**Конспект занятия от 02.11.2024**

# Техническое мышление – из каких навыков оно состоит?

* Знание алгоритмов;
* Умение структурировать информацию;
* **Декомпозиция** – разбить большую задачу на несколько меньших;
* Выделение главного;
* Не думать “В лоб” (иногда всё же полезно);
* **Абстракция** – умение упрощать задачу;

# Уровни языков (от низкого к высокому)

**Hardware -> Machine Language (Машинный Код) -> Assembly Language (Язык Ассемблера) -> High Level Language (ЯП высокого уровня)**

Основная черта высокоуровневых языков — это абстракция, то есть введение смысловых конструкций, кратко описывающих такие структуры данных и операции над ними, описания которых на машинном коде (или другом низкоуровневом языке программирования) очень длинны и сложны для понимания.

# Типы языков программирования (по виду программирования)

* **Структурное**
* **Процедурное**
* **Объектно-ориентированное**
* **Функциональное**

# Работа с объектами (пример)

Console.WriteLine(“Hello, World!”); - Эта строка кода вызывает статический метод **WriteLine**

# Примеры абстракции

1. Голубое озеро в городе Казань -> озеро в городе Казань -> водоём в г. Казань -> озеро в Республике Татарстан -> водоём в Республике Татарстан -> озеро на территории РФ -> водоём на территории РФ -> водоём;
2. Сковородка Gipfel для индукционной плиты -> сковородка Gipfel для плиты -> сковородка Gipfel -> одна единица посуды Gipfel -> сковородка -> посуда для приготовления пищи;
3. Пирамида Хеопса -> одно из семи чудес света -> пирамида в Египте -> древнее египетское сооружение -> большое по размерам сооружение в Египте -> большое по размерам сооружение в африканской стране -> большое сооружение;
4. Карета скорой помощи -> служебная машина с опознавательными знаками и мигалкой -> служебная машина, имеющая опознавательные знаки -> служебная машина -> автомобиль -> транспортное средство;

# .Net, C#

Исходный код -> (работа компилятора) -> CIL код(байт-код) -> (работа CLR) -> Native код;

CIL-код исполняется на любой платформе, где есть CLR. У каждого ЯП, поддерживающего .NET, свой собственный компилятор, который компилирует исходный код в CIL-код.

**Примечание.** Мультиязычность != Кроссплатформенность

# Понятия Сборка/Решение/Проект

* **Решение(solution)** – большой контейнер для проектов, в С# - файл с расширением “.sln”;
* **Проект** – одна программная единица, которая может работать (или быть использована другими проектами) автономно или вместе с другими проектами;
* **Сборка(assembly)** – готовый к “употреблению” проект. Сам по себе проект не работает, его надо “собрать”;

Решение содержит информацию о всех проектах, а у каждого проекта есть своя сборка.