

2.1 Języki Programowania.

W pamięci program jest zapisany jako **Kod Maszynowy**, czyli instrukcje dla konkretnego procesora. Są one pamiętane jako liczby w systemie binarnym (zerojedynekowym).

Powstał więc **Asembler** - język pierwszej generacji. Jego rozkazy odpowiadają bezpośrednio instrukcjom procesora, jednak zapisywane są w bardziej zrozumiałej formie.

Druga generacja to języki nie mające już tak bliskiego związku z procesorem, ale nadal podobnie jak assembler stanowiące jeden ciąg instrukcji. Są to tak zwane języki interpretowane. Przykładem takiego języka może być ogólnie znany **Basic**. Jedna instrukcja języka drugiej generacji, może być tłumaczona na kilka, kilkadziesiąt a czasem więcej instrukcji procesora.

Języki kolejnej generacji, np. **Pascal**, rozwijają koncepcję programowania o możliwość podzielenia programu na moduły wykonujące różne operacje wchodzące w jego skład. Każda taka część stanowi pod pewnym względem zamkniętą całość i jej "wnętrze" nie jest dostępne. Wykorzystanie jej polega na przekazaniu parametrów (i ewentualnym odebraniu wyniku). Typowymi modułami są funkcje i procedury.

Kolejny to języki **Zorientowane Obiektowo (OO)**. Przykład to C++, Delphi. Chociaż w Pascalu też można pisać programy OO.

Powstały również języki **czwartej generacji**. Są to języki umożliwiające pisanie programów tzw. bardzo wysokiego poziomu, niezależne od sprzętu, a nawet systemu operacyjnego. Przykład to języki dostępu do baz danych.