

Регламент экзамена



FAQ об экзамене

Система оценивания

Полученные в рамках семестра баллы за компоненты курса, описанные в рейтинговой системе, конвертируются в оценку за экзамен по следующим правилам:

Таблица 1. Границы баллов для перевода в экзаменационную оценку.

Оценка	Граница баллов
Удовлетворительно	≥ 55
Хорошо	≥ 75
Отлично	≥ 95

Подтверждение оценки

Любая оценка за экзамен, получаемая по рейтингу, кроме «Неудовлетворительно», требует подтверждения на экзамене.



- **Отказ от подтверждения оценки** соответствует **отказу от оценки по рейтингу**, т.е. решение полного билета.

Повышение оценки

Если студента не удовлетворяет оценка по итогам работы в семестре (на основании баллов), он может отказаться от нее и сдать экзамен



- накопленная за семестр оценка **теряется без возможности отката**
- экзамен представляет из себя полный экзаменационный билет по темам дисциплины

Допсессии, комиссии, дни качества и прочее

Допсессия и комиссия

Для студентов, **несдавших экзамен в рамках основной сессии** (т.е. получивших «Неудовлетворительно» при сдаче экзамена):

- накопленная за семестр **оценка обнуляется** (поскольку студент не смог её подтвердить на экзамене)
- экзамен проводится **по правилам полного экзамена**

Для студентов, недопущенных до экзамена (например, несдача курсовой работы) или пропустивших экзамен **по уважительной причине**:

- накопленная за семестр **оценка сохраняется**
- экзамен проводится по стандартным правилам:
 - все положительные оценки требуют подтверждения путём решения упрощенного экзаменационного билета
 - в случае оценки «Неудовлетворительно» по рейтингу или повышения оценки - решение полного экзамена

День качества

Курсовая работа

Повышение оценки за курсовую происходит путём выполнения нового варианта курсовой на соответствующую оценку.

- Для получения нового варианта необходимо написать своему преподавателю.

Экзамен

Повышение оценки в рамках дня качества соответствует правилам, описанным в разделе "Повышение оценки"

Экзаменационные билеты

В данном разделе описаны виды и составляющие экзаменационных билетов.
Примеры билетов приведены в конце страницы.



Каждый компонент билета выполняется студентов **самостоятельно без использования каких-либо сторонних ресурсов и материалов**.

Упрощенный билет

Подтверждение оценок на экзамене предполагает упрощенный экзаменационный билет, состоящий из теоретического минимума и практических задач, и беседу с преподавателем.

- Оценивание решений теор. минимума и задач проводится до или во время беседы с преподавателем.
- Беседа с преподавателем не даёт возможности исправить/изменить решения практических задач или получить время на доработку

Теоретический минимум

- является **первым и обязательным этапом экзамена** для подтверждения любой оценки
- представляет из себя тест из 20 вопросов (с выбором ответа или свободным ответом)
- выполняется в течение 15 минут

Если результат студента по теоретическому минимуму меньше 50% - **оценка за экзамен снижается на 1 балл**



- При подтверждении оценки «Удовлетворительно» в случае несдачи теор. минимума студент получает «Неудовлетворительно»

Практические задачи

- Подтверждение оценки «Удовлетворительно» - решение одной практической задачи
- Подтверждение оценки «Хорошо» - решение двух практических задач
- Подтверждение оценки «Отлично» - решение двух практических задач

Подтверждение более высокой оценки автоматически включает задачи более низких оценок. Таким образом, для подтверждения оценки «Отлично» требуется решить 5 задач, для «Хорошо» - 3 задачи.



Время, отведенное на решение практических задач, зависит от подтверждаемой оценки (т.е. от кол-ва задач)
- 30/60/90 минут для оценок «Удовлетворительно»/«Хорошо»/«Отлично» соответственно.



Полный билет

Полный билет содержит

- теоретический анализ
- расширенный теоретический минимум (30 вопросов со свободным ответом на 30 минут)
 - Примечание от 14.01: вопросы со свободным ответом заменены на утверждения, которые нужно отметить как верные / неверные - время на компоненту снижено до 25 минут.
- практические задачи - по 2 задачи на каждую оценку (Удовлетворительно/Хорошо/Отлично)
 - более высокая оценка автоматически включает задачи более низких оценок (т.е. на оценку «Отлично» нужно решить 6 задач, на «Хорошо» - 4 задачи)
 - на решение всех задач (независимо от желаемой оценки) отводится 90 минут

Теоретический анализ

За **два дня** до экзамена* студенту, изъявившему желание писать полный экзамен, случайным образом выдаются **3 вопроса** по теоретическому материалу дисциплины:

- Кол-во вопросов может быть снижено до 2, если студент не претендует на оценку выше «Удовлетворительно»
 - На оценку «Хорошо» или «Отлично» количество вопросов = 3
- Студент самостоятельно готовится по данным вопросам, чтобы **на экзамене** предоставить **максимально полные ответы** по ним
 - На экзамене для написания ответов отводится 40 минут
- Если на экзамене дано корректных ответов менее, чем на 1.5 вопроса - ставится оценка «**Неудовлетворительно**».
 - В остальных случаях студент допускается до следующих этапов экзамена.
 - Результаты данного этапа могут (и будут) влиять на итоговую оценку за экзамен в спорных и иных случаях.

Полный список вопросов теор. анализа приведен на странице [Вопросы теоретического анализа](#).

Теоретический минимум и практические задачи

Аналогично разделу «Подтверждение оценки», отличие только в количестве вопросов / задач (описаны ранее).

Пример компонентов экзаменационного билета

Теоретический анализ

1. (вопрос №1) Сделайте сравнительный анализ O и Θ -нотаций, включая их формальные математические определения.
 - Объясните, почему в практике программирования и при проектировании систем наибольшее распространение получила O -нотация, несмотря на её «пессимистичность».
 - Проиллюстрируйте свой ответ примером алгоритма, для которого O , Ω и Θ -оценки различны, и объясните, какие практические выводы для разработчика из этого следуют.
2. (вопрос №2) Проведите детальный сравнительный анализ временной и пространственной сложности для односвязных, двусвязных и развернутых списков.
 - Для каждой структуры данных: постройте таблицу сложности операций (доступ по индексу, поиск, вставка/удаление в начало/середину/конец) с указанием лучшего, худшего и среднего случаев.
 - На основе анализа предложите оптимальную структуру данных для каждого из сценариев:
 - Частые случайные доступы по индексу, последовательный обход
 - Непредсказуемый паттерн доступа с преобладанием операций в середине структуры.
3. (вопрос №3) Развёрнутый связный список представляет собой гибридную структуру, сочетающую преимущества массивов и связных списков. Дайте строгое математическое доказательство теоремы об оптимальном размере блока $k = \Theta(\sqrt{n})$, включая анализ функции $f(k) = n/k + k$ и нахождение её минимума.
 - Объясните, почему на практике выбирают фиксированные значения $k = 16-64$, а не вычисляют \sqrt{n} динамически. Какие архитектурные особенности современных процессоров влияют на этот выбор?
 - Предложите модификацию развёрнутого списка, которая сохраняла бы асимптотику $O(\sqrt{n})$ для операций, но улучшала производительность для паттерна доступа «частое добавление в начало». Обоснуйте эффективность вашего решения.
4. (вопрос №4) На примере сортировки Шелла и TimSort
 - представьте анализ сложности сортировки Шелла, включая: формальную запись теоремы Ханта о среднем числе инверсий в h -упорядоченной перестановке.
 - для TimSort дайте строгое обоснование выбора параметра $minrun$ и его оптимизации в диапазоне [32, 64]. Алгоритма балансировки слияния подмассивов с поддержанием инвариантов стека.
 - Сравните практическую эффективность этих алгоритмов для различных типов данных
5. (вопрос №5) Проведите сравнительный анализ трех классических жадных алгоритмов
 - Для дробного рюкзака докажите оптимальность стратегии удельной стоимости, используя выпуклость целевой функции и линейность ограничений

- Для кодирования Хаффмана докажите оптимальность через индукцию по размеру алфавита, обосновав, что объединение двух наименее вероятных символов сохраняет оптимальность.

Теоретический минимум

- Какова временная сложность поиска элемента в отсортированном массиве с использованием двоичного поиска?
- Какую структуру данных лучше всего использовать для реализации очереди и почему?
- Приведите примеры устойчивых сортировок?
- Для какого случая хеш-таблицы эффективны?
- Что означает «балансировка» в АВЛ-дереве?
- Какова разница между обходом дерева в глубину и в ширину?
Приведите пример для объяснения.
- Что такое хеш-функция? Каковы её основные свойства?
- Какие проблемы могут возникнуть при использовании хеш-функций?
Опишите способы их решения.
- Какой из алгоритмов сортировки основан на подходе «Разделяй и властвуй»?
- Что происходит при большом повороте в АВЛ-дереве?
- Что представляет собой стек?
- Какая из структур данных подходит для реализации алгоритма обхода дерева в ширину?
- Приведите примеры трёх жадных алгоритмов и их критерий жадности
- Опишите основные свойства красно-черного дерева
- Какую временную сложность имеет операция вставки в куче?
- Какие структуры данных не поддерживают произвольный доступ к элементам?
- Какие преимущества имеет использование самобалансирующихся деревьев? Приведите примеры применения.
- Опишите алгоритм работы сортировки Timsort. В чем его преимущества по сравнению с другими сортировками?

Практические задачи

Пример «билета» с набором задач на каждую оценки (более высокая оценка включает задачи более низких оценок).

Задачи выполняются строго **последовательно**.

Удовлетворительно

1. Реализуйте класс бинарного дерева Tree с использованием класса узла Node и функцию для нахождения путей от корня до листа длины в диапазоне [a, b], за один обход дерева. Проверьте работу функции на различных конфигурациях деревьев. Обоснуйте и подтвердите сложность алгоритма (график теор. и практич. времени)

Хорошо

1. Проверьте за один проход, является ли бинарное дерево 1) линейным

- списком с диапазоном значений [c, d]. 2) АВЛ-деревом высотой больше A и меньше B.
2. Реализуйте модифицированный односвязный и двусвязный списки, позволяющие за O(1) получить минимальное/максимальное значение в списке, с сохранением асимптотики операций. Подтвердите сложность операций, сравнив с простыми списками

Отлично

1. Реализуйте два алгоритма сортировки: слиянием и вставками. Сравните их производительность, а также производительность встроенной сортировки, на случайных, почти отсортированных и обратно отсортированных массивах.
2. Реализуйте хеш-таблицу с методом цепочек, вместо цепочек - любой класс с методами add/remove/find (по умолчанию - односвязный список). Подберите различные конфигурации данных / хеш-функции и сравните скорость работы с использованием цепочки 1) по умолчанию 2) бинарного дерева поиска

From:

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



Permanent link:

https://se.moevm.info/doku.php/courses:algorithms_structures:exam

Last update: