

INFORMATYKA

C

WYDZIAŁ ELEKTROTECHNIKI I
INFORMATYKI
ELEKTROTECHNIKA
DR INŻ. MICHAŁ ŁANCZONT

I

- [www: lanczont.pollub.pl](http://www.lanczont.pollub.pl)
- E-mail: m.lanczont@pollub.pl
- Tel. 81-538-4293
- Budynek WEiI – pokój 419



Informacje wstępne

- Wykład – 2h przez 10 zjazdów
- Laboratorium – 3h przez 5 zjazdów

Warunki zaliczenia:

- Wykład rozliczany jest egzaminem
- Egzamin + 2 terminy poprawkowe
- Egzamin w formie pisemnej (test)

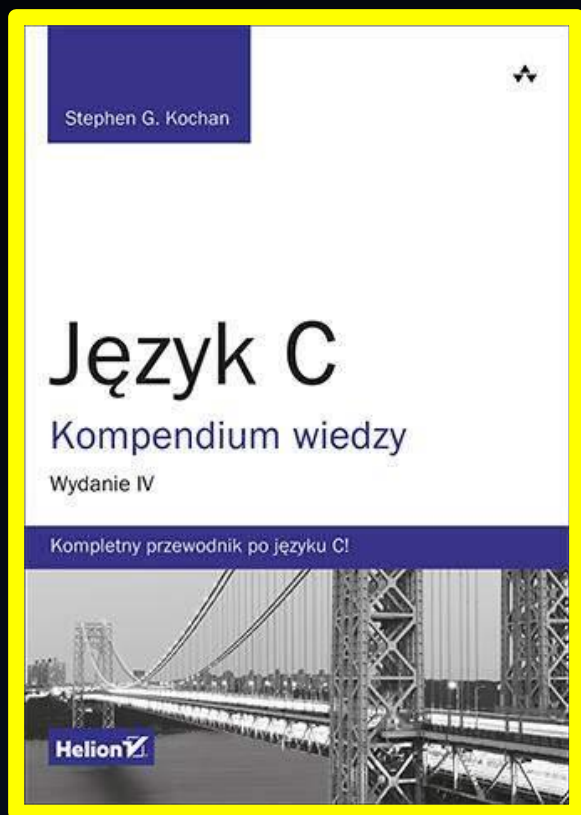
Zakres materiału

1. Pojęcia podstawowe związane z programowaniem, komputerami.
2. Algorytmika. Wprowadzenie do schematów blokowych i *Nassi-Schneidermana*. Narzędzia wspomagające rysowanie schematów.
3. Instalacja i konfiguracja środowiska programistycznego.
4. Struktura programu, zmienne, deklaracje zmiennych, typy danych, podstawowe wyrażenia arytmetyczne
5. Programowanie strukturalne i proceduralne, funkcje, prototypy funkcji, pliki nagłówkowe
6. Pętle iteracyjne i warunkowe, instrukcje przerwania pętli

Zakres materiału

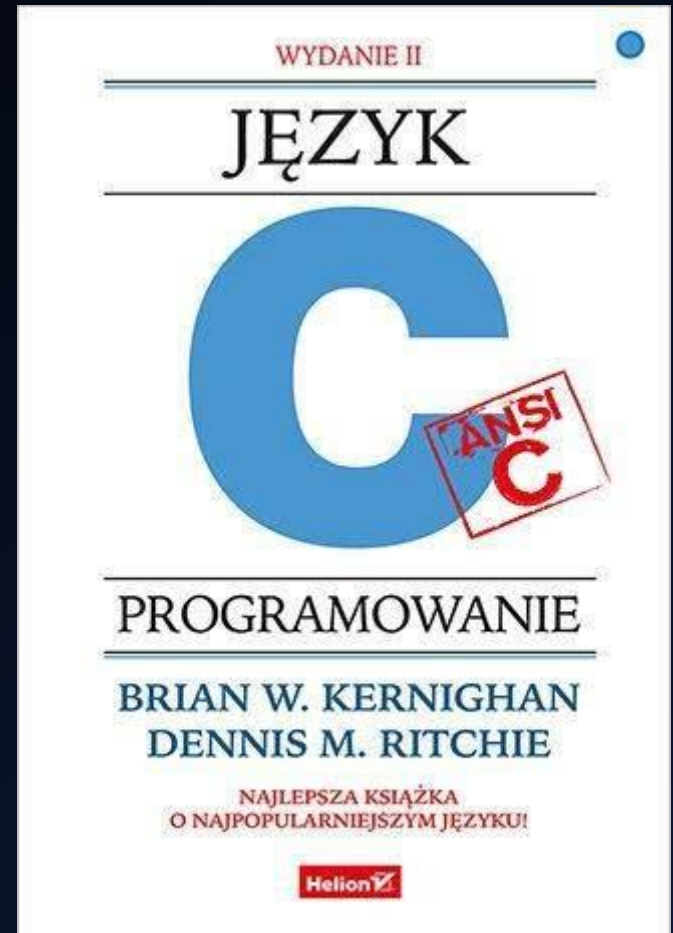
7. *Podjmowanie decyzji za pomocą instrukcji warunkowej i wyboru, operatory i wyrażenia logiczne*
8. *Zmienne tablicowe i strukturalne*
9. *Dynamiczna alokacja danych w pamięci*
10. *Łańcuchy znakowe i operacje na nich, biblioteka `string.h`*
11. *Wskaźniki i referencje*
12. *Operacje wejścia i wyjścia (zapis i odczyt danych z pliku tekstowego)*

- A. Kwiatkowska, E. Łukasik: **Schematy zwarte NS. Przykłady i zadania**, Mikom 2004



- Stephen G. Kochan: **Język C kompendium wiedzy**, Wydanie IV, Helion 2015

- Brian W. Kernighan, Dennis M. Ritchie: **Język C programowanie ANSI C**, Helion 2010



- Zed A. Shaw, **Programowanie w C**, Helion 2016

Literatura



Język C. Szkoła programowania la. ...
helion.pl



Helion Tomasz Francuz Język C ...
ekapiec.pl - W magazynie



Język C++ Szkoła Programowani ...
sprzedajemy.pl



Język C. Szkoła programowania la. ...
books.apple.com

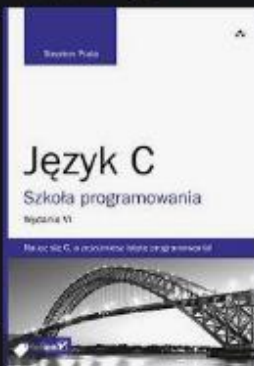


Język C Szkoła programowania la. ...
allegro.pl - W magazynie



Greg Perry, Dean Miller

Podręcznik do Informatyki Język ...
ceneo.pl - W magazynie



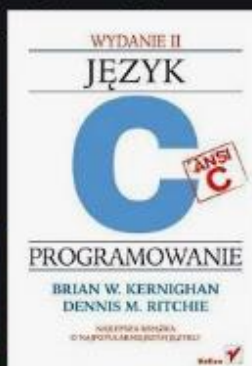
Język C. Szkoła programowania la. ...
helion.pl



Język C++ Szkoła programowani ...
tanialaskazka.pl - Brak w magazynie



Język C dla mikrokontrolerów AV/ ...
helion.pl



Język ANSI C. Programowanie - ...
lubimyczytac.pl



Język C. Nowoczesne programo...
cyfrotelka.pl



Szkola programowania język c++...
img.ed.pl

Komputer i peryferia

Urządzenia peryferyjne

Urządzenia peryferyjne wyjściowe

drukarka

Urządzenie wyjściowe służące do drukowania informacji tekstowych lub graficznych.



głośniki

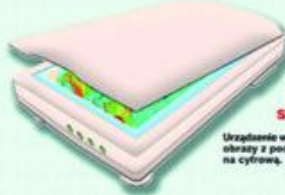
Pomocają odtwarzać dźwięki generowany przez kartę dźwiękową.



Urządzenia peryferyjne wejściowe

skaner

Urządzenie wejściowe przetwarzające obrazy z postaci analogowej na cyfrową.



tablet

Urządzenie wejściowe służące do wprowadzania do komputera grafik za pomocą specjalnego pióra i podkładki z czujnikami.



joystick

Manipulator drążkowy, używany głównie w grach komputerowych.



Urządzenia podstawowe

naped CD-ROM/DVD/nagrywarka

CD-ROM/DVD-ROM - dysk optyczny służący tylko do odczytu informacji. Do zapisu danych na płytach CD-ROM/CD-RW służą nagrywarki.



stacja dyskietek (FDD)

Naped służący do zapisu i odczytu informacji umieszczonych na nośniku magnetycznym.



klawiatura

Urządzenie wejściowe służące do wprowadzania poleceń i danych do jednostki centralnej.



monitor

Urządzenie wyjściowe za pomocą którego jednostka centralna wyprowadza informacje wizualne na ekran.



jednostka centralna

Zasadnicza część komputera obejmująca procesor, pamięć operacyjną (RAM), pamięć ROM, karty rozszerzeń, magistralę oraz zasilacz. Steruje działaniem układów elektronicznych i przetwarza informacje.

mysz

Urządzenie wejściowe służące do wskazywania obiektów na ekranie, zaznaczania ich, uruchamiania programów i sterowania zachowaniem się programu.



Urządzenia umieszczone wewnątrz jednostki centralnej

procesor

CPU - central processing unit (centralna jednostka wykonawcza) - układ scalony, którego działanie polega na wykonywaniu instrukcji programów. Nadzoruje i synchronizuje pracę wszystkich urządzeń w komputerze.



pamięć RAM

Random Access Memory - pamięć operacyjna, przechowuje aktualnie używane dane, tak aby były one błyskawicznie dostępne dla procesora.



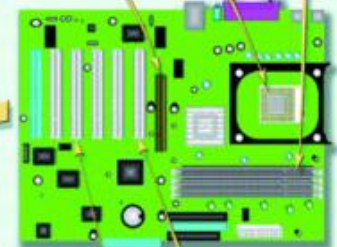
karta graficzna

Karta rozszerzeń, umieszczona na płycie głównej poprzez gniazdo PCI lub AGP - odpowiada za obraz wyświetlany przez monitor.



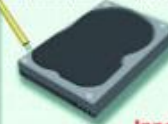
płyta główna

Składa się z wielu chipsetów, czyli układów scalonych, których zadaniem jest integracja oraz zapewnienie współpracy poszczególnych komponentów komputera (procesora, dysków twardego, monitora, klawiatury, kart rozszerzeń i innych).



dysk twardy

Służy do trwałego przechowywania danych. Na twardego dysku znajduje się oprogramowanie decydujące o funkcjonalności komputera: system operacyjny i programy użytkowe.



Inne karty rozszerzeń, np.: modem wewnętrzny, karta sieciowa

Modem - urządzenie dzięki któremu można połączyć komputer z internetem poprzez łączą telefoniczne. Zamienia on cyfrowy sygnał komputera na analogowy telefonu i odwrotnie.

Karta sieciowa - urządzenie umożliwiające połączenie komputera w sieć z innymi komputerami.



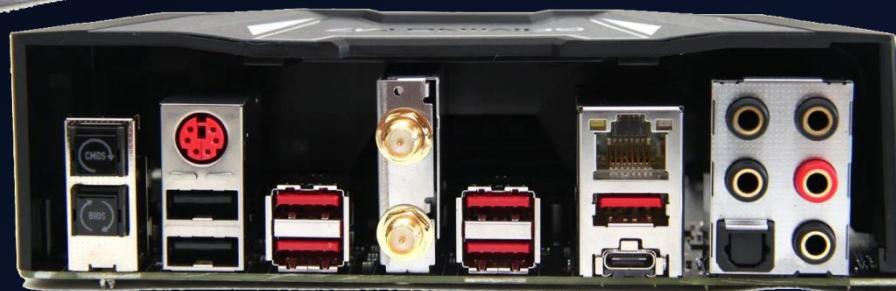
karta dźwiękowa

Karta rozszerzeń pozwalająca na odtwarzanie oraz zapis dźwięku w formie plików muzycznych. Karty muzyczne umożliwiają podłączenie do nich głośników, słuchawek, syntezatora, mikrofonu oraz urządzeń MIDI.



Płyta główna

- Baza na której opiera się każdy komputer
- Pośredniczy pomiędzy wszystkimi elementami komputera
- Decyduje o stabilności pracy
 - Gniazdo procesora
 - Gniazda pamięci RAM (od 2 do nawet 8)
 - Sloty kart PCIe
 - Porty SATA/M.2.
 - porty USB, LAN oraz porty audio.
 - W bardziej rozbudowanych:
 - wyjścia na anteny dla wbudowanej bezprzewodowej karty sieciowej,
 - cyfrowe wyjście audio
 - przyciski do resetowania CMOS/BIOS procesor.



- Centralna jednostka decydująca o wydajności komputera
- Obecnie dominującą pozycję mają Intel i AMD
- Z procesorem CPU często zintegrowana jest karta graficzna (Intel)
- Od niedawna Intel oferuje procesory bez GPU (oznaczone literą F)
- AMD także integruje GPU w CPU (procesory z oznaczeniem G)

- Pamięci ewoluowały wraz z rozwojem komputerów
- Obecnie dominują typy DDR3 i DDR4
- Wybór typu jest wymuszony przez zastosowaną płytę główną
- Każdy z typów występuje w szeregu wersji różniących się częstotliwością pracy

- Dominująca pozycja dysków z zapisem magnetycznym pomalu przemija
- Znaczenia nabierają dyski SSD, przechowujące dane w układach scalonych NAND
- SSD w formacie HDD2.5 ze złączem SATA – transfer do około 550 MB/s
- SSD ze złączem M.2. – standard SATA i NVMe
- Standard NVMe transfer do około 4000MB/s
- Pojawienie się płyt głównych z interfejsem PCI₂ 4.0 pozwoli na wzrost transferu danych do poziomu około 8000 MB/s

Pozostałe elementy

- **Zasilacz** – decyduje o zapewnieniu odpowiedniego poziomu mocy dla pozostałych elementów zainstalowanych w komputerze
- **Karta graficzna** – obecnie dwie dominujące marki Nvidia i AMD – wybór determinowany zastosowaniem komputera
- **Obudowa** – ergonomii ułożenia elementów komputera oraz parametrach środowiskowych (chłodzenie)

Programowanie – proces projektowania, tworzenia, testowania i utrzymywania **kodu źródłowego programów** komputerowych lub urządzeń mikroprocesorowych (mikrokontrolery).

- projektowanie aplikacji
- algorytmika
- struktury danych
- języki programowania
- narzędzia programistyczne
- kompilatory
- sposób działania podzespołów komputera

Algorytm – dokładny, jednoznacznie sformułowany sposób postępowania, umożliwiający rozwiązanie określonego zadania w skończonej liczbie kroków.

- a) w postaci opisu słownego
- b) w postaci pseudokodu
- c) w postaci schematu blokowego

Pojęcia podstawowe

Algorytm określa **dane** oraz **skończony ciąg operacji**, jakie należy na tych danych wykonać, aby rozwiązać dowolny problem z określonej klasy.

1. Kompletność

- algorytm uwzględnia wszystkie możliwe przypadki które mogą wystąpić w trakcie realizacji algorytmu
- algorytm przewiduje wystąpienia błędów (*numerycznych, logicznych, itd.*) oraz posiada system reakcji (*komunikaty o błędach, odpowiednie zakończenie działania*)

2. Skończoność

- algorytm zapewnia osiągnięcie rozwiązania w skończonej liczbie kroków (*czyli w skończonym czasie*)
- dokładna liczba kroków nie jest z góry określona
- algorytm posiada warunek zakończenia

3. Jednoznaczność

- dla tych samych danych wejściowych algorytm musi dawać zawsze te same wyniki
- algorytm jest niezależny od momentu jego uruchomienia, wpływu innych programów, sprzętu, kodowania znaków, itp.

Algorytm powinien być:

- **Poprawny** — dla każdego poprawnego zestawu informacji wejściowych prowadzić do poprawnych rezultatów.
- **Jednoznaczny** — każdorazowo, dla każdego poprawnego zestawu informacji wejściowych, prowadzić do tych samych rezultatów.

Algorytm powinien być:

- **Szczegółowy** — aby ktoś albo coś wykonujące algorytm rozumiał dokładnie opisane operacje i potrafił je wykonać.
- **Uniwersalny** — pozwalał na rozwiązanie dowolnego problem z określonej klasy, a nie dotyczył pewnych przypadków.

Metody zapisu algorytmu

1. **Język naturalny (potoczny)** określający ciąg kroków algorytmu.
2. **Notacje graficzne** – najpopularniejsze są schematy blokowe.
3. **Pseudokod** — połączenie języka naturalnego i notacji matematycznej z elementami języków programowania.
4. **Kodu programu** – zapis w pewnym języku programowania.

Pojęcia podstawowe

Język programowania – zbiór zasad określających, kiedy ciąg symboli tworzy program komputerowy oraz jakie obliczenia opisuje.

Język programowania pozwala na precyzyjny zapis algorytmów oraz innych zadań, jakie komputer ma wykonać.

*Niekiedy pojęcie **języka programowania** jest ograniczane wyłącznie do tych języków, w których można zapisać wszystkie istniejące algorytmy*

Pojęcia podstawowe

- **Język maszynowy**, kod maszynowy – zestaw rozkazów procesora, w którym zapis programu wyrażony jest w postaci liczb binarnych stanowiących rozkazy oraz ich argumenty.
- Język programowania komputerów **zerowej generacji** z wyjątkiem komputera **Z4**, a powszechny w początkowym okresie rozwoju komputerów pierwszej generacji.

Pojęcia podstawowe

- Kod maszynowy może być generowany w procesie **kompilacji** (języki wysokiego poziomu) lub **aseblacji** (języki niskiego poziomu).
- Język maszynowy jest **nieprzenośny**, ponieważ każda **architektura procesora** ma swój własny model programowy, a więc m.in. listę rozkazów maszynowych.

