

# Регламент выполнения и защиты курсовой работы

## Требования к отчёту по курсовой работе

В отчёте должны быть:

- титульный лист, лист задания, аннотация, содержание;
- формальная постановка задачи;
- описание алгоритма;
- описание структур данных и функций;
- описание интерфейса пользователя - для работ с визуализацией;
- тестирование;
- исследование реализованной структуры / алгоритма;
  - исследование всех операций над структурой
  - графики практической и теоретической оценки
- программный код (в приложении);
- выводы.

Титульный лист, лист задания и программный код являются абсолютно необходимыми разделами, без которых даже не может быть начат процесс защиты.

Отчёт по курсовой работе должен быть оформлен в соответствии с шаблоном оформления курсовой работы: <https://etu.ru/ru/studentam/dokumenty-dlya-uchebny/>

## Общие требования к работе

### Реализация структур и алгоритмов

- Все необходимые в рамках выполнения работы структуры должны реализовываться студентом **самостоятельно**
  - **Не допускается** использование встроенных / импортированных классов, представляющих из себя структуры или их аналоги (Python: **list**, **dict**, **set** и др. C/C++: **map**, **set**, **deque** и пр.)
    - Использование в тестировании / исследовании - разрешается, в иных частях работы - по согласованию с преподавателем.
- Все необходимые в рамках выполнения работы алгоритмы должны реализовываться студентом **самостоятельно**
  - **Не допускается** использование встроенных / импортированных функций/классов, предоставляющих реализацию алгоритмов

### Исследование реализованных структур и алгоритмов

- Исследование реализованной структуры проводится по всем реализованным операциям

- Должны быть приведены обоснования лучшего, среднего, худшего случаев работы операций, а также способ формирования входных данных (или характеристик структуры) для этих случаев (если применимо)
- Графики практической и теоретической оценки (для всех операций и случаев)
- Исследование реализованного алгоритмы проводится по всем случаям работы алгоритма
  - Должны быть приведены обоснования лучшего, среднего, худшего случаев работы, а также способ формирования входных данных (или настроек алгоритма) для этих случаев (если применимо)
  - Графики практической и теоретической оценки (для всех случаев)

См. также раздел «Исследования в рамках ЛР/КР» в документе [\[ALG\] Типовые ошибки отчета / работы](#)

## Требования к обоснованию

### Цель обоснования

Продемонстрировать понимание не только выбранной структуры данных, но и альтернативных вариантов, а также умение проводить системный анализ задачи на основе её операционной семантики и характеристик данных.

### Формализация постановки задачи

#### Операции

**Явный нумерованный список** всех операций, которые должна поддерживать структуры в рамках решаемой задачи:

- ```
# Пример
1. Добавление записи
2. Поиск по уникальному ключу
3. Удаление по ключу
4. Получение всех данных в отсортированном порядке.
```

#### Анализ частоты операций

Качественная или количественная (если применимо) оценка относительной частоты выполнения **каждой операции**:

- если указана в задании - берётся из задания
- иначе - вводится студентов самостоятельно (или берётся равномерной)

#### Характеристики данных

- Объем данных
  - Ожидаемое количество элементов ( $100 / 10^6 / 10^9$ )
  - Фиксированность размера данных - постоянное (после создания структуры) или растущее/уменьшающееся в процессе работы.
- Динамика данных
  - Данные статичны после загрузки или активно изменяются (вставка/удаление).
- Ключи (если применимо)
  - Тип ключа (целое число, строка, составной ключ)
  - Характер доступа (случайный, последовательный)
  - Наличие уникальности.
- Потребления по памяти (по желанию)
  - Размер хранимого объекта (байты, килобайты).

## Критерии выбора

Явно перечислить, какие **критерии оптимальности** важны для конкретно вашей задачи и почему.

