

# Stopnie abstrakcji

- Asembler
- Języki strukturalne (C, Pascal, BASIC)
- Języki obiektowe (Simula, Smalltalk, C++, Java, C#)

# Simula-67

- Pierwszy język obiektowy
- Na bazie języka Algol
- Program jest symulacją
- Pierwszy wprowadził takie pojęcia jak klasy, obiekty, metody wirtualne, zbieranie nieużytków (garbage collection)

# OOP (Object-oriented programming)

- Obiekty, klasy
- Pola (właściwości obiektu)
- Metody (właściwości klasy)
- Dziedziczenie
  - Lista obiektów Kształt w C i C++, dziedziczenie klas Kształt, Okrąg, Prostokąt, Kwadrat
- Polimorfizm
  - Metody UstawKolor() vs. Przesun()/Rysuj()
- Tworzenie i usuwanie obiektów; zarządzanie pamięcią przez program vs. garbage collection

# Etapy OOP

1. Wyszukanie (zdefiniowanie) obiektów.
2. Budowa obiektów.
3. Budowa systemu z obiektów.
4. Ewolucja systemu.
5. W miarę możliwości wykorzystanie obiektów w innych projektach.

# Smalltalk-80

- Pierwszy w pełni obiektowy język (object oriented)
- Cechy Smalltalk (wg. Alana Kay):
  - wszystko jest obiektem
  - program jest zbiorem obiektów przesyłających do siebie komunikaty
  - każdy obiekt posiada własne dane, które też są obiektami (nowy obiekt składamy z wielu starych)
  - każdy obiekt ma typ (jest instancją klasy)
  - wszystkie obiekty danego typu mogą otrzymywać te same komunikaty

# C++

- 1979 – „C with classes”, 1983 - „C++”, obecny standard ISO/IEC 14882:2003
- Kompatybilny z C
- Kompilowany do kodu maszynowego
- Zwykle wykonanie prawie tak szybkie jak w C, tworzenie oprogramowanie zdecydowanie szybsze
- Dziedziczenie wielokrotne
- Programista zarządza pamięcią (brak garbage collector)

# Java

- Pierwsza wersja 1995
- Programy kompilowane do kodu pośredniego i wykonywane na Java Virtual Machine
- Wolniejszy od C++
- Bardziej bezpieczny od C++, ale nie pozwala użyć funkcji systemowych
- Zbieranie nieużytków (garbage collection)
- Dziedziczenie tylko po jednej klasie
- Działa na wielu systemach operacyjnych

# C# (C sharp)

- Pierwsza wersja 2001
- Programy kompilowane do kodu pośredniego (Common Intermediate Language) i wykonywane na platformie .NET
- Zbieranie nieużytków (garbage collection)
- Dziedziczenie tylko po jednej klasie
- Platforma .NET działa głównie na systemach firmy Microsoft; projekty Mono i DotGNU dla Linuxa

# Dlaczego C++?

- Wydajność
- Elastyczność
- Łatwa migracja z C
- Popularność (coraz mniejsza)