



AGH

Akademia Górniczo-Hutnicza
Wydział Elektrotechniki, Automatyki,
Informatyki i Inżynierii Biomedycznej



Adrian Horzyk

WSTĘP DO INFORMATYKI

HISTORIA INFORMATYKI





**Na planecie Ziemia
wiele milionów lat temu
narodziła się inteligencja,
która doprowadziła
do powstania informatyki.**

**Od tego czasu ludzie
programują komputery,
tworząc coraz bardziej
inteligentne aplikacje.**

**Studenci – nowi adeptci informatyki
specjalizują się w różnych
obszarach informatyki
tworząc nowy cyfrowy
porządek świata...**



Na początku była światłość



z której zrodziła się inteligencja...







OBECNIE



**ludzie rządzą światem, a ich wiedza
i inteligencja nieustannie rozwijają się...**



Inteligencja dalej ewoluuje



**aż do momentu, w którym obdarzone
inteligencją maszyny zwolnią ludzi
z konieczności myślenia...**



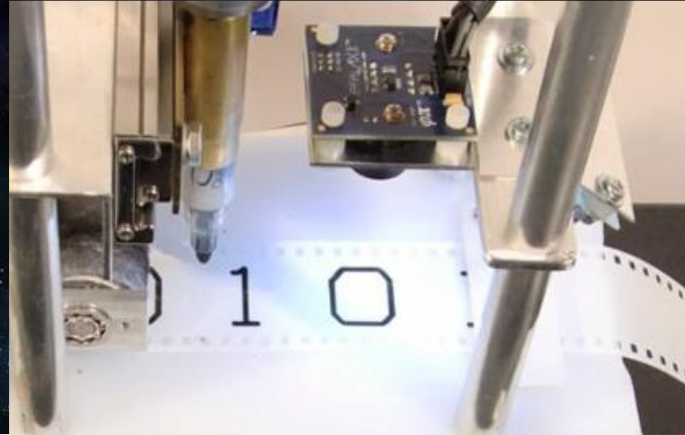
Nie!

Ten scenariusz zostawmy Hollywoodowi...

MASZYNA TURINGA



**Pierwszy
teoretyczny
abstrakcyjny
model
obliczeniowy
operujący na
symbolach...**



**Nieskończona
taśma zawiera
dane i instrukcje,
pod wpływem
których może się
przesuwać...**



MASZYNA TURINGA



Składa się z nieskończonej długości taśmy zawierającej pola, w których zapisuje się dane.

Każde pole może znajdować się w jednym z N stanów.

Maszyna zawsze jest ustawiona na jednym z pól i znajduje się w jednym z M stanów.

W zależności od stanu maszyny i zawartości pola maszyna zmienia stan na inny, a następnie może się przesunąć o jedno pole w lewo lub w prawo.

Taką operację nazywamy rozkazem.

Maszyna Turinga jest więc sterowana listą rozkazów.

Lista rozkazów tworzy program reprezentujący algorytm.

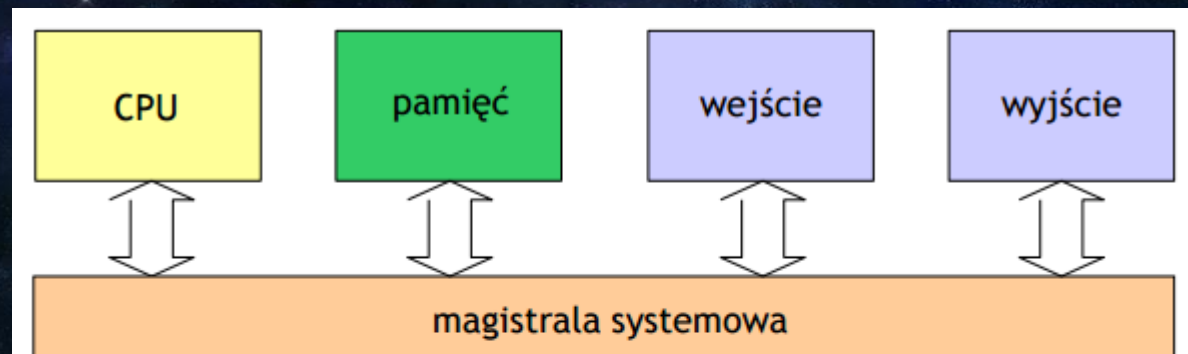
Współczesne komputery realizują model maszyny Turinga.

MASZYNA TURINGA



Stała się podstawowym modelem obliczeniowym stosowanym do dziś w większości sprzętu komputerowego, cyfrowego, procesorach...

Model ten został wykorzystany w pierwszej architekturze komputerów zaprojektowanej przez zespół Johna von Neumanna.