Pojęcia podstawowe

- **Język niskiego poziomu** – język programowania umożliwiający zapis rozkazów maszynowych za pomocą stosunkowo prostych oznaczeń symbolicznych, np. każdy język assemblerowy jest językiem niskiego poziomu.

- W assemblerze operujemy bezpośrednio na rejestrach procesora i pośrednio także na komórkach pamięci.

```
mov ax, 0D625h  
mov es, ax  
mov al, 24  
mov ah, 0  
int 21h
```

Pojęcia podstawowe

- **Język wysokiego poziomu** – typ języka programowania, którego składnia i słowa kluczowe mają maksymalnie ułatwić rozumienie kodu programu dla człowieka, tym samym zwiększając poziom abstrakcji i dystansując się od sprzętowych niuansów.
- Kod napisany w **języku wysokiego poziomu** nie jest bezpośrednio „zrozumiały” dla komputera, aby umożliwić wykonanie programu napisanego w tym języku należy dokonać procesu **kompilacji**.

Pojęcia podstawowe

- **Asemblerowy język wysokiego poziomu** to termin określający język programowania wysokiego poziomu, posiadający cechy i elementy umożliwiające programowanie **bliskie sprzętowi**.
- Istnieją bezpośrednie odwzorowanie instrukcji wysokiego poziomu danego języka na język asemblera .
- Przynależność danego języka do tej grupy języków jest dość umowna i płynna: **BCPL, C--, FAP (FORTRAN Assembly Program), FORTH, PL/M**

Pojęcia podstawowe

- Miernikiem „**poziomu**” danego języka programowania jest również jego powiązanie ze **sprzętem komputerowym** (ang. *hardware*).
- **Implementacja** – zestaw narzędzi pośredniczący w tłumaczeniu kodu zapisanego w danym języku na kod maszynowy danego **procesora**.
- Każdy język programowania ma z reguły kilka **implementacji** przeznaczonych na różne **platformy** i robione przez różnych **producentów**

Pojęcia podstawowe

- **Plik (Kod) źródłowy** - program komputerowy w postaci tekstu zawierającego instrukcje języka programowania.
- **Kod źródłowy** jest jedynym formatem programu czytelnym dla człowieka.
- **Kod źródłowy** jest przetwarzany na **kodek maszynowy** przez program zwany **translatorem**.

Pojęcia podstawowe

- **Translator** – program komputerowy (lub urządzenie), dokonujący tłumaczenia (translacji) programu napisanego w określonym języku programowania, z postaci źródłowej do postaci wynikowej możliwej do wykonania przez maszynę:
 - **kompilatory** tłumaczące programy zapisane w językach wysokiego poziomu,
 - **assemblery** tłumaczące programy zapisane w językach symbolicznych.

Etapy kompilacji

- a) **Preprocessing** – obejmuje on odczyt i analizę kodu źródłowego na podstawie reguł danego języka.
- a) **analiza leksykalna** – rozdzielenie kodu źródłowego na elementarne jednostki języka programowania zwane tokenami;
- b) **analiza składniowa** – kontrola, czy ułożenie tokenów nie łamie reguł danego języka programowania (czy kod źródłowy był poprawny składniowo);
- c) **analiza semantyczna** – określenie znaczenia poszczególnych tokenów. Przykładem może być sprawdzanie, czy nie występuje niezgodność typów.

2. Kompilacja – służy do przetłumaczenia kodu źródłowego na kod obiektowy.

- a) W przypadku jednoczesnej kompilacji wielu plików źródłowych, każdy z nich przetworzony jest na oddzielny kod obiektowy.
- b) Na tym etapie nie są analizowane zależności pomiędzy poszczególnymi segmentami programu.

3. Konsolidacja – służy do połączenia wszystkich plików obiektowych oraz dodatkowych bibliotek statycznych w jeden kod wynikowy.

- a) Na tym etapie sprawdzane są zależności pomiędzy poszczególnymi segmentami programu.
- b) Dodanie biblioteki statycznej wymaga podania odpowiednich opcji kompilacji.
- c) Po konsolidacji program jest gotowy do uruchomienia.

Pojęcia podstawowe

Interpreter – program tłumaczący i wykonujący kod źródłowy linia po linii.

- a) W tym przypadku nie ma pliku wynikowego, a każde uruchomienie wymaga ponownego przeprowadzenia etapu interpretacji – interpreter jest więc niezbędny do uruchomienia i wykonania programu.
- b) Typowymi przykładami języków interpretowanych są: Python, MatLab, **Scilab**, GNU Plot, HTML, BPL.

Pojęcia podstawowe

Biblioteka statyczna – rodzaj biblioteki, która łączona jest z programem w etapie konsolidacji. W systemach z Windows mają rozszerzenia **.lib** lub **.obj**, natomiast w systemach Unix są to zwykle **.a** lub **.o**.

Biblioteka dynamiczna – rodzaj biblioteki, która łączona jest z programem dopiero w momencie jego wykonania. W systemach Windows biblioteki dynamiczne mają zazwyczaj rozszerzenie **.dll**.

- Służy do pisania tekstu kodu źródłowego.
- Edytory mogą być uniwersalne (np. ConText, Amigo, Crimson, Codex, EditPlus, UltraEdit, Editeur, ATOM, **Visual Studio Code**, ...) lub też zintegrowane z konkretną implementacją języka (np. Borland Delphi, Compaq Visual Fortran ...).
- Edytory uniwersalne pozwalają na podłączanie kompilatorów jednego lub wielu języków i są rozprowadzane jako oddzielne programy.

Debugger (analizator kodu)

- Służy do analizy poprawności tworzonego kodu źródłowego i stanowi zazwyczaj integralną część translatora (może być to również oddzielny program).
- Podczas analizy kodu źródłowego generowane są informacje o błędach
- Analizatory podają również tzw. ostrzeżenia, tzn. informacje o znalezieniu niejasności w kodzie lub o wykryciu miejsca potencjalnie niebezpiecznego.

Etapy rozwoju języków

Pierwszej generacji - są to języki maszynowe, czyli języki procesorów.

Instrukcje są w nich zapisywane w postaci liczb binarnych.

