

Coding style guidelines

Общие замечания

- При написании кода старайтесь следовать трем правилам оптимизации: сначала пишите код, который запускается и работает, затем заставьте его работать правильно, и уже после этого, при необходимости, оптимизируйте код.
- Обязательно покрывайте тестами те участки кода, где легко допустить ошибку.
- Код должен компилироваться без каких-либо предупреждений (warnings) на максимальном уровне предупреждений компилятора.

Стиль оформления кода

- Не используйте символы табуляции (вместо них для отступов используйте четыре пробела)
- При расстановке скобок и выравнивании кода придерживайтесь K&R style (https://en.wikipedia.org/wiki/Indent_style#K.26R_style)

```
int main(int argc, char *argv[])    {  
  
    ...  
  
    while (x == y) {  
        something();  
        somethingelse();  
        if (some_error) {  
            do_correct();  
        } else  
            continue_as_usual();  
    }  
    finalthing();  
    ...  
}
```

Именование файлов

- Имена файлов могут содержать латинские буквы, цифры и символ подчеркивания (_).
- Расширение заголовочных файлов - .h
- Расширение файлов реализации - .cpp

Структура файла

- В начале заголовочных файлов используйте #pragma once.
- Запрещено использовать директиву using в заголовочных файлах.
- После #pragma once в заголовочных файлах размещайте директивы #include в следующем порядке: сначала соответствующий заголовочный файл (для файлов реализации), затем файлы проекта, и в конце заголовочные файлы стандартной библиотеки и сторонних библиотек.
- Не включайте неиспользуемые файлы.
- Рекомендуется использовать пустую строку для отделения логических блоков кода.
- Избегайте больших файлов (> 300 строк), больших функций (> 40 строк) и длинных строк кода (> 78 символов).

Названия переменных, классов и констант

- Давайте осознанные имена переменным, за исключением, быть может, общепринятых обозначений для итераторов и индексов массивов (напр, for (int i = 0; i < size; ++i)).
- Не начинайте название переменных с символа нижнего подчеркивания (они зарезервированы стандартом).

- Используйте Pascal style для названия классов и структур (напр, class DefaultParameters, struct UserInfo).
- Используйте Camel style для названия переменных, функций, членов и методов класса (напр., void parseArguments(const CommandLineArguments& args)).
- Называйте константы и дефайны препроцессора заглавными буквами, разделяя слова символом подчеркивания.

Классы и структуры

- Используйте существительные (и, при необходимости, несколько прилагательных) в названиях классов.
- Объявляйте методы и поля класса в следующем порядке: public, protected, private.
- В пределах блока доступа, объявляйте сначала методы, затем, через пустую строку, поля класса.
- Не используйте множественное определение полей класса - каждое поле на новой строке.
- К private и protected поля класса добавляйте символ подчеркивания (напр., double factor_).
- Называйте методы, которые проверяют некоторое свойство, как isSmth или hasSmth.
- При объявлении методов используйте const везде, где это возможно.

Преобразование типов

- Используйте C++ операторы преобразования типов (static_cast, dynamic_cast, reinterpret_cast, const_cast) вместо явного преобразования типа (как в C).
- Объявляйте конструктор от одного параметра explicit чтобы избежать неявного преобразования типа.

Управление памятью

- Избегайте использования new и delete. Используйте их только тогда, когда уверены, что умные указатели (напр., std::unique_ptr и std::shared_ptr) и стандартные контейнеры (напр., std::vector) не применимы.
- Для передачи владения динамически созданным объектом используйте std::unique_ptr
- Не используйте умные указатели с разделяемым доступом (std::shared_ptr) там, где в этом нет необходимости. Используйте вместо этого std::unique_ptr, где очевидно, кто владеет объектом и ответственен за его удаление.