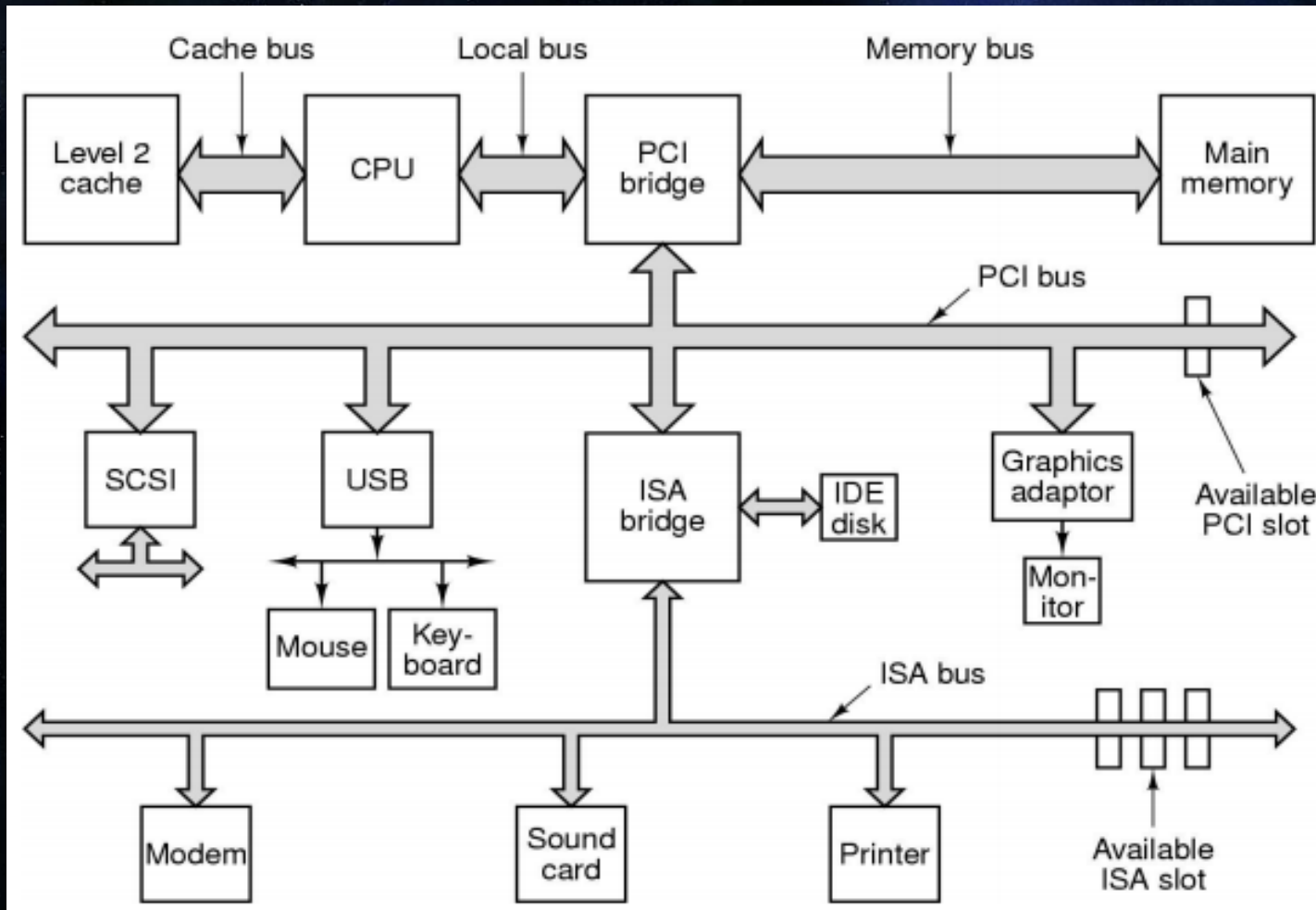
- CPU - jednostka centralna (procesor)
- pamięć - obszar przechowywania programu i danych
- wejście - urządzenia wejściowe
- wyjście - urządzenia wyjściowe



WSPÓŁCZESNE ARCHITEKTURY KOMPUTERÓW



Posiadają wiele magistral systemowych (bus) łączących ze sobą poszczególne podzespoły komputera:



JEDNOSTKA CENTRALNA CPU



Odpowiada za:

- Pobieranie rozkazów z pamięci do wykonania.
- Dekodowanie rozkazów i ich wykonywanie.
- Pobieranie danych z pamięci do przetworzenia.
- Zapisywanie wyników obliczeń (danych) do pamięci.
- Obsługę innych urządzeń wejścia/wyjścia poprzez magistrale systemowe za pośrednictwem przesyłanych danych, a czasami również rozkazów.

Zawiera operacje:

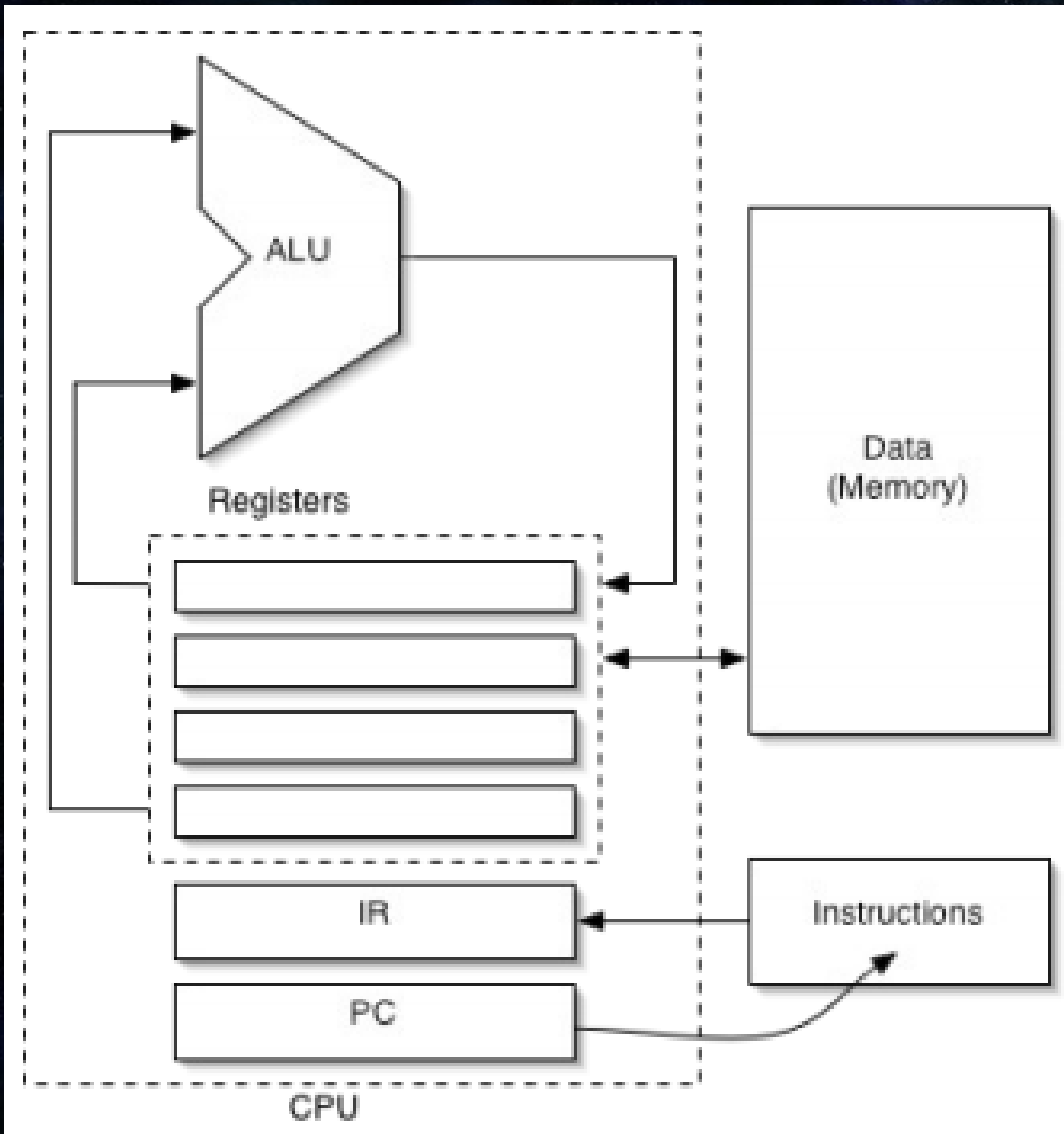
- Przesyłania danych pomiędzy rejestrami i pamięcią
- Arytmetyczno-logiczne wykonywane na rejestrach
- Sterujące wykonaniem programu (porównywanie, skoki (PC), przerwania)

Wykonanie pojedynczego rozkazu zajmuje od jednego do kilkunastu taktów zegara, które obecnie taktowane są zegarami GHz (10⁹ taktów na sekundę).

