

Чеклист по коду программ

Илья Булгаков

Московский физико-технический институт

Москва, 2023

Содержание

- 1 Правила кодирования
- 2 Разбор типичных ошибок

Чеклист: структура кода

- Проведена функциональная декомпозиция, т.е. код решения грамотно разбит на функции/классы
 - в `main` должен находиться ввод данных, вызов функции, которая решает задачу, вывод результата
 - Каждая функция или класс должны решать ровно одну задачу
 - Имя функции должно отвечать на вопрос «Что сделать?». Не по имени автора алгоритма, не сокращение, не `foo`.
- Переменные должны объявляться максимально близко к месту использования.
- Все переменные базового типа должны при объявлении инициализироваться, классы и структуры должны иметь конструктор.
- Глобальными переменными пользоваться нельзя.
- Нельзя создавать константы через `#define`
- На каждый вызов аллокации памяти должен быть вызов освобождение памяти, `new -> delete`. Начиная со второго модуля штраф “-1” за любую утечку памяти или неконтролируемое копирование контейнера в сдаваемом и показываемом коде.

Чеклист: именование и стиль

- Все переменные и функции должны иметь осмысленное имя на англ. языке. Однобуквенные имена разрешаются только для счётчиков цикла
- Необходимо придерживаться одного стиля написания кода. Вы должны быть последовательны в оформлении программ, все должно быть единообразно. Рекомендуется взять за основу один из публичных стандартов оформления.
 - Нельзя использовать camelCase вперемешку со snake_case.
 - В коде не должно быть больше одной пустой строки подряд,
 - Вы должны использовать единый тип отступов: либо пробелы, либо табуляции. Размер отступа важен и должен быть постоянным
 - Вы должны использовать единые правила для расстановки фигурных и круглых скобок, расстановку пробелов вокруг них
- Следует предпочитать один из видов комментариев по коду программу (либо `//`, либо `/**/`)

Разбор типичных ошибок

Main solver

```
63 + int main() {  
64 +     int n;  
65 +     std::cin >> n;  
66 +     auto arr = readArray(n);  
67 +  
68 +     Solver solver(arr);  
69 +     solver.solve();  
70 +  
71 +     printSolution(solver.getSolutionIndices());  
72 +     return 0;  
73 + }
```

Чеклист: что реализовывать, а что нет?

- Не стоит реализовывать простые операции, которые уже есть в стандартной библиотеке: следует предпочитать `std::max`, `std::swap` и т.д.
Исключение: если у вас задача – реализовать сортировку, то использовать `std::sort` нельзя, если динамический массив – `std::vector` и т.д.
- Массивы неконстантного размера, а также большого константного размера, необходимо выделять в динамической памяти.

Чеклист: оформление условия задачи

- В шапке сдаваемого сrr должно находиться закомментированное условие задачи.

Чеклист: классы

- Классы:
 - 1). `public` секция выше `private`
 - 2). Все поля должны быть `private`
 - 3). Реализация всех методов, длиннее одной строки, должна быть вынесена
 - 4). Правило трех

Настройка линтера 2023

- Кодстайл основан на гугловом
- Указатель выравниваем по левому краю (Type* t)
- Длина функции: не больше 60
- Длина строки: не больше 80
- Везде принят формат именования имен CamelCase, кроме переменных, где snake_case
- В функциях разрешено не более 5 параметров
- Все константы с префиксом k
- Все параметры в нижнем регистре, локальные переменные тоже
- Минимальная длина имени переменной 2 // Исключение x, y, z. Не касается переменных цикла
- Magic numbers запрещены кроме 1, 2, 3, 4, 5, а так же отрицательные аналоги и 0
- auto надо использовать, если тип больше 20 символов