

Dane mają/mogą mieć różne postacie czy rozmiary, ale mogą być zorganizowane podobnie.

Przykład: a) lista rzeczy do zrobienia na jutro b) lista składników pieczonego ciasta itp.

Lista to prosty przykład struktury danych, są oczywiście inne struktury danych: listy jedno czy dwukierunkowe, stosy, kolejki, zbiory, tablice asocjacyjne, drzewa, sterty, kolejki priorytetowe, grafy.

Za korzystaniem z struktur danych przemawiają:

- **szybkość działania;** struktury danych korzystniej układają dane; przemyśleli ich budowę matematycy (stosowani);
- **ogólność;** struktury danych pozwalają na zrozumienie natury danych; np. jeśli korzystamy z umieszczania danych na stosie, to koncentrujemy się na tym, jak stosów się używa, jak zbudować operacje umieszczania na stosie i zdejmowania ze stosu; czyli struktury danych pozwalają na omawianie programu z mniej programistycznego punktu widzenia (możliwa jest abstrakcja napotkanego problemu);
- **możliwość wielokrotnego wykorzystania;** struktury danych są niezależne od kontekstu, gdyż można ich używać do danych dowolnych typów oraz w różnych sytuacjach.

Dla struktur danych charakterystyczne są operacje (metody), które mogą być na nich wykonywane.

Np. dla listy konieczne jest wstawianie, usuwanie, przeglądanie, zliczanie elementów.

Algorytmy to jednoznacznie zdefiniowane procedury pozwalające rozwiązywać pewne problemy. Na przykład problem sortowania danych z uwagi na jakąś ich cechę. Za ich wykorzystaniem ponownie przemawia:

- **szybkość działania;** (zagadnienie: jak dobrać właściwy algorytm albo zbudować go częściami z algorytmów istniejących?...albo wymyślić własny?);
- **ogólność;** algorytmy umożliwiają potraktowanie sformułowanego zadania w oderwaniu od konkretnego języka programowania; zwykle złożony problem można podzielić na części, dla których już istnieją skuteczne algorytmy;
- **możliwość wielokrotnego wykorzystania;** o skuteczności algorytmu decyduje nie tyle typ danych lecz powiązania między danymi, czyli rodzaj struktury danych do której rozpatrywane dane udaje się sprowadzić; również algorytmy są w bardzo dużym stopniu niezależne od kontekstu, tzn. bardzo często daje się je wykorzystać w różnych sytuacjach.

Algorytmy zwyczajowo dzieli się na klasy (podział dość nieprecyzyjny).