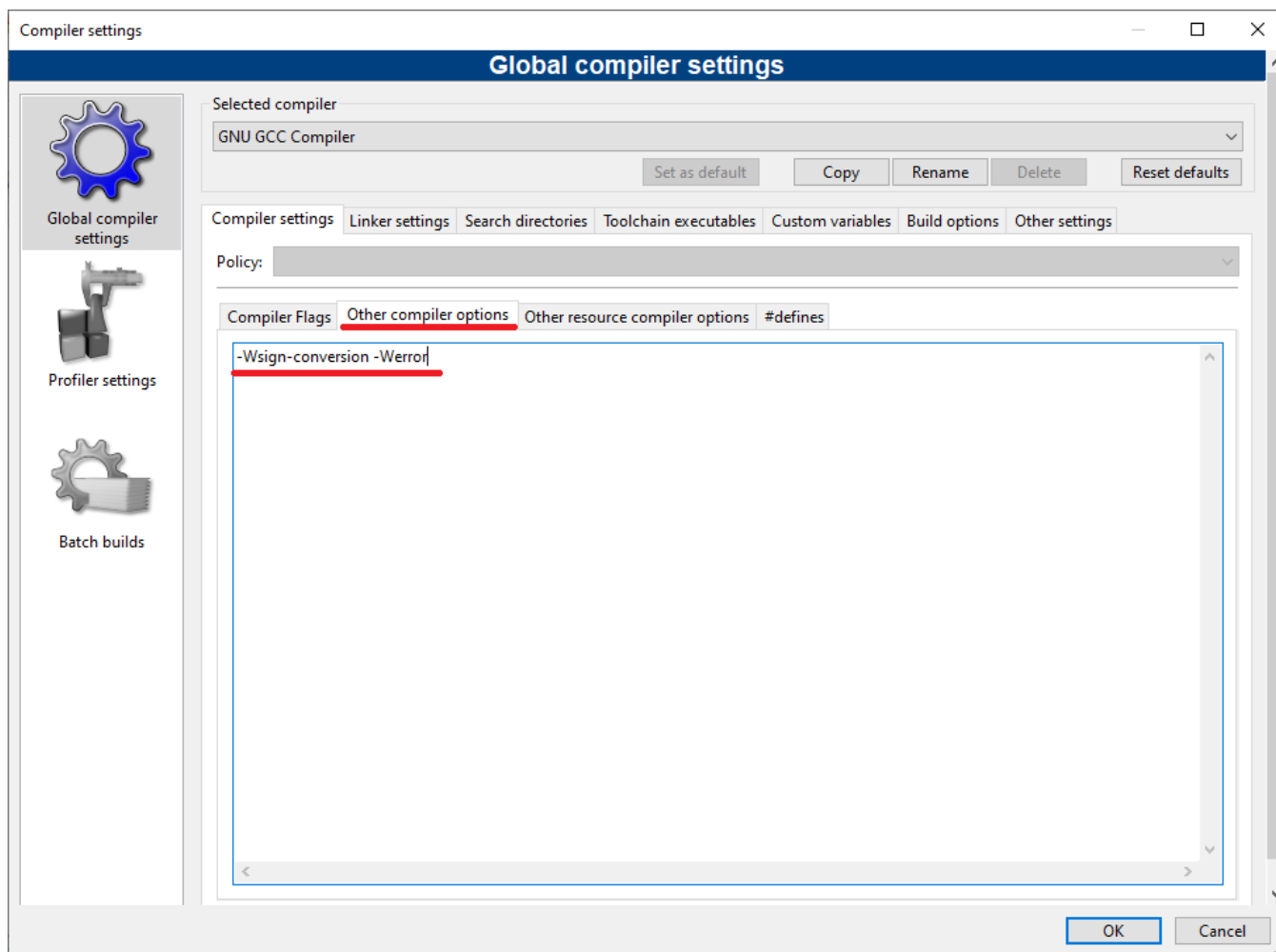
И на вкладке "Compiler settings" > "Compiler flags" поставьте галочки возле следующих пунктов:

- Enable all common compiler warnings (overrides many other settings) [-Wall]
- Enable Effective-C++ warnings (thanks Scott Meyers) [-Weffc++]
- Enable extra compiler warnings [-Wextra]

Смотрим:



Нажмите "OK" и затем перейдите на вкладку "Other compiler options" и добавьте в область редактирования текст `-Wsign-conversion -Werror`:



И нажмите "Ок".

