Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Информатика и вычислительная техника

**ЛАБОРАТОРНАЯ РАБОТА 8**

По Технологии и методы программирования

наименование дисциплины

Преподаватель \_\_\_\_\_\_\_\_ Матковский И.В.

подпись, дата фамилия, инициалы

Студент КИ16-02 №031624307 \_\_\_\_\_\_\_\_ Полушин С.В.

номер группы, зачетной книжки подпись, дата фамилия, инициалы

Красноярск 2019

Анализ чужого программного кода

**Цель:** научиться основам командной работы

**Ход работы:**

Список изменений, которые, на мой взгляд, могут улучшить программу:

* Разделить строки, в выходном файле, на более короткие – для удобства восприятия информации пользователем. При выводе информации в выходной файл, получаются слишком длинные строки – что, на мой взгляд, будет неудобно для восприятия (т.к. чтобы прочесть полностью строку – придётся развернуть файл на полный экран, либо пользоваться мышкой – для перемещения по файлу).
* Разделить объекты, в выходном файле, переносом на новую строку или другим разделителем – для удобства восприятия информации пользователем.
* Привести код к стандарту кодирования – для удобства восприятия кода. Так, к примеру, в файле “pch.h” название класса Node – начинается с большой буква, а названия остальных классов – с маленькой. Также названия переменных и функций начинаются то с большой буквы, то с маленькой, примером могут послужить переменные – Next и data в классе container, и функции – get\_count и Out\_only\_plane, в том же классе.
* Разделить код по смысловым файлам – для удобства восприятия кода. К примеру – все классы переместить в отдельный “\*.cpp” файл, все функции определенного транспорта в другой файл и т.п.
* Разделить код разделителями, по смысловым частям - для удобства восприятия кода. Так как функции и классы, в коде, пишутся один за другим, без разделителей, это может вызвать проблемы с восприятием кода.
* Удалить лишний код – для “чистоты” кода. Например, в файле “pch.h”, , в классе container, функция get\_Top и get\_count - не используются в коде.
* В классе container, в деструкторе используется функция Clear, следовательно в теле main можно удалить строку с явным её вызовом, т.к. деструктор сам ее вызовет.

**Ссылка на репозитории (bitbucket):**

<https://github.com/dostoevsky-plus/OOP>