# Пример

1. Скорость доступа по ключу (поиск, вставка, удаление).
2. Скорость выполнения специализированных операций (поиск минимума/максимума, слияние).
3. Поддержка упорядоченности данных.
4. Предсказуемость времени выполнения.
5. Эффективность по памяти.

## Анализ альтернативных структур данных

- Необходимо **минимум 2-3 реальных альтернативы**
- Можно оформить в виде таблицы «Критерий / Структура»
- Для каждого аналога необходимо провести анализ по обозначенным критериям выбора (см. предыдущий раздел «Критерии выбора»)
  - Теоретическая сложность операций
    - Для каждой операции из раздела «Операции» указать сложность при работе с исследуемой структурой (средний и худший случаи)
  - Практические особенности
    - Указать имеющиеся нюансы реализации, усложняющие работу, и вероятность встречи с ними (например, рехеширование для хэш-таблиц, сложность балансировки и пр.)
  - Пригодность под характеристики данных
    - Подходит ли структура под ожидаемый объем, тип ключа, динамику
  - Вывод по каждому аналогу
    - Вывод, почему аналогу подходит / не подходит

## Выбор структуры

- На основе таблицы и анализа делается итоговый вывод.
- Необходимо явно и четко ответить на вопросы

- Почему преимущества выбранной структуры максимально соответствуют сформулированным критериям, а недостатки — минимально критичны в контексте задачи?
- (если применимо) Почему не выбрана комбинация из структур, максимально покрывающих операции в контексте задачи?.

# Пример

Плохое обоснование: "хэш-таблица работает быстрее".

Хорошее обоснование (краткий пример):

- \* Выбрана хеш-таблица с открытой адресацией, поскольку основная операция в задаче — ..., что позволяет ....
- \* Операция ... из-за отсутствия встроенной упорядоченности / диапазонных запросов будет производиться за счет ...
- \* Ожидаемый размер данных (до 10 000 записей) позволяет выбрать ....

## Порядок защиты курсовой работы

Защита курсовой работы состоит из двух этапов: решение билета на защиту и беседа с преподавателем. На решение билета студенту отводится 40 минут. Билет на защиту содержит (билеты):

- задачу на программирование (модификация существующего кода курсовой работы студента или другая смежная задача в зависимости от выбранной темы)
- теоретические вопросы по теме курсовой работы студента

В случае, если студент не выполнил часть билета на защиту (например, одно из заданий: задача / теор. вопросы), он может быть допущен ко второму этапу защиты со снижением максимально возможной оценки за защиту (на 1 балл). Невыполнение обоих пунктов билета на защиту означает прекращение защиты и оценку «неудовлетворительно».

В результате беседы с преподавателем, ответов на вопросы по билету на защиту, исходному коду и теме курсовой работы оценка студента может быть снижена (на 1 балл). После беседы с преподавателем студент получает максимальную возможную оценку с учётом всех её снижений.

Студент получает максимальную оценку согласно выбранному уровню курсовой работы, если выполнены все следующие условия:

- Студент выполнил полностью задание на защиту, программа запускается и демонстрирует корректность работы на тестах, предложенных студентом и преподавателем. Задание было успешно встроено в курсовую работу.
- Студент верно отвечает на все вопросы преподавателя.
- Студент способен прокомментировать свою работу как с точки зрения логики её выполнения, так и с точки зрения использования конструкций языка.

Основания для снижения оценки во время беседы с преподавателем:

- Задание на программирование выполнено полностью, но корректный результат работы программы показывает только на нескольких тестах.

- Студент совершает более двух ошибок при ответе на вопросы.
- Студент не может прокомментировать логику работы своей программы и/или дать комментарии по программному коду. Неправильно данные ответы приравниваются к неспособности студента дать комментарии.

From:

<https://se.moevm.info/> - МОЭВМ Вики [se.moevm.info]



Permanent link:

[https://se.moevm.info/doku.php/courses:algorithms\\_structures:cw](https://se.moevm.info/doku.php/courses:algorithms_structures:cw)

Last update: