

Лекция №15  
по дисциплине  
«ТЕОРИЯ АЛГОРИТМОВ»

# СТРУКТУРЫ ДАННЫХ

Преподаватель:  
Золотоверх Д.О.

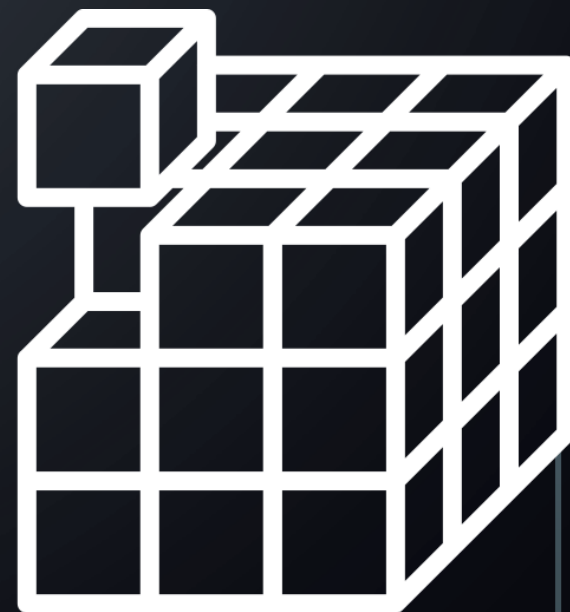
# СТРУКТУРА ДАННЫХ

Способ **организации** информации: ее **формат** хранения, **способы изменения** и **доступ**.

Такой информацией может быть **совокупность** и **однотипных, и многотипных данных**.

Данные могут быть **связаны**.

Управление данными предоставлен **определенным способом**.



# СТРУКТУРА ДАННЫХ VS ТИП ДАННЫХ

Тип данных:

- привязан к определенному способу хранения;
- хранит значение, не информацию;
- значение можно изменить прямым способом (через присвоение);
- проблема сложности вычислений не затрагивается;

100  
1010  
01

# ПРИМЕРЫ ТИПОВ ДАННЫХ

- целочисленный `int`;
- булевой `bool`;
- дробный `float`;
- символьный `char`;
- их вариации (целочисленные `short` и `long`, дробный двойной точности `double` и буквенный стандарта UNICODE `char32_t` и т.д.)

100  
1010  
01

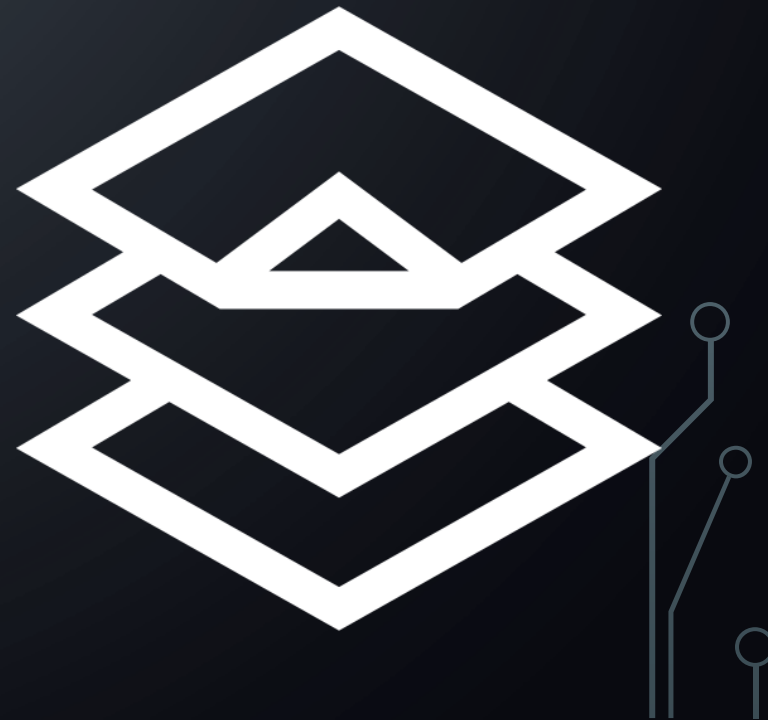
# КОМПОЗИТНЫЙ ТИП

Существует случай, когда способ организации **не дотягивает** до структуры данных (например модификация может быть выполнена на прямую), **но он вмещает в себя несколько типов данных.**

Это называется **КОМПОЗИТНЫМ ТИПОМ** или же **сложный типом.**

Примеры:

- Обычный **список**;
- Структура **struct** (record).



# СТРУКТУРА ДАННЫХ VS ТИП ДАННЫХ

Структура данных:

- имеет внутреннюю форму, данные могут быть связаны;
- может иметь несколько разных типов данных;
- хранит информацию об предмете;
- значение нельзя изменить прямым способом, только с помощью специальной операции;
- нужно учитывать проблему сложности вычислений.



# ПРИМЕРЫ СТРУКТУР ДАННЫХ

Примеры структур данных:

- **Стек** (определенный порядок доступа и модификации);
- **Очередь** (схож в реализации со Стекком);
- **Связанный список** (массив, но элементы связаны опр. способом);
- **Множество** (может хранить только уникальные элементы);
- **Хеш-таблица** (пара ключ-значение);
- **Дерево** (древовидная структура связанных элементов);
- **Граф** (связанное множество).



# АБСТРАКЦИЯ

Структуры данных являются более абстрактными сущностями, чем массивы и типы данных.

Они определяются, прежде всего, своим интерфейсом: набором разрешенных операций, которые могут выполняться с ними.

Интерфейс этих структур проектируется с расчетом на поддержку ограничений доступа.

Базовый механизм, используемый для их реализации, обычно остается невидимым для пользователя.





СПАСИБО ЗА ВНИМАНИЕ!