Neurony w biologicznym mózgu są zdolne maksymalnie do 100 aktywacji / s.

Współczesne procesory zawierają setki milionów tranzystorów i komunikują się z innymi elementami komputera za pośrednictwem zwykle 64-bitowych magistral systemowych.

JEDNOSTKA CENTRALNA CPU



Składa się z:

ALU – jednostki arytmetyczno-logicznej wykonującej operacje arytmetyczne i logiczne na rejestrach procesora, do których ładowane są z pamięci dane.

IR – rejestr rozkazów wczytywanych z pamięci wykorzystywany też do optymalizacji i przewidywania kolejności wykonywania kolejnych.

PC – licznik rozkazów wskazujący pozycję kolejnego rozkazu do wykonania w rejestrze rozkazów.

Dzień z życia procesora wygląda tak:

```
PC := 0;  
repeat
```

```
    rozkaz = pamiec[PC];  
    dekoduj (rozkaz);  
    pobierz (argumenty);  
    wynik = wykonaj (rozkaz);  
    zapisz (wynik);
```

```
until ( instrukcja = STOP )
```



ARCHITEKTURY JEDNOSTEK CENTRALNYCH CISC i RISC



CISC – Complex Instruction Set Computer:

- ✓ rozbudowany język rozkazów o różnej długości, nieraz bardzo złożonych,
- ✓ wykonywanych w różnej ilości cykli maszynowych,
- ✓ ułatwiających konstrukcję kompilatorów,
- ✓ redukujących ilość rozkazów kodu maszynowego,
- ✓ posiadających rozbudowane i elastyczne tryby adresowania danych.

RISC – Reduced Instruction Set Computer:

- ✓ prosty i oszczędny język rozkazów podstawowych o tej samej długości,
- ✓ wykonywane w jednym cyklu maszynowym,
- ✓ zawierający niewielką liczbę dostępnych trybów adresowania o prostej i przejrzystej strukturze,
- ✓ zapewniający wysoką wydajność przy niewielkim poborze energii, gdyż procesory RISC zawierają dużo mniejszą ilość tranzystorów niż CISC.



RÓWNOLEGŁOŚĆ



to powszechna cecha w przypadku biologicznych systemów, tj. układ nerwowy, mózg, komunikacja pomiędzy komórkami, osobnikami itp.

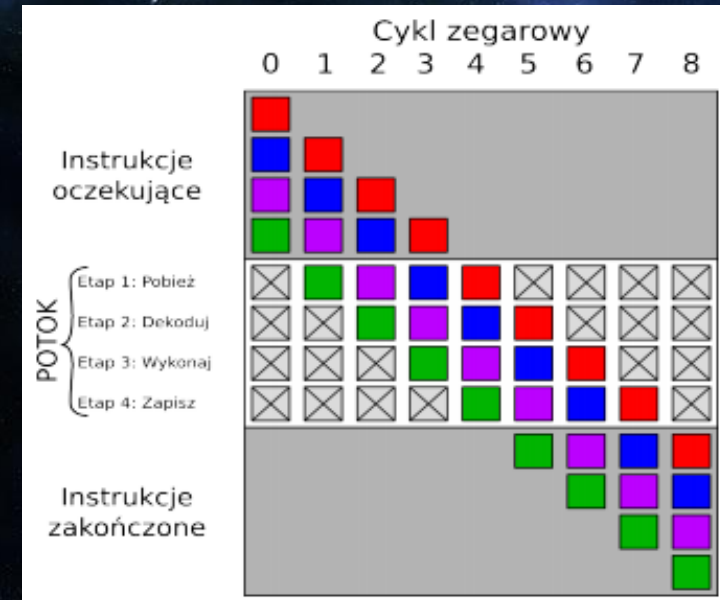
Komputery zbudowane na bazie **Maszyny Turinga** posiadają w stosunku do biologicznych rozwiązań poważną wadę – **sekwencyjność!**

Równoległość (zwykle synchroniczną) w można próbować wprowadzić w:

- ✓ Wykonywanie instrukcji (ILP – *Instruction Level Parallelism*),
- ✓ Przetwarzanie wielu danych tą samą instrukcją (DLP – *Data Level Parallelism*)
- ✓ Wykonywanie wątków (TLP – *Thread Level Parallelism*)

Przewarżanie potokowe (pipelining):

- ✓ Zmienia kolejność i zależności czasowe pomiędzy wykonywanymi rozkazami, gdyż kolejne instrukcje są przewidywane, ale nie zawsze przewidywanie wykonania kolejnych instrukcji jest poprawne.
- ✓ Umożliwia przyspieszenie wykonywania instrukcji w przypadku poprawnego przewidywania kolejnych rozkazów.





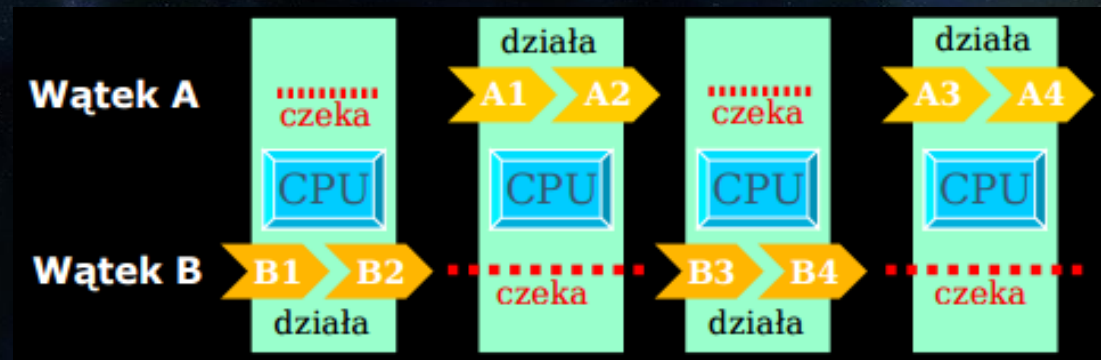
WIELOWĄTKOWOŚĆ



Istnieją też rozwiązania polegające na wektorowym lub macierzowym przetwarzaniu rozkazów, czyli ta sama instrukcja wykonywana jest wielu danych umieszczonych w wektorze lub macierzy (**SIMD** – *single instruction multiple data*).

Wielowątkowość:

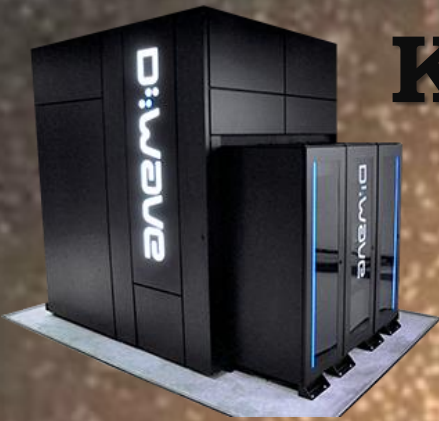
- ✓ Polega na rozbiciu przetwarzania zadania na wiele wątków, które mogą być wykonywane częściowo niezależnie i czasami w różnej kolejności, co umożliwia wykorzystanie wielu rdzeni procesora do przetwarzania różnych wątków programu równolegle.
- ✓ Warunkiem zastosowania wielowątkowości jest niezależność obliczeń wykonywanych w poszczególnych wątkach programu. W odwrotnym przypadku wykonanie wątków musi być zawieszane i oczekuje na wykonanie innych. Podobnie się dzieje w przypadku symulacji wielu wątków na procesorze jednordzeniowym:





KOMPUTERY KWANTOWE

Wykorzystują mechanikę kwantową w celu ewolucji układu kwantowego tak, aby wynik reprezentował rozwiązanie określonego zadania.



Jednostką obliczeniową jest bit kwantowy, tzw. **kubit**, który przyjmuje wartość z określonym prawdopodobieństwem.

Wartość tą nazywamy superpozycją 0 i 1.

Wyniki działania obliczeń kwantowych wyznaczamy na podstawie uśrednienia wartości z dużej serii obliczeń.

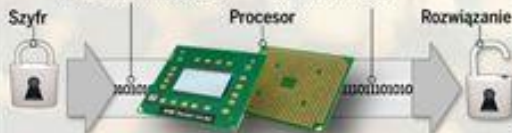
Algorytmy kwantowe są w stanie równolegle przeszukiwać przestrzeń wyników (wiele opcji na raz).

Dwie strategie łamania skomplikowanego szyfru

Komputer typu PC

Komputer otrzymuje skomplikowane dane wejściowe (szyfr)

Komputer sprawdza każdą z kombinacji znaków po kolei, jedna po drugiej

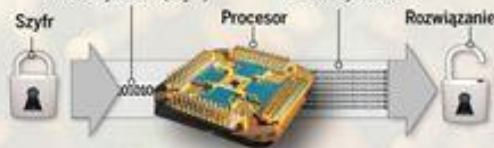


By sprawdzić wszystkie możliwe kombinacje, którymi szyfrowane są np. operacje bankowe, miliard komputerów o mocy miliarda operacji na sekundę potrzebowalby więcej czasu, niż istnieje wszechświat.

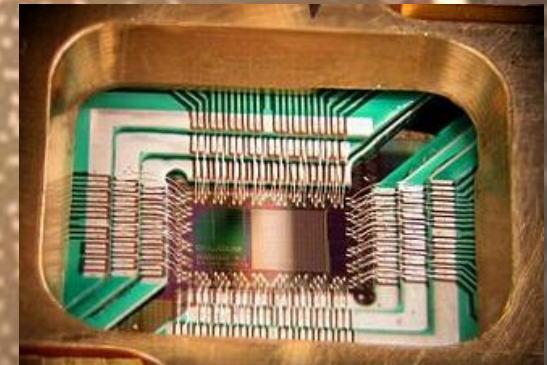
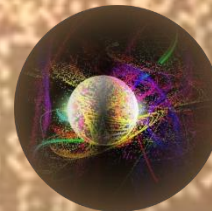
Komputer kwantowy

Komputer otrzymuje skomplikowane dane wejściowe (szyfr)

Komputer sprawdza jednocześnie wiele kombinacji znaków.



Złamanie skomplikowanego szyfru chroniącego połączenia z bankiem internetowym zajmuje komputerowi kwantowemu kilka sekund.





A CONNATTO BIOLOGIA?

Mózg



Jeszcze inny model przetwarzania danych!



MÓZG

OŚRODEK NASZEJ INTELIGENCJI



I SIEDLIŚKO NASZEJ WIEDZY
Jak działa ten dziwny galaretowany twór?

MÓZG – OŚRODEK NASZEJ INTELIGENCJI



ARCHITEKTURA

NEURONY

ZMYSŁY

POŁĄCZENIA

RECEPTORY



SYNAPSY

KOMÓRKI
GLEJOWE

PŁYN MÓZGOWO-RDZENIOWY
ZWIĄZKI PRZESTRZENNE I CZASOWE



WIEDZA?

CZYM JEST WIEDZA?



**Dane bombardują nas z każdej strony!
A nasze mózgi jakoś sobie z nimi radzą!**



**Wiedza jest aktywnym kontekstowym odwzorowaniem informacji skojarzonych w mózgu (sieci neuronowej).
Powstaje na skutek kojarzenia ze sobą faktów i reguł.**



DEFINICJA WIEDZY



- ✓ **Wiedza** – to „ogół wiarygodnych informacji o rzeczywistości wraz z umiejętnością ich wykorzystywania.” (Encyklopedia Powszechna)
- ✓ **Wiedza** jest wynikiem automatycznej **agregacji** powtarzających się **wzorców danych** oraz **asocjacyjnej konsolidacji** ich sekwencji w kontekście wcześniejszych danych, formujących reguły, metody i algorytmy przetwarzania danych.
- ✓ **Wiedza** jest ściśle powiązana z **inteligencją**, gdyż pozwala na wnioskowanie i rozwój inteligencji osobniczej poprzez tworzenie algorytmów/metod/reguł efektywnego postępowania, realizacji zadań i rozwiązywania problemów w kontekście znanych wzorców danych, które rozpoznaje i umie powiązać z innymi.
- ✓ **Inżynieria wiedzy** – zajmuje się zagadnieniami formowania, reprezentacji i wydobywania wiedzy, wykorzystując w tym celu dorobek informatyki, kognitywistyki, psychologii, filozofii, matematyki, fizyki i neurobiologii.



WIEDZA



- ✓ **Wiedza** nie jest więc dostępna czy też przekazywana bezpośrednio, lecz za pośrednictwem informacji przekazywanych poprzez różnego rodzaju fakty i reguły zrozumiałe dla odbiorcy w kontekście jego wiedzy.
- ✓ **Wiedza** jest zwykle wynikiem działania inteligentnego systemu posiadającego **system motywacji i potrzeby**, zdolnego dążyć do zdobywania informacji i dzięki **wewnętrznym mechanizmom asocjacji** reprezentować występujące w nich klasy obiektów kontekstowo względem siebie, co umożliwia ich odtwarzanie, przypominanie i generalizację, prowadząc do zwrotnego przekazu informacji wywołanych kontekstem pytania, stwierdzenia lub innego zdarzenia pobudzającego system posiadający wiedzę.
- ✓ Takim systemem zdolnym formować i przechowywać wiedzę jest np. nasz **ludzki mózg**, który dzięki licznym interakcjom ze światem zewnętrznym może poszerzać wiedzę.

MÓZG - SIECI NEURONÓW



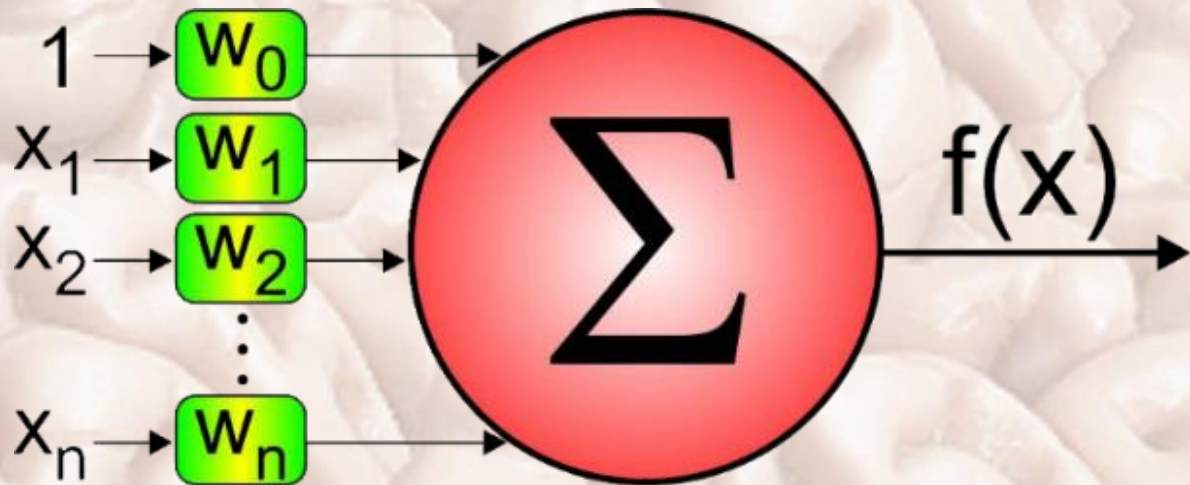
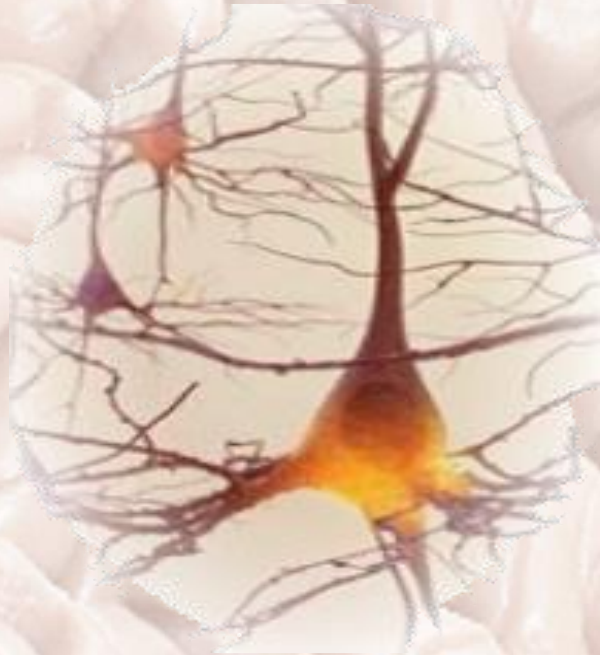
SIECI NEURONÓW



są w stanie błyskawicznie rozwiązać obliczeniowo bardzo skomplikowane zadania.



NEURONY BIOLOGICZNE I SZTUCZNE



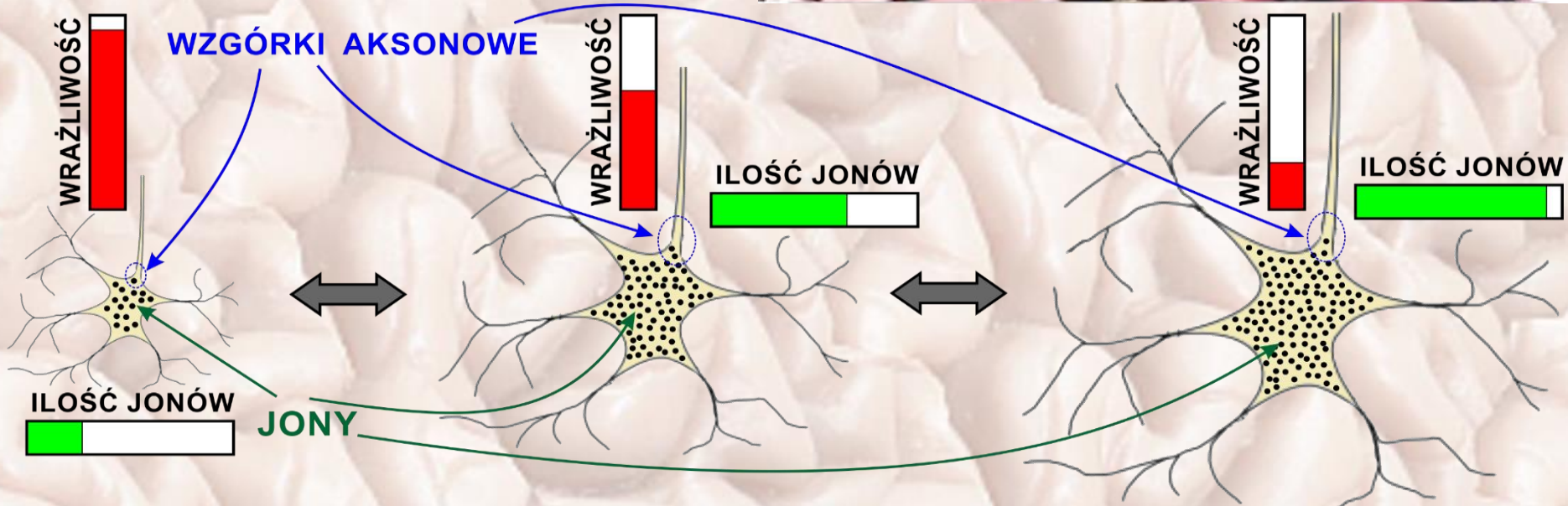
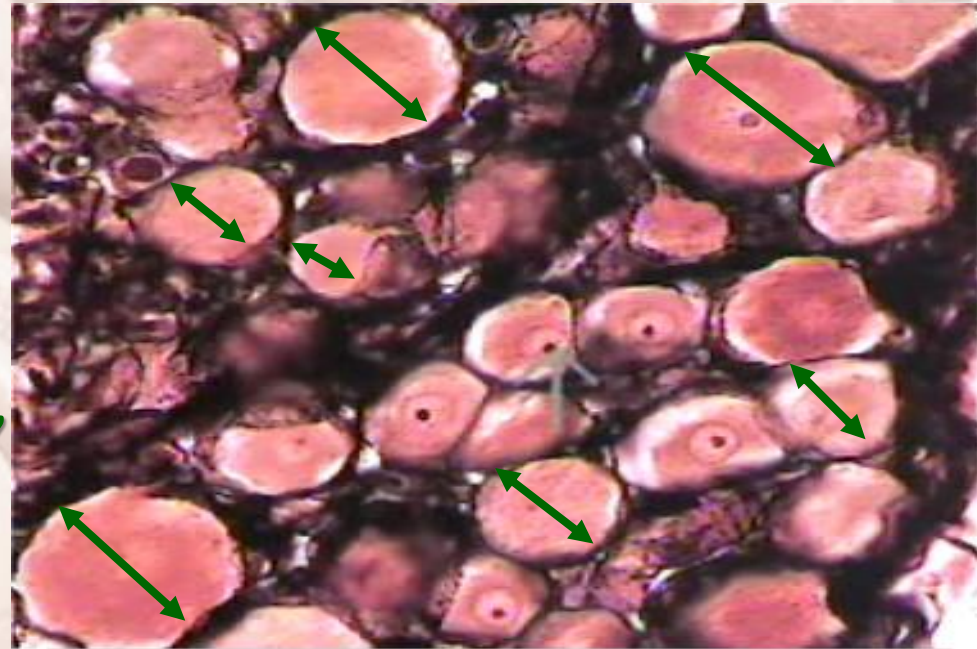
- Neurony sumują ważone sygnały wejściowe w czasie.
- Podlegają procesom relaksacji i refrakcji w czasie.
- Neurony posiadają progi aktywacji i swoją wrażliwość.
- Reprezentują różne kombinacje bodźców wejściowych.
- Definiują modele jako klasy obiektów.