Примечание: О параметре `-Werror` мы поговорим чуть позже.

Пользователям GCC/G++

Добавьте следующие флаги в вашу командную строку:

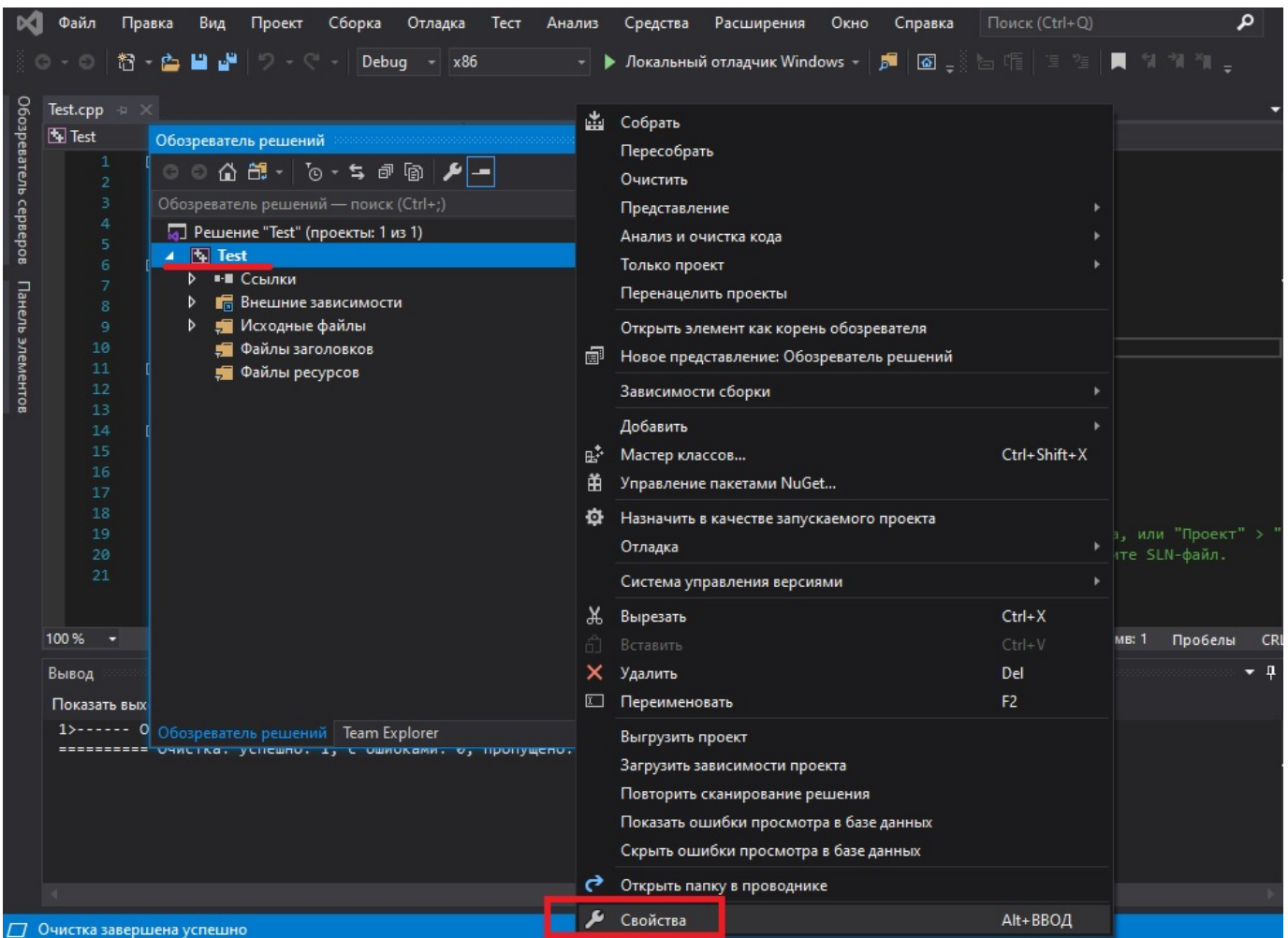
```
-Wall -Werror -Wextra -Wsign-conversion
```

