



ШКОЛА АНАЛИЗА ДАННЫХ



Семинар № 1. Введение

Алгоритмы и структуры данных

Правила жизни на курсе

Контакты

| `ys.algorithms@gmail.com` (*общая почта курса*)

— Срочные письма — с темой «URGENT!!!»

| Вики-страница курса:

<https://wiki.school.yandex.ru/shad/groups/2017/Semester1/Algorithms1/>

| Канал с новостями: <https://t.me/joinchat/AAAAAEJ2hkBm1zv3pdCenw>

| Чат: <https://t.me/joinchat/CBXLm0LqXEtwuuxQTn-kWw>

Структура курса

- | Лекции Максима Бабенко и Михаила Левина
- | 6 домашних заданий (контестов)
- | За весь семестр 3 задачи на ревью
- | Семинары:
 - Разбор сложных задач по теме (можно додумать и сдать в контест)
 - Комментарии по ревью
 - Построение интерфейсов
 - Фишки C++

Составляющие оценки

Полностью сданные ревью + баллы за контесты

- | **Отлично:** 1500 баллов и 2 ревью *или* 1300 баллов и 3 ревью
- | **Хорошо:** 1300 баллов и 2 ревью *или* 1100 баллов и 3 ревью
- | **Зачёт:** 900 баллов и 2 ревью

Ревью

| Теория (pdf-файл)

- Описание алгоритма решения
- Доказательство корректности
- Временная сложность — асимптотика
- Затраты памяти — асимптотика

| Интерфейс

- Заголовки функций и методов, описания классов
- Функция `main` целиком

| Полный код

Как набрать баллы?

- | Контесты
- | Задачи с семинаров
- | Теоретические задачи
- | Конкурсные задачи от Поиска
- | Вклад в Open source:
<https://wiki.school.yandex.ru/users/x4mmm/osshack/>
- | Челлендж

Особенности задач в контекстах

- | Предварительная проверка стиля

- | Санитайзеры:

- <https://wiki/shad/groups/2017/Semester1/Algorithms1/debugging/>

- | Оффлайн-задачи

Дедлайны

| По теории

- Нужно успеть отправить адекватную теорию

| По интерфейсу + конец контеста

- Нужно успеть отправить адекватный интерфейс на ревью

- Сначала сдать теорию не обязательно, но очень желательно

| Промежуточный дедлайн: 1 ноября

- Сдать в тестирующую систему не меньше 2 задач

| По полному коду: **20 декабря — конец семестра**

- Нужно не отправить код, а *исправить все замечания* и сдать полностью

Как испортить себе оценку

- | Все дедлайны **жёсткие**
- | Если просрочить дедлайн по теории или интерфейсу, ревью может быть зачтено только после конца семестра
- | Если к концу семестра баллов или ревью не хватает даже для зачёта, студент становится *должником*
- | Должник досдаёт долги, но уже не может получить больше зачёта
- | К концу семестра (20 декабря) — хотя бы один сданный курс
- | К 10 января — хотя бы два
- | **1 февраля** — окончательный и бесповоротный дедлайн для должников

Стайл-гайд

Зачем нужен стайл-гайд?

- | Полезная привычка
- | Повышение читаемости
- | Увеличение производительности
- | Уменьшение вероятности ошибки

| <https://wiki.school.yandex.ru/shad/AlgorithmsStyleguide/>

| Основа — [Google C++ Style Guide](#)

Основные требования

- | Код должен удовлетворять указанным в условии задачи ограничениям по времени и памяти
- | Если с прохождением ограничений всё в порядке, дальше оптимизировать не нужно
- | Код должен быть максимально простым, читаемым и гибким
- | Если это не сказывается на читаемости, можно пооптимизировать по мелочам
- | Наибольшее зло — **дублирование кода**

Структура программы

- | Функция `main` не делает ничего, кроме запуска трёх блоков функций:
 - Считывание входных данных
 - *Одна* функция решения задачи, возвращающая ответ по заданным входным данным
 - Вывод результата
- | Все смысловые части выносятся в отдельные функции и называются соответственно

```
#include <iostream>

int ComputeSum(int first_number, int second_number) {
    return first_number + second_number;
}

int ReadNumber(std::istream &in_stream = std::cin) {
    int number;
    in_stream >> number;
    return number;
}

void PrintNumbersSum(int numbers_sum, std::ostream &out_stream = std::cout) {
    out_stream << numbers_sum << std::endl;
}

int main() {
    std::ios_base::sync_with_stdio(false);
    std::cin.tie(nullptr);

    int first_number = ReadNumber();
    int second_number = ReadNumber();

    int numbers_sum = ComputeSum(first_number, second_number);

    PrintNumbersSum(numbers_sum);

    return 0;
}
```


Популярные грабли

- | В задаче с тяжёлым выводом используется `std::endl`
- | C-style массивы, в том числе статические
- | `rand()` или даже `rand() | (rand() << 15)` вместо модуля `<random>`

Литература

Литература

- | Кормен, Лейзерсон, Ривест, Штайн. *Алгоритмы: построение и анализ*
- | Роберт Седжвик. *Алгоритмы на C++*
- | Bjarne Stroustrup. [*The C++ Programming Language*](#)
- | Scott Meyers. [*Effective C++*](#)
- | Scott Meyers. [*Effective STL*](#)
- | Scott Meyers. [*Effective Modern C++*](#)
- | Martin Fowler. [*Improving the Design of Existing Code*](#)



В следующий раз —
о тестировании кода