Etapy rozwoju języków

Drugiej generacji - języki symboliczne, asemblery.

- Języki niskiego poziomu, pod względem składni tożsame z maszynowymi, z tą różnicą że zamiast liczb używa się tu łatwiejszych do zapamiętania mnemoników.
- Pod względem znaczenia kod ten jest jednoznaczny z kodem binarnym (tylko dla procesorów z grupy 80x86).

Etapy rozwoju języków

Trzeciej generacji - języki wysokiego poziomu, proceduralne (imperatywne).

- W językach tych jedna instrukcja jest tłumaczona na kilka instrukcji procesora, najczęściej od 5 do 10.
- Pierwszym językiem tego typu był **ALGOL**.
- Do tej grupy należą między innymi:
 - FORTH, BASIC - języki niestukturalne,
 - Pascal, **C**, FORTRAN - języki strukturalne,
 - **C++**, Java - języki zorientowane obiektowo.

Etapy rozwoju języków

Czwartej generacji - języki bardzo wysokiego poziomu, nieproceduralne (deklaratywne).

- Korzystając z tych języków programista skupia się na problemie, a nie na sposobie jego rozwiązania.
- Składnia wielu języków czwartej generacji przypomina składnię języka naturalnego.
- Są one często używane do dostępu do baz danych. Przykładem języka z tej grupy jest: **SQL**.

Etapy rozwoju języków

Piątej generacji - języki sztucznej inteligencji, języki systemów ekspertowych.

- Języki najbardziej zbliżone do języka naturalnego.
- Przykładem języka piątej generacji jest PROLOG.

Paradygmat programowania

- wzorzec programowania komputerów
- definiuje sposób patrzenia programisty na przepływ sterowania i wykonywanie programu komputerowego
- różne języki programowania mogą wspierać różne paradygmaty programowania. Niektóre języki wspierają kilka paradygmatów, są określane jako języki hybrydowe
- to projektanci i programiści decydują, jak zbudować z nich wszystkich w pełni działający program

Programowanie strukturalne

- paradygmat programowania opierający się na podziale kodu źródłowego programu na procedury i hierarchicznie ułożone bloki z wykorzystaniem struktur kontrolnych w postaci instrukcji wyboru i pętli.
- charakteryzuje się zwiększoną czytelnością i ułatwia analizę programów, co stanowi znaczącą poprawę w stosunku do trudnego w utrzymaniu „**spaghetti code**” często wynikającego z użycia instrukcji **goto**.

Programowanie strukturalne

Programowanie strukturalne –

paradygmat **programowania** opierający się na podziale kodu źródłowego programu na procedury i hierarchicznie ułożone bloki z wykorzystaniem struktur kontrolnych w postaci instrukcji wyboru i pętli.

Programowanie proceduralne –

paradygmat **programowania** zalecający dzielenie kodu na procedury, czyli fragmenty wykonujące ściśle określone operacje.

Programowanie proceduralne

- paradygmat programowania zalecający dzielenie kodu na procedury,
- procedury nie powinny korzystać ze zmiennych globalnych (w miarę możliwości),
- powinny pobierać i przekazywać wszystkie dane (czy też wskaźniki do nich) jako parametry wywołania.

Programowanie obiektowe

- paradygmat programowania, w którym programy definiuje się za pomocą obiektów – elementów łączących *stan* (czyli dane) i *zachowanie* (czyli procedury).
- Obiektowy program komputerowy wyrażony jest jako zbiór takich obiektów, komunikujących się pomiędzy sobą w celu wykonywania zadań.

Programowanie obiektowe

- Programowanie obiektowe ma ułatwić pisanie, konserwację i wielokrotne użycie programów lub ich fragmentów.
- Największym atutem programowania, projektowania oraz analizy obiektowej jest zgodność takiego podejścia z rzeczywistością – mózg ludzki jest w naturalny sposób najlepiej przystosowany do takiego podejścia przy przetwarzaniu informacji.

Programowanie zdarzeniowe

- metodologia tworzenia programów komputerowych, która określa sposób ich pisania z punktu widzenia procesu przekazywania sterowania między poszczególnymi modułami tej samej aplikacji.
- Programowanie sterowane zdarzeniami jest mocno powiązane ze środowiskami wielopprocesowymi z graficznymi środowiskami systemów operacyjnych oraz z programowaniem obiektowym.