Обрабатывать предупреждения как ошибки

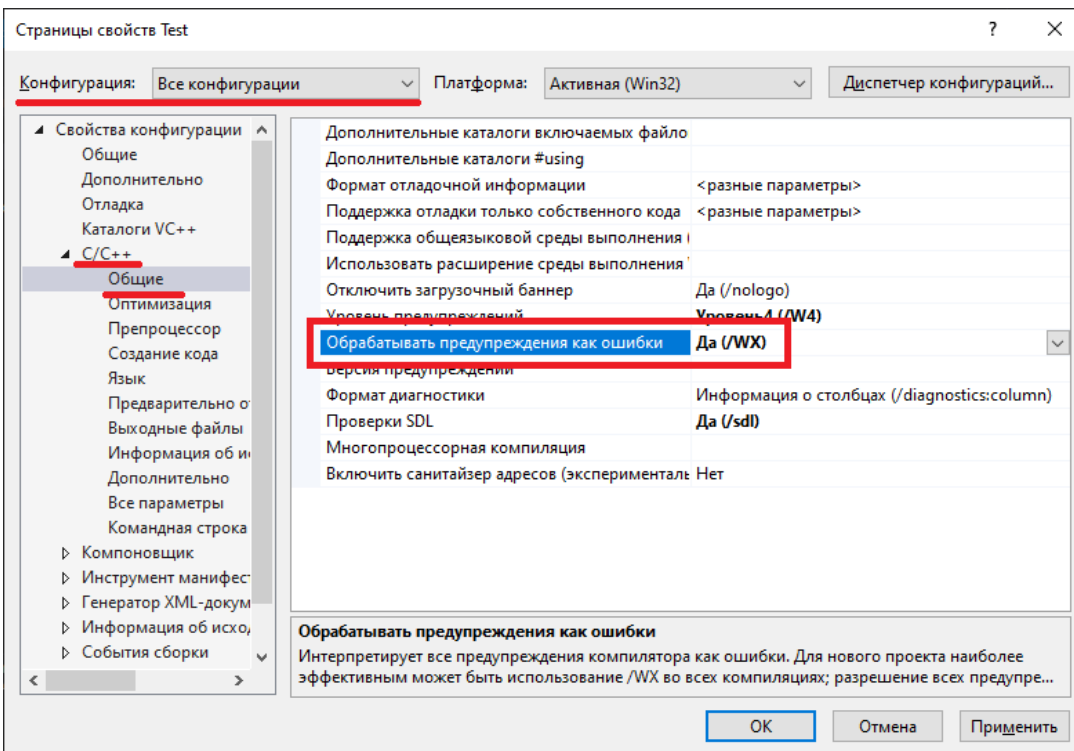
Вы также можете указать вашему компилятору обрабатывать все предупреждения так, как если бы они были ошибками (в таком случае, компилятор будет останавливать процесс компиляции, если обнаружит какие-либо предупреждения). Это хороший вариант заставить себя исправлять все предупреждения, особенно, если вам не хватает самодисциплины (как, впрочем, большинству из нас).

