**Стандарт оформления кода** – набор соглашений, правил и рекомендаций, применяющихся при написании исходного кода. В отличие от стандарта программирования, не описывает устройство или структуру кода, концентрируясь на вопросах его “внешнего” устройства. Никак не влияет на корректность программы, однако способен увеличивать легкость её восприятия. Следует понимать, что разница между стандартами оформления кода – вопрос скорее вкусовой; куда важнее не выбор единственно правильного идеального стандарта кодирования, но использование одного и того же стандарта оформления кода во всем проекте. Единый стандарт кода позволяет заметно легче воспринимать программу; программа, написанная в нескольких стандартах, читается также тяжело, как текст с хаотично меняющимися шрифтами.

**Именование объектов:**

1. Длина имени идентификатора ограничена **16 символами**.
2. Использование одинаковых имен в разных пространствах **допустимо**.

**Оформление имен объектов**

1. Если какие-либо случаи именования не указаны ниже, следует использовать стиль именования **camelCase**.
2. Любые функции именовать в стиле **snake\_case**, заглавные буквы не использовать.
3. Имена констант, объявленных глобально, указывать заглавными буквами.
4. Имена констант, объявленных локально, а также имена локальных переменных – в стиле snake\_case.
5. Допускается применение имен переменных, начинающихся с символа “\_” для указания принадлежности переменной к пользовательскому коду, а не к полям библиотек.

Пример: int \_size;

Для имен пространств, функций и классов по этой же причине разрешена приставка “m”.

Пример: namespace mLab {}

1. Поля классов именуются на подобие локальных переменных. Методы классов – аналогично функциям.
2. Использование длинных идентификаторов допустимо.

**Отступы**

1. Допустимо писать однострочные тела циклов и условных блоков в одной строке или на одном уровне с циклом/условием.
2. Символ отступа – табуляции.
3. При использовании фигурных скобок, открывающую указывать на той же строке, что и начало оператора/объявления сущности. Закрывающую скобку указывать на отдельной строке. Тело оператора должно находиться по отношению к родительской структуре на один отступ дальше.
4. При использовании круглых скобок указывать аргументы или операнды на одной строке. Если строка превышает приемлемую длину, расположить выражение под скобкой на нескольких строках.

**Стиль комментариев**

1. Рекомендуемый объем комментариев: 1-4 строки при комментировании назначения и принципа работы функции. 1-2 строки при комментировании конкретной части кода.
2. Комментарии перед функциями указывать тремя линиями - “///”.
3. Допустимо использование одно/многострочных комментариев.

**Ссылки на репозитории:**

<https://github.com/ytwog/timp_lab1>

<https://github.com/ytwog/tmp_lab1_func>