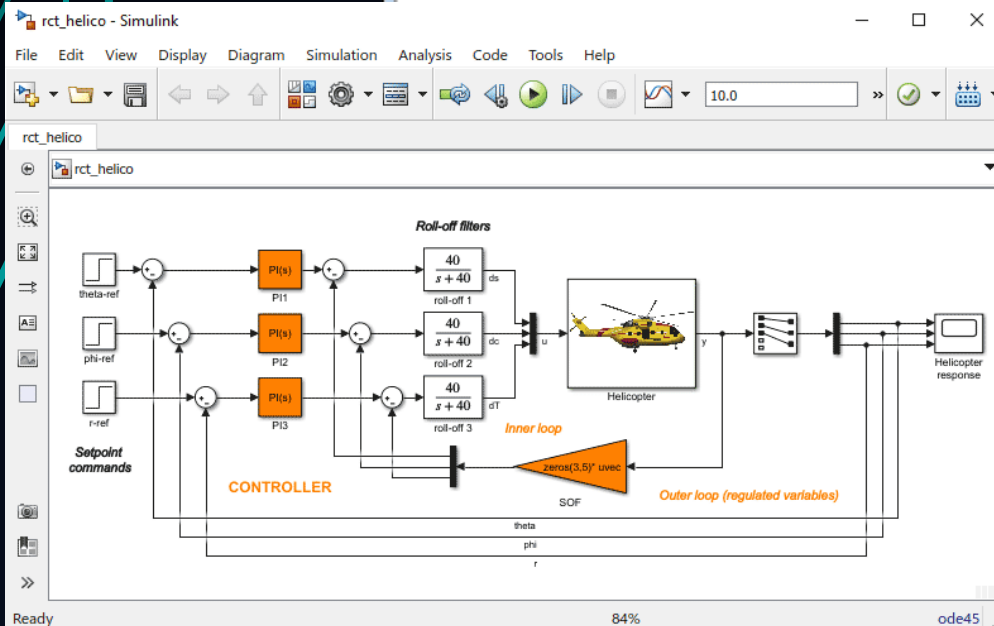
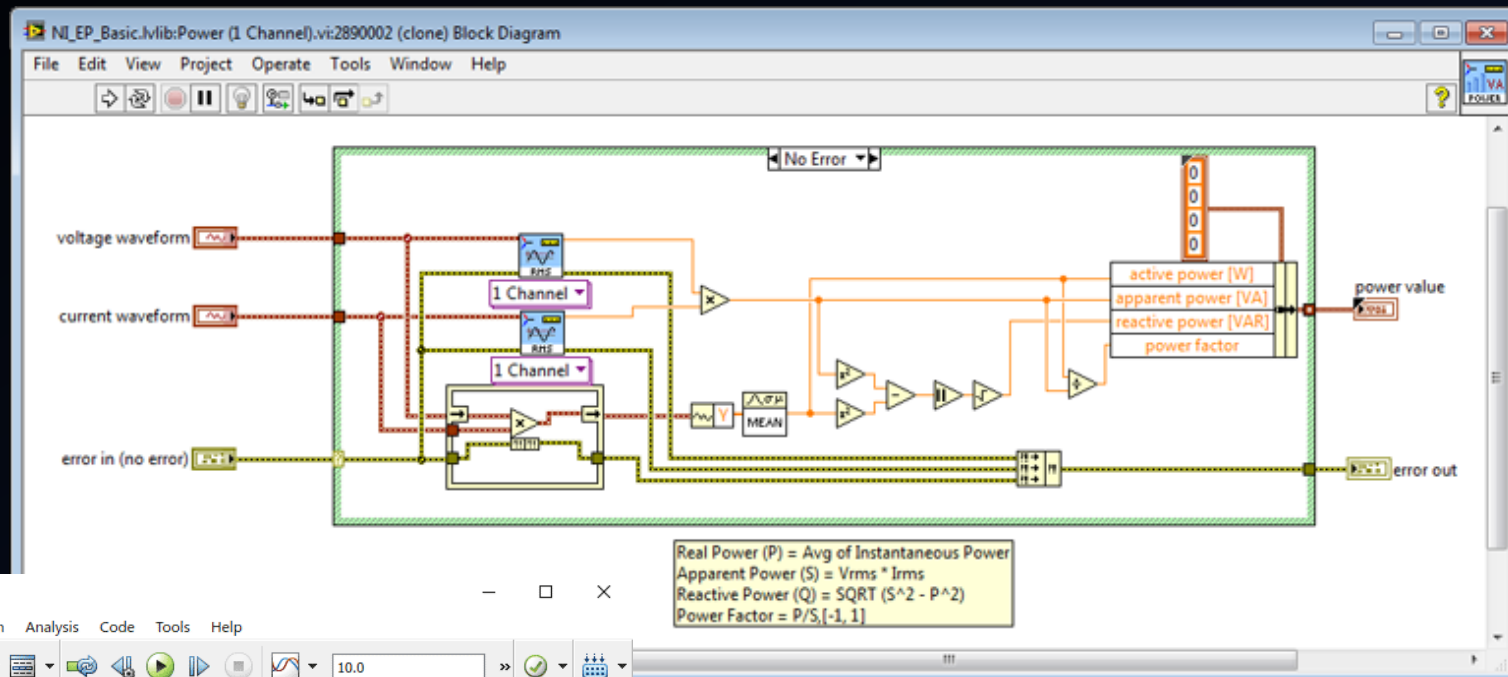
Programowanie zdarzeniowe

- program jest cały czas „bombardowany” zdarzeniami (*events*), na które musi odpowiedzieć, i zakładający, że przepływ sterowania w programie jest całkowicie niemożliwy do przewidzenia z góry.
- Programowanie zdarzeniowe jest dominującym typem programowania GUI – zdarzenia to naciśnięcia myszy, klawiszy, żądania odświeżenia przez system okienkowy, różne zdarzenia sieciowe i inne.

Modele programowania

- **Wizualne** – sposób programowania związany z środowiskami programowania wizualnego: **Matlab/Simulink, Scilab/Xcos, LabView, Scratch**
- Program tworzony jest za pomocą łączenia wyspecjalizowanych bloków realizujących konkretne zadania.
- Numeryczna analiza obiektów dynamicznych, nauka programowania dla dzieci i młodzieży

Programowanie Wizualne



Inne modele programowania

- programowanie **funkcyjne**
- programowanie **imperatywne**
- programowanie **uogólnione**
- programowanie **logiczne**
- programowanie **aspektowe**
- programowanie **deklaratywne**
- programowanie **agentowe**
- programowanie **modularne**

Style programowania

- Programowanie **zstępujące (Top-Down)** – rozwiązanie programistyczne polegające na zdefiniowaniu problemu ogólnego poprzez podzielenie na pod problemy, które są dzielone na jeszcze mniejsze pod problemy aż do rozwiązań oczywistych, łatwych do zapisania.
- Następnie złożenie z rozwiązań pod problemów niższego rzędu rozwiązań problemów wyższego rzędu aż do całkowitego rozwiązania problemu

Style programowania

Programowanie **wstępujące (Bottom-Up)** – pomijając analizę globalną zadania, programowanie prowadzi się od małych elementów do coraz większych, finalnie uzyskując cały program

Style programowania

- Wraz z rozwojem technologii tworzone języki coraz lepiej odpowiadające konkretnym potrzebom lub bardziej efektywne
- Rozwój determinowany był pod kątem szybkość pisania w danym języku lub efektywność obliczeniową.
- Do dnia dzisiejszego powstało ich ponad dwa i pół tysiąca.

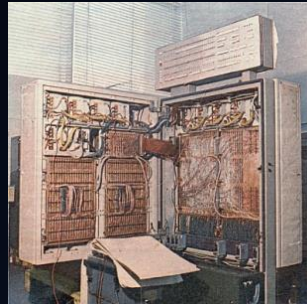
Style programowania

- Za pierwowzór programu komputerowego uznaje się taśmy perforowane.
- Odpowiednie dziurkowanie dawało możliwość zapisu, a następnie przetwarzania danych za pomocą maszyny.

Style programowania

- Za ich wynalazcę uważa się Josepha Jacquarda, a początkowo służyły do uzyskania tkaniny o powtarzalnym wzorze tkanej na krośnie.
- Na kartach zapisywane były dane o wzorach, które miały pojawić się na materiale.
- Na karcie w wyznaczonych miejscach mogły pojawiać się dziurki – w nie wsuwały lub nie wsuwały się trzpienie kontrolujące cały mechanizm krosna.

Dopiero Herman Hollerith na ich podstawie stworzył wzór karty na której zaczęto przechowywać dane.



The image contains two 8x26 dot grid puzzles. The top puzzle is for uppercase letters A-Z, and the bottom puzzle is for lowercase letters a-z. Each puzzle consists of an 8x26 grid of dots where some dots are filled in to form a hidden message. The filled dots, when read row by row, spell out 'THE SECRET MESSAGE' for the top puzzle and 'THE SECRET MESSAGE' for the bottom puzzle.

Taśma 8-ścieżkowa z kodem stosowanym w komputerach Odra 1300



- Pierwszym językiem programowania jest Fortran, stworzony w latach 1954 – 1957 przez pracownika IBM, Johna Backusa i jego zespół.
- Pełna nazwa języka to Formula Translation, a początkowo służył on do przeprowadzania obliczeń statystycznych i matematycznych.
- Jest pierwszym językiem wysokiego poziomu (w którym kod i składnia są bardzo przystępne dla człowieka)

- **C**ommon **B**usiness **O**riented **L**anguage powstał w 1959 roku w zespole prowadzonym przez Grace Murray Hopper dla systemów bankowych i przeprowadzanych przez nie transakcji.
- Prace na opracowanie języka zostały zlecone przez Departament Obrony w celu ustalenia wspólnego języka programowania dla środowisk przemysłowych.
- Pod koniec XX wieku aż 90 proc. transakcji finansowych na całym świecie było przetwarzanych przez programy napisane w COBAL'u.

- Nazwa języka jest skrótem **B**eginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode.
- Został stworzony w 1964 roku przez John Kemeny'ego i Thomas Kurtza z Dartmouth College
- Składnia języka wywodzi się z Fortran'a.
- Podstawowym celem twórców było stworzenie języka łatwego w użytkowaniu, o wszechstronnym zastosowaniu
- Ciekawostką jest, że Microsoft powstał po to, aby wydać i sprzedawać własny interpreter języka Visual BASIC.

- Język C powstał w 1969 roku. Jego twórcą jest Dennis Ritchie.
- Od niego też pochodzi większość znanych dziś języków programowania.
- To prosty w konstrukcji i praktyczny w użyciu język, który zyskał sławę dzięki nieograniczonej możliwości przenoszenia na różne urządzenia.
- Napisano w nim wiele systemów operacyjnych (Linux, Mac OS)

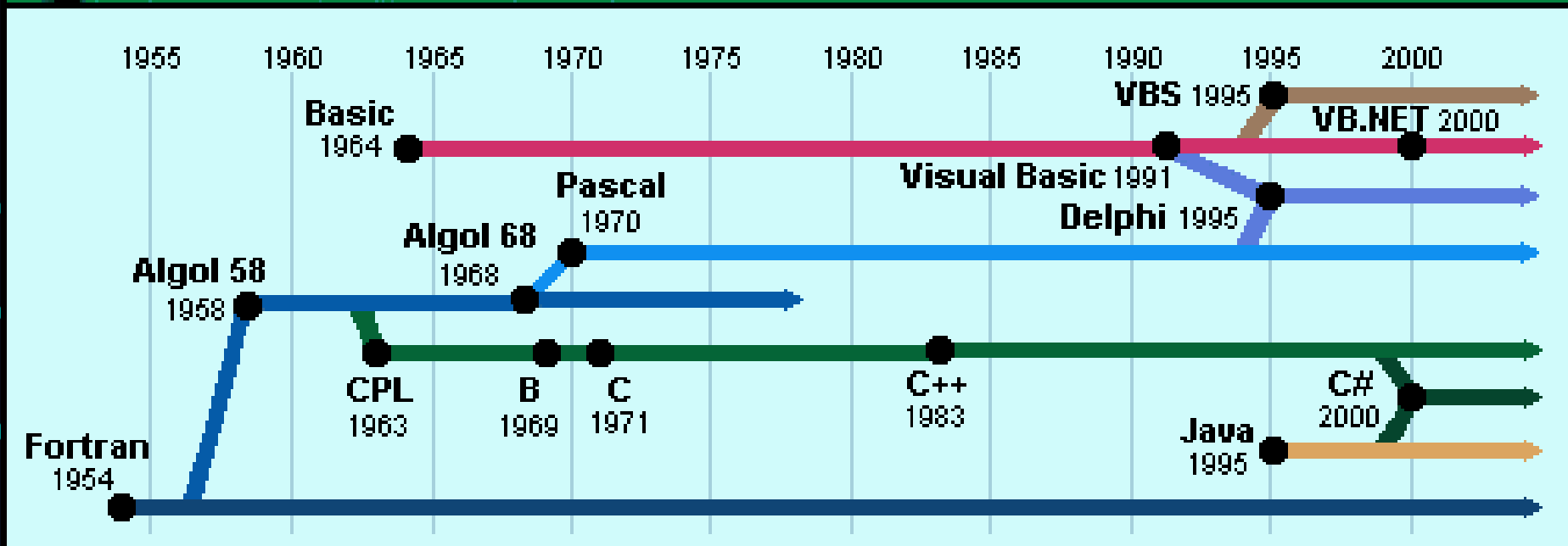
- C wykorzystuje się głównie do programowania systemowego, sterowników urządzeń i innych niskopoziomowych części oprogramowania (pośredniczą między urządzeniami podłączonymi do komputera, a systemem operacyjnym).
- Z C wywodzi się C++ (stworzony w 1983 roku), nazwany nawet językiem programowania lat 90.
- Jest łatwiejszy w użyciu niż jego pierwowzór.

Inne znaczące języki

- Python
- Ruby
- PHP
- JAVA
- JavaScript
- C#

Ewolucja niektórych języków programowania

1 Historia języków programowania



Popularność języków programowania według StackOverflow

Programming, Scripting, and Markup Languages



StackOverflow – serwis społecznościowy na którym programiści mogą zadawać pytania dotyczące szeroko pojętego wytwarzania oprogramowania.

Indeks popularności języków programowania według PYPL

Worldwide, Mar 2018 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Java	22.7 %	-0.8 %
2		Python	21.69 %	+5.4 %
3	↑↑	Javascript	8.53 %	+0.3 %
4	↓	PHP	8.33 %	-1.7 %
5	↓	C#	7.99 %	-0.7 %
6		C	6.42 %	-1.3 %
7	↑	R	4.23 %	+0.4 %
8	↓	Objective-C	3.81 %	-1.1 %
9		Swift	3.0 %	-0.6 %
10		Matlab	2.39 %	-0.4 %

Popularność języków programowania mierzona jest poprzez sprawdzanie, jak często są wyszukiwane są w Google kursy z danej dziedziny programowania.

Indeks TIOBE

Apr 2018	Apr 2017	Change	Programming Language	Ratings	Change
1	1		Java	15.777%	+0.21%
2	2		C	13.589%	+6.62%
3	3		C++	7.218%	+2.66%
4	5	▲	Python	5.803%	+2.35%
5	4	▼	C#	5.265%	+1.69%
6	7	▲	Visual Basic .NET	4.947%	+1.70%
7	6	▼	PHP	4.218%	+0.84%
8	8		JavaScript	3.492%	+0.64%
9	-	▲▲	SQL	2.650%	+2.65%
10	11	▲	Ruby	2.018%	-0.29%

Ranking tworzony jest na podstawie liczby zapytań zawierających nazwę języka programowania, jakie użytkownicy Internetu kierują do wiodących na rynku wyszukiwarek, takich jak Google, Bing, Baidu i Wikipedia.

Ranking podsumowujący

22 NAJPOPULARNIEJSZYCH JĘZYKÓW PROGRAMOWANIA 2018

		Tiobe	StackOverflow	PYPL GitHub
1	JAVA	1	5	1
2	JAVASCRIPT	8	1	3
3	PYTHON	4	7	2
4	C#	5	8	5
5	C	2	11	6
6/7	C++	3	10	-
	SQL	9	7	-
8	PHP	7	9	4
9	RUBY	10	13	11
10/11	VISUAL BASIC.NET	6	18	-
	TYPESCRIPT	-	12	12

Copyright © TeamQuest.pl

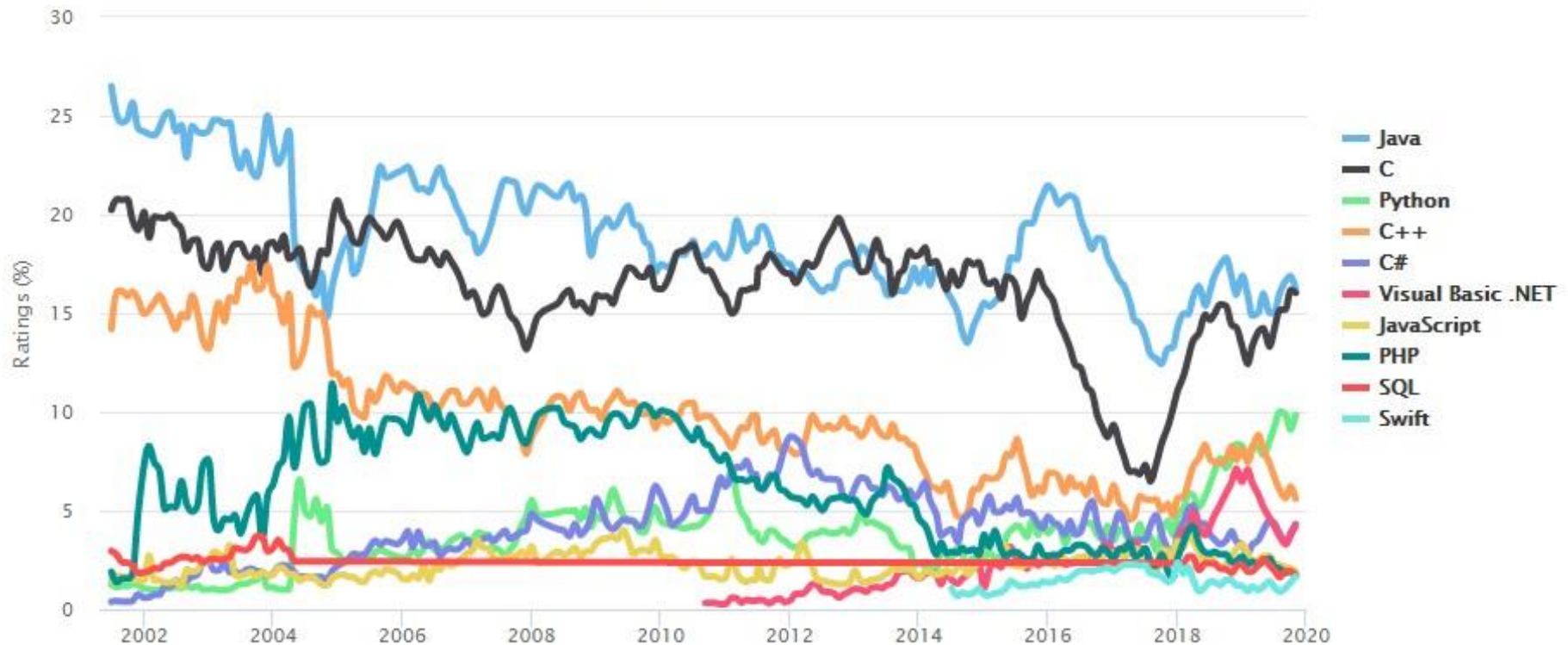
Ranking języków

