**Перечень запретов, рекомендаций и требований для оформления исходных кодов проекта на C++**

**Запреты:**

1. **Запрет на использование макросов для переименования операторов и ключевых слов.**

Пример запрещенного использования:

#define closed protected

1. **Запрет на создание глобальных идентификаторов, начинающихся с подчерка или содержащих \_\_.**

Пример запрещенного использования:

int \_\_globalVar;

**Требования:**

1. **Инициализация локальных переменных перед использованием.**
   * Пример правильного использования:

int x = 0;

1. **Использование перечислимых типов данных (enum) вместо препроцессорных констант.**

Пример правильного использования:

enum Color { RED = 1, GREEN = 2, BLUE = 3 };

Color light = RED;

**Рекомендации:**

1. **Рекомендация использовать типизированные константы вместо макросов для обозначения констант.**

Пример рекомендуемого использования:

const int my\_const = 0;

1. **Рекомендация давать файлам имена, сходные с именами функций или классов, объявленных в них.**

Пример рекомендуемого использования:

// Файл: basic\_class.h

class CBasicClass {

//…

};

**Чек-лист для проверки корректности кода перед инспекцией:**

1. **Проверка инициализации всех локальных переменных.**

Убедитесь, что все локальные переменные инициализированы перед использованием.

1. **Проверка отсутствия макросов для переименования операторов и ключевых слов.**

Убедитесь, что в коде нет макросов, переименовывающих операторы или ключевые слова.

1. **Проверка использования перечислимых типов (enum) вместо препроцессорных констант.**

Убедитесь, что для констант используются перечислимые типы, а не макросы.

1. **Проверка отсутствия глобальных идентификаторов, начинающихся с подчерка или содержащих \_\_.**

Убедитесь, что в коде нет таких идентификаторов.

1. **Проверка длины строк кода (не более 100 символов).**

Убедитесь, что длина строк не превышает 100 символов.

1. **Проверка наличия комментариев в конце операторов ветвлений и циклов, если тело содержит более 5 строк.**

Убедитесь, что такие комментарии присутствуют.

1. **Проверка использования директив условной компиляции вместо комментирования блоков кода.**

Убедитесь, что для исключения кода из компиляции используются директивы условной компиляции.

1. **Проверка отсутствия оператора goto.**

Убедитесь, что оператор goto не используется, за исключением крайних случаев.

1. **Проверка наличия заголовков файлов и функций в виде комментариев установленного образца.**

Убедитесь, что каждый файл и функция имеют соответствующие заголовки.

1. **Проверка использования констант в условиях на первом месте.**

Убедитесь, что в условиях константы стоят на первом месте, например:

if (0 == j) { ... }

Этот перечень и чек-лист помогут обеспечить соблюдение стандартов кодирования и улучшить качество кода перед его инспекцией.