WIELKOŚĆ I PLASTYCZNOŚĆ NEURONÓW



Neurony biologiczne różnią się:

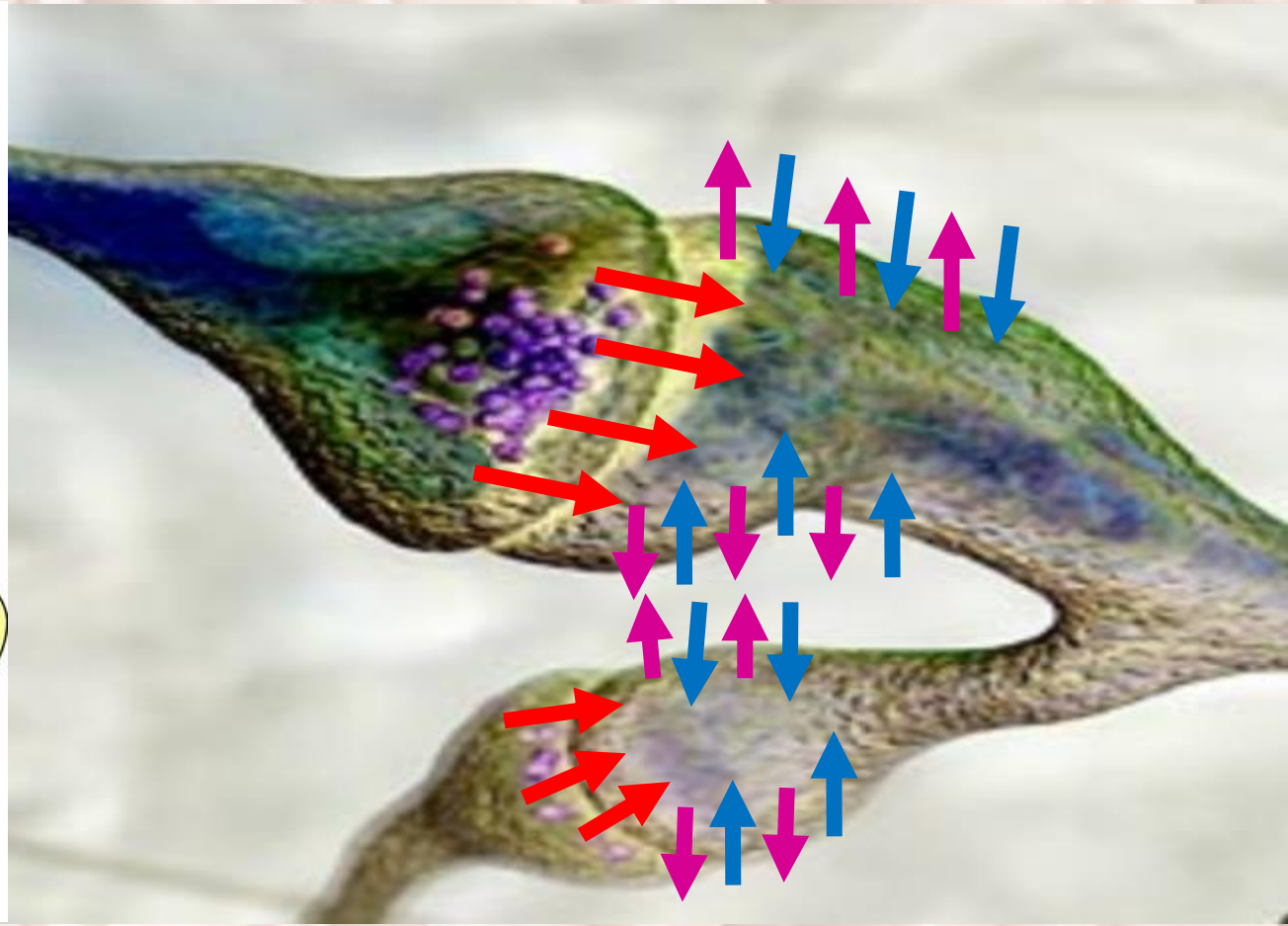
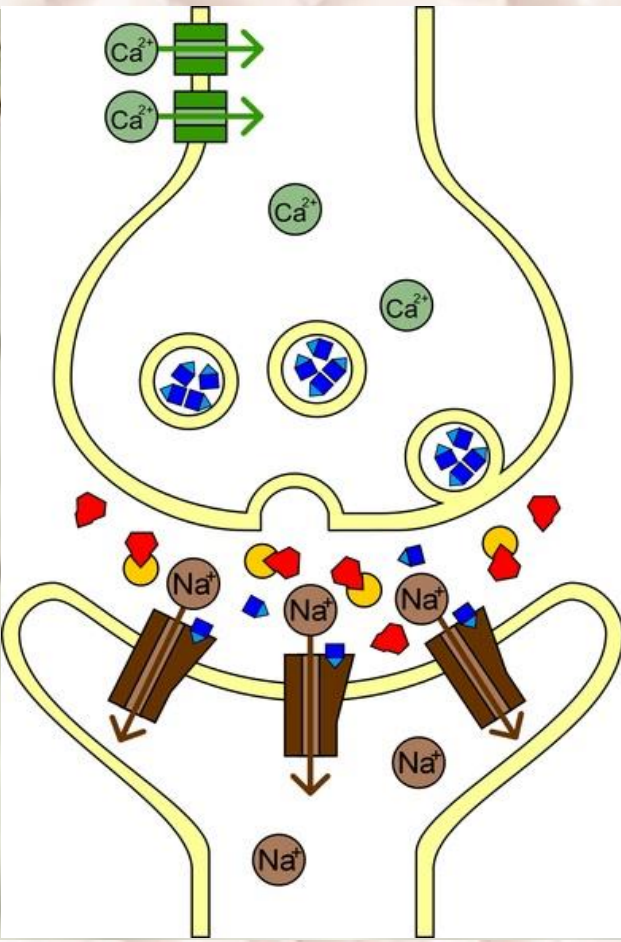
- wielkością i pojemnością,
- wrażliwością i reaktywnością,
- reprezentowanymi zbiorami kombinacji bodźców (danych),
- połączeniami
- i innymi cechami...



POŁĄCZENIA I SYNAPSY



- Szybkie i wybiórcze przekazywanie informacji
- Adaptatywne reagowanie na nadchodzące sygnały
- Zróżnicowanie reprezentowanych kombinacji





DANE



**napływają do nas z każdej strony,
o każdym czasie, w różnej postaci
i oddziałują na nasz układ nerwowy
poprzez zmysły wyposażone
w receptory, które przetwarzają je
na zunifikowaną postać
możliwą do przetworzenia
przez nasz system nerwowy.**



INFORMACJA



Czy informacja jest zbiorem danych?



Ten znak dla osoby znającej język chiński oznacza miłość.



Ta liczba może kojarzyć się z różnymi kwestiami.

Jeśli ten sam **zbiór danych** niesie różną informację dla różnych ludzi w różnych okolicznościach, oznacza to, że nie możemy zdefiniować informacji jako zbioru danych, gdyż wtedy takie pojęcie byłoby niejednoznaczne!



INFORMACJA



Czym więc jest informacja?

- ✓ **Informacja** jest przekazywana/odbierana/dostarczana za pośrednictwem danych.
 - ✓ Zbiór danych staje się **informacją** dopiero w momencie ich odebrania przez odbiorcę, dla którego te dane mają pewne określone znaczenie w kontekście posiadanej przez niego wiedzy lub stanu, w jakim się znajduje.
- ✓ Identyczne dane mogą kodować wiele całkowicie odmiennych **informacji** dla tego samego lub różnych odbiorców w zależności od kontekstu zdarzeń, jego/ich stanu lub aktualnej wiedzy.
- ✓ **Informacja** zwykle poszerza wiedzę odbiorcy i/lub wpływa na zmianę jego stanu w kontekście wykonywanych czynności.



ENTROPIA



Entropia H to średnia ilość informacji przypadająca na znak symbolizujący zajście zdarzenia x_i z pewnego n -elementowego zbioru X , czyli średnia ważona ilość informacji niesionej przez pojedynczą wiadomość o przypisanym prawdopodobieństwie jej wystąpienia. Każde zdarzenie x_i w tym n -elementowym zbiorze ma przypisane pewne prawdopodobieństwo wystąpienia $p(i)$:

Entropia H jest:

$$H(X) = \sum_{i=1}^n p(x_i) \log_r \frac{1}{p(x_i)} = - \sum_{i=1}^n p(x_i) \log_r p(x_i)$$

- **Niejemna.**
- **Maksymalna**, gdy prawdopodobieństwa zajść wszystkich zdarzeń są takie same.
- **Zerowa**, gdy prawdopodobieństwa zajść zdarzeń przyjmują wartości tylko 0 (niemożliwe) lub 1 (pewne).
- **Sumą entropii**, gdy zdarzenia są niezależne (zachodzi tzw. superpozycja).

Z entropii korzystamy np. przy budowie drzew decyzyjnych:

Zachmurzenie = słonecznie:

$$I([2, 3]) = H\left(\frac{2}{5}, \frac{3}{5}\right) = -\frac{2}{5} \log\left(\frac{2}{5}\right) - \frac{3}{5} \log\left(\frac{3}{5}\right) = 0.971$$

Zachmurzenie = deszczowo:

$$I([3, 2]) = H\left(\frac{3}{5}, \frac{2}{5}\right) = -\frac{3}{5} \log\left(\frac{3}{5}\right) - \frac{2}{5} \log\left(\frac{2}{5}\right) = 0.971$$

Zachmurzenie = pochmurno:

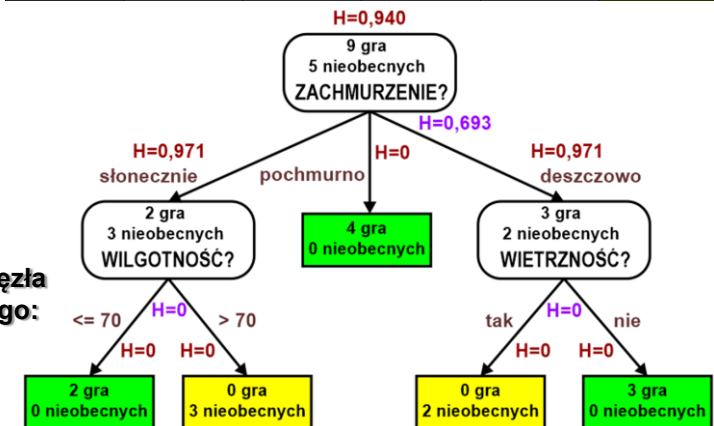
$$I([4, 0]) = H(1, 0) = -1 \log(1) - 0 \log(0) = 0$$

Następnie na podstawie określonych entropii potomków wyznaczamy wartość entropii węzła z wykorzystaniem prawdopodobieństwa LaPlace'a wystąpienia danego zdarzenia losowego:

$$I([3, 2], [4, 0], [3, 2]) = \left(\frac{5}{14}\right) * 0.971 + \left(\frac{4}{14}\right) * 0 + \left(\frac{5}{14}\right) * 0.971 = 0.693$$

Jeśli entropia = 0, oznacza to utworzenie liścia w drzewie.

Lp.	Zachmurzenie	Temperatura	Wilgotność	Wietrzność	Obecność
1	słonecznie	29	85	nie	nieobecny
2	słonecznie	27	90	tak	nieobecny
3	pochmurno	28	78	nie	gra
4	deszczowo	21	96	nie	gra
5	deszczowo	20	80	nie	gra
6	deszczowo	18	70	tak	nieobecny
7	pochmurno	18	65	tak	gra
8	słonecznie	22	95	nie	nieobecny
9	słonecznie	21	70	nie	gra
10	deszczowo	24	80	nie	gra
11	słonecznie	24	70	tak	gra
12	pochmurno	22	90	tak	gra
13	pochmurno	27	75	nie	gra
14	deszczowo	22	80	tak	nieobecny





INFORMATYKA



Czym jest informatyka?

Stara definicja encyklopedyczna:

*„**Informatyka** zajmuje się całokształtem przechowywania, przesyłania, przetwarzania i interpretowania informacji.”*

nie jest w pełni ścisła i poprawna, gdyż nawet współczesna informatyka nie realizuje tych postulatów, a ściśle mówiąc przechowywane, przesyłane i przetwarzane są dane, reprezentujące informacje, które jest na razie w stanie zinterpretować tylko człowiek obdarzony wiedzą i inteligencją, a nie komputer.

Definicja opracowana w 1989 r. przez Association for Computing Machinery głosi:

*„**Informatyka** to systematyczne badanie procesów algorytmicznych, które charakteryzują i przetwarzają informacje, teoria, analiza, projektowanie, badanie efektywności, implementacja i zastosowania procesów algorytmicznych.*

*Podstawowym pytaniem informatyki to:
co można (efektywnie) zalgorytmizować.”*

*„**Algorytmika** - fundament informatyki, wiedza o sposobach rozwiązywania zagadnień, czyli konstruowaniu algorytmów.”*

HISTORIA INFORMATYKI



związana jest z rozwojem języków programowania, których obecnie istnieje kilkaset włącznie z różnymi ich dialektami, najważniejsze to:

Asemblery – służą do tworzenia kodu maszynowego

Fortran – najstarszy, duże biblioteki i dobre kompilatory

Basic – miał być prostym językiem do nauki dla początkujących

Algol – na jego podstawie powstały współczesne języki programowania

Pascal – język strukturalny pozwalający na dzielenie zadań na mniejsze, operujący na dynamicznych strukturach danych

C – język strukturalny, bardzo powszechny, pozwala na dużą swobodę, skrótowość zapisu, a nawet dobieranie się do rejestrów procesora.

C++ – to język obiektowy bazujący na C, umożliwiający tworzenie klas, dziedziczenie, enkapsulację

C# - to język obiektowy wywodzący się z C++

Java – to język obiektowy podobny do C++ umożliwiający łatwe i szybkie tworzenie apletów o niewielkich rozmiarach i bogatej funkcjonalności.

Python – to też język obiektowy upraszczający zapis wielu często wykonywanych operacji na danych, nadający się do obliczeń naukowych.

PODZIAŁ JĘZYKÓW PROGRAMOWANIA



Języki niskiego poziomu – kod maszynowy i asemblery związane z typem komputera i procesora.

Języki wysokiego poziomu – posiadające translator (interpreter lub kompilator) tłumaczący komendy języka na rozkazy (kod maszynowy) dla danego typu komputera.

Języki imperatywne (tj. Fortran, Basic, Algol) wykorzystują rozkazy lub instrukcje służące do wykonania czynności składających się na wykonanie programu.

Języki proceduralne (tj. C, Pascal) dzielące kod na procedury, czyli pewne wydzielone fragmenty kodu, wykonujące ściśle określone operacje dla określonych parametrów danych wejściowych oraz danych lokalnych.

Języki deklaratywne (tj. SQL) umożliwiają skupienie się na problemie, a nie na sposobie jego rozwiązania, przypominają składnię języka naturalnego i opisują warunki, jakie musi spełniać końcowe rozwiązanie, a nie sekwencję kroków, rozkazów czy instrukcji.

Języki funkcyjne (tj. Lisp, F#, rachunek lambda) wyrażają funkcje umożliwiające obliczenie wartości tak jak funkcje.

Języki w logice (tj. Prolog) służą do zapisywania relacji i formuł logicznych ewaluowanych zgodnie z zasadami logiki, rachunkiem predykatów itd.

Języki obiektowe (tj. Delphi, C++, C#, Java, Python) pozwalają na tworzenie obiektów oraz łączenie ich poprzez pewne relacje, tworzeniu zależności i komunikację między nimi poprzez wysyłanie komunikatów lub interfejsy, zaś obiekty mają za zadanie enkapsulować właściwości i operacje związane z tymi obiektami, umożliwić dziedziczenie danych i metod po obiektach rodzicielskich tworząc obiekty potomne oraz drzewa zależności pomiędzy obiektami.

Obecnie najpopularniejsze są języki programowania obiektowego i deklaratywnego.

GENERACJE JĘZYKÓW PROGRAMOWANIA



Języki pierwszej generacji – to języki maszynowe procesorów składające się z rozkazów procesora zapisywanych w postaci liczb, operujące na rejestrach i wykonujące podstawowe operacje wbudowane w procesor (CPU lub GPU).

Języki drugiej generacji – to języki symboliczne i asemblery są tożsame z językiem maszynowym, lecz pozwalające na łatwiejsze zapamiętanie rozkazów zapisanych w postaci mnemoników, np.: `mov ax, bx` `add ax, 1`

Języki trzeciej generacji – to języki proceduralne, imperatywne i obiektowe pozwalające na wykorzystanie instrukcji tłumaczonych przez kompilator lub interpreter na kod maszynowy danego procesora lub komputera, dzięki czemu stają się łatwiej przenośne, a stosowne instrukcje bardziej intuicyjne.

Języki czwartej generacji – to języki deklaratywne, zorientowane problemowo, stosujące składnię przypominającą język naturalny, stosowane np. do dostępu do baz danych:

np. `SELECT Imię, Nazwisko FROM Studenci WHERE Średnia > 4.5`

Języki piątej generacji – to języki sztucznej inteligencji, metod inżynierii wiedzy oraz systemów ekspertowych, operujące językiem logiki i językiem naturalnym.