Пользователям Visual Studio

Чтобы обрабатывать все предупреждения как ошибки, щелкните правой кнопкой мышки по названию вашего проекта в меню "Обозреватель решений" > "Свойства":



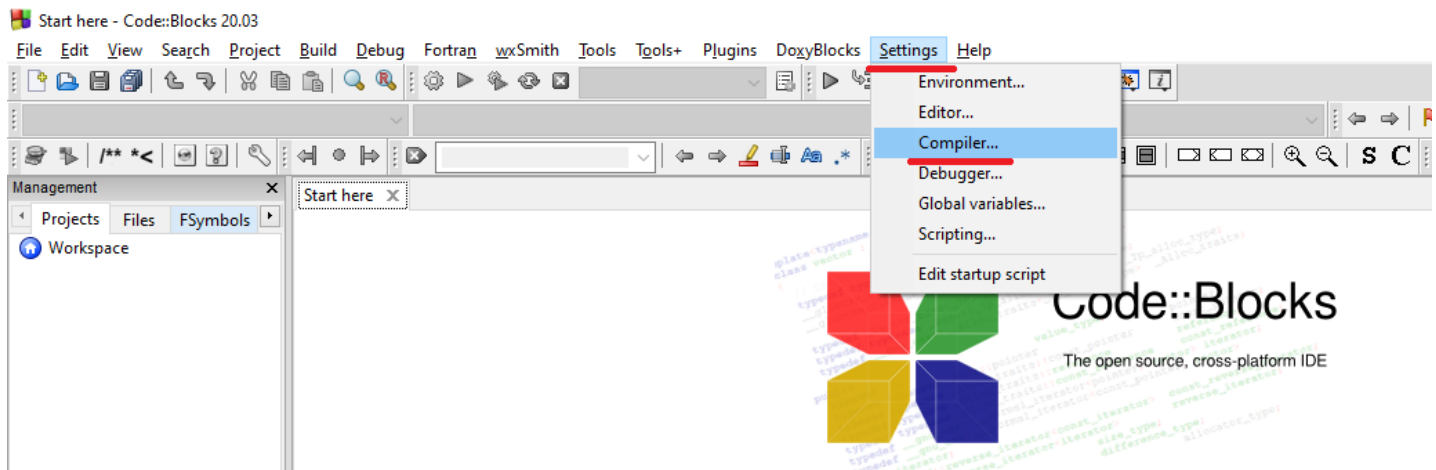
В диалоговом окне вашего проекта убедитесь, что в поле "Конфигурация" установлено значение "Все конфигурации". Затем перейдите на вкладку "C/C++" > "Общие" и в пункте "Обрабатывать предупреждения как ошибки" выберите значение "Да (/WX)":



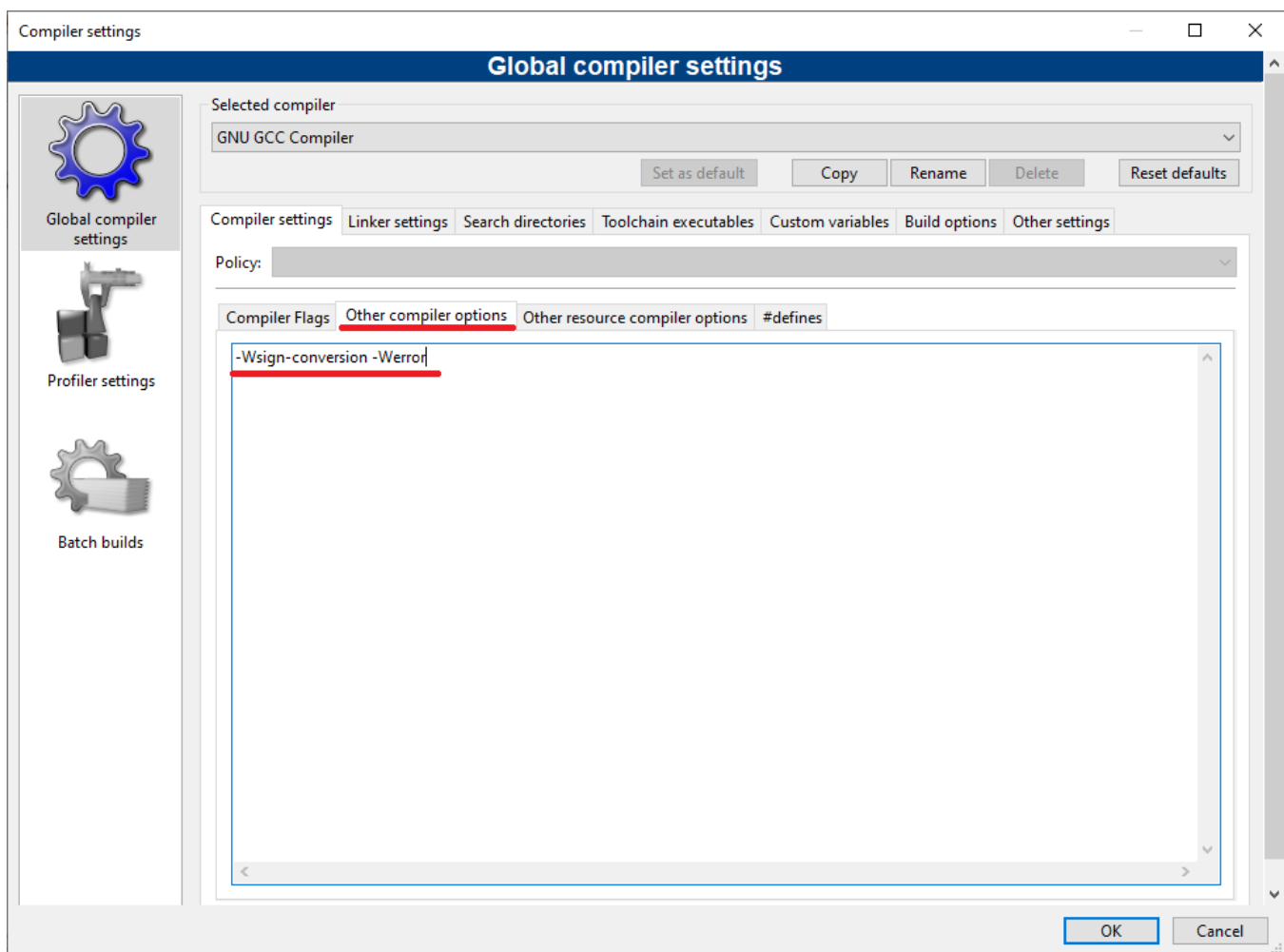
Затем нажмите "Применить" и "OK".

Пользователям Code::Blocks

Перейдите в меню "Settings" > "Compiler":



Затем "Compiler settings" > "Other compiler options" и добавьте -Werror в область редактирования:



Затем нажмите "OK".

Пользователям GCC/G++

Добавьте следующий флаг в вашу командную строку:

-Werror

Оценить статью:

★★★★★ (85 оценок, среднее: 4,96 из 5)

[← Конфигурация компилятора: Расширения компилятора](#)[Конфигурация компилятора: Выбор стандарта языка C++ →](#)

Комментариев: 2



1. *Rejer:*
[24 сентября 2020 в 10:41](#)

Здравствуйте, спасибо вам огромные за уроки!!!!

[Ответить](#)



1. *Юрий:*
[24 сентября 2020 в 15:05](#)

Пожалуйста)

[Ответить](#)

Добавить комментарий

Ваш E-mail не будет опубликован. Обязательные поля помечены *

Имя *

Email *

Комментарий

☐ Сохранить моё Имя и E-mail. Видеть комментарии, отправленные на модерацию

☐ Получать уведомления о новых комментариях по электронной почте. Вы можете [подписаться](#) без комментирования.






[TELEGRAM](#)  [КАНАЛ](#)

Электронная почта



[ПАБЛИК](#) 

ТОП СТАТЬИ

-  [Словарь программиста. Сленг, который должен знать каждый кодер](#)
-  [Урок №1. Введение в программирование](#)
-  [70+ бесплатных ресурсов для изучения программирования](#)
-  [Урок №1: Введение в создание игры «Same Game»](#)
-  [Урок №4. Установка IDE \(Интегрированной Среды Разработки\)](#)

- [Ravesli](#)
- - [О проекте](#) -
- - [Пользовательское Соглашение](#) -
- - [Все статьи](#) -
- Copyright © 2015 - 2020