* + - 1. Что такое комбинаторика?

**Комбинаторика** — это раздел математики, изучающий **методы подсчёта, построения и анализа различных способов выбора и расположения элементов в множествах**.

Она отвечает на вопросы:

* **Сколько существует способов выбрать m элементов из n?**
* **Сколько существует способов расположить элементы в определённом порядке?**
* **Сколькими способами можно объединить или разделить объекты?**

**🔢 1️⃣ Основные объекты комбинаторики**

В комбинаторике есть **три главных типа выборок элементов:**

1. **Перестановки** (**P(n)**) – когда порядок важен, все элементы участвуют.
2. **Размещения** (**A(n, m)**) – когда порядок важен, но выбираем только m элементов.
3. **Сочетания** (**C(n, m)**) – когда порядок НЕ важен, но выбираем m элементов.

Размещения (A(n, m)) – порядок важен, но выбираем m из n

Сочетания (C(n, m)) – порядок НЕ важен, выбираем m из n

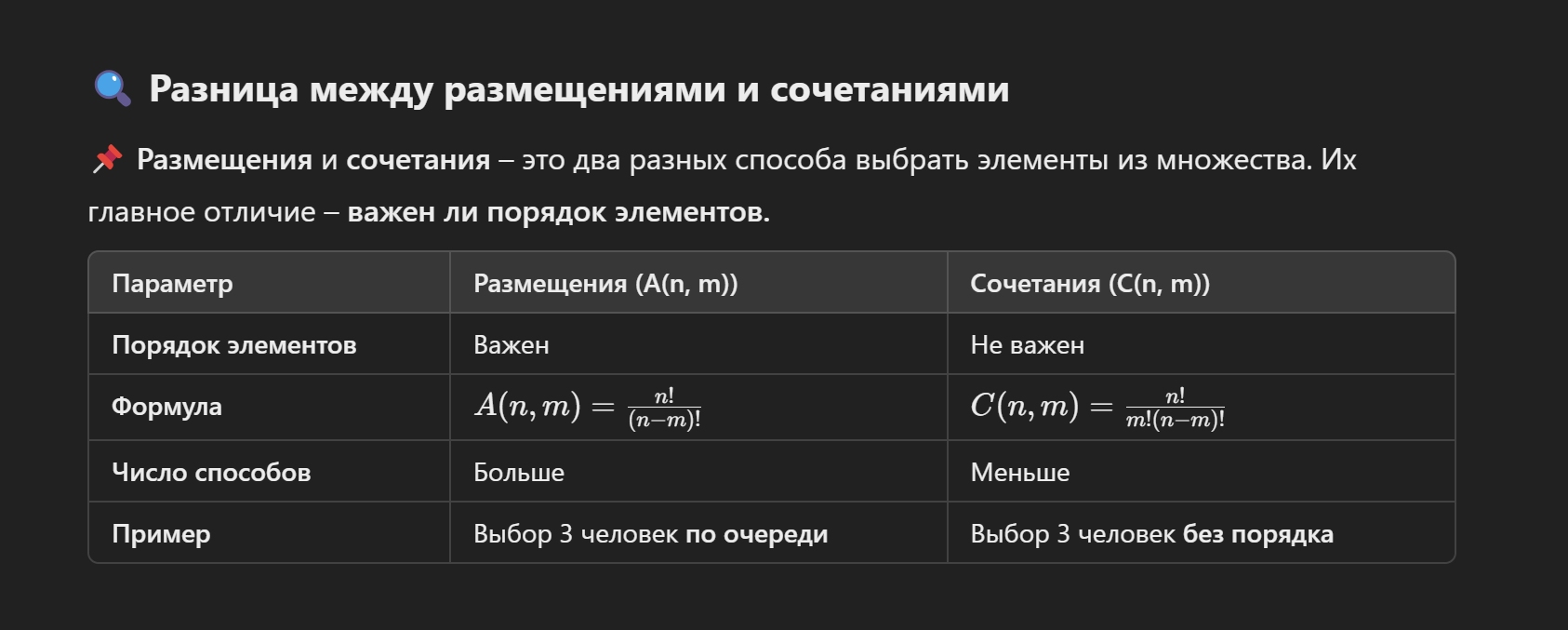
* + - 1. Что такое генератор?

**Генератор в программировании**

**Генератор** – это **специальная функция или объект**, который позволяет **пошагово (лениво) вычислять и возвращать значения по требованию**, вместо того чтобы вычислять их все сразу.

🔹 **Главные особенности генераторов:**

1. **Они не возвращают сразу все результаты, а вычисляют их по мере необходимости.**
2. **Позволяют экономить память, так как не создают весь список значений сразу.**
3. **Используют yield (в языках, где он поддерживается), а не return, чтобы сохранять состояние.**
4. **Они являются итераторами – можно перебирать значения с помощью for или next().**
   * + 1. Чем размещения отличаются от сочетаний?



* + - 1. Каких подмножеств множества любой размерности всегда по одному?

В любом множестве **всегда есть два уникальных подмножества**, которые присутствуют независимо от размера множества:

1. **Пустое множество {}** – всегда одно.
2. **Само множество {A, B, C, ...}** – всегда одно.

**📌 Объяснение**

* **Пустое множество** {} (∅) – это подмножество любого множества, содержащее **0 элементов**.
* **Полное (само) множество** {A, B, C, ...} – это подмножество, содержащее **все элементы исходного множества**.

**устое множество {}** – всегда одно.  
✔ **Полное множество {A, B, C}** – всегда одно

* + - 1. С помощью какого генератора решается задача о загрузке судна?

**Решение строится на основе генератора сочетаний (C(n, m))**.

**Почему сочетания, а не размещения?**

* Важно **выбрать m контейнеров из n**, но **порядок их размещения в судне не имеет значения**.
* Используется **перебор всех возможных сочетаний m контейнеров** и выбор наиболее выгодного набора.
  + - 1. С помощью какого генератора решается задача о рюкзаке?

Для решения задачи о рюкзаке используется генератор всех подмножеств множества

**Генератор всех подмножеств (2^n)**

* **Задача о рюкзаке** требует **перебора всех возможных комбинаций предметов**, так как **каждый предмет можно включить или не включить**.
* **Число возможных подмножеств** – **2^n**, так как **каждый предмет имеет два состояния (взять или не взять).**
  + - 1. С помощью какого генератора решается задача о коммивояжере?

## **Какой генератор нужен?**

### ✅ ****Генератор перестановок (****permutation****)****

Для задачи о коммивояжере **используется генератор перестановок**, так как:

1. **Порядок посещения городов имеет значение** → мы ищем **оптимальную перестановку** маршрута.
2. **Перебираем все возможные маршруты (перестановки городов)**.
3. **Выбираем маршрут с минимальной длиной (стоимостью пути).**

✔ **Генератор перестановок позволяет проверить все возможные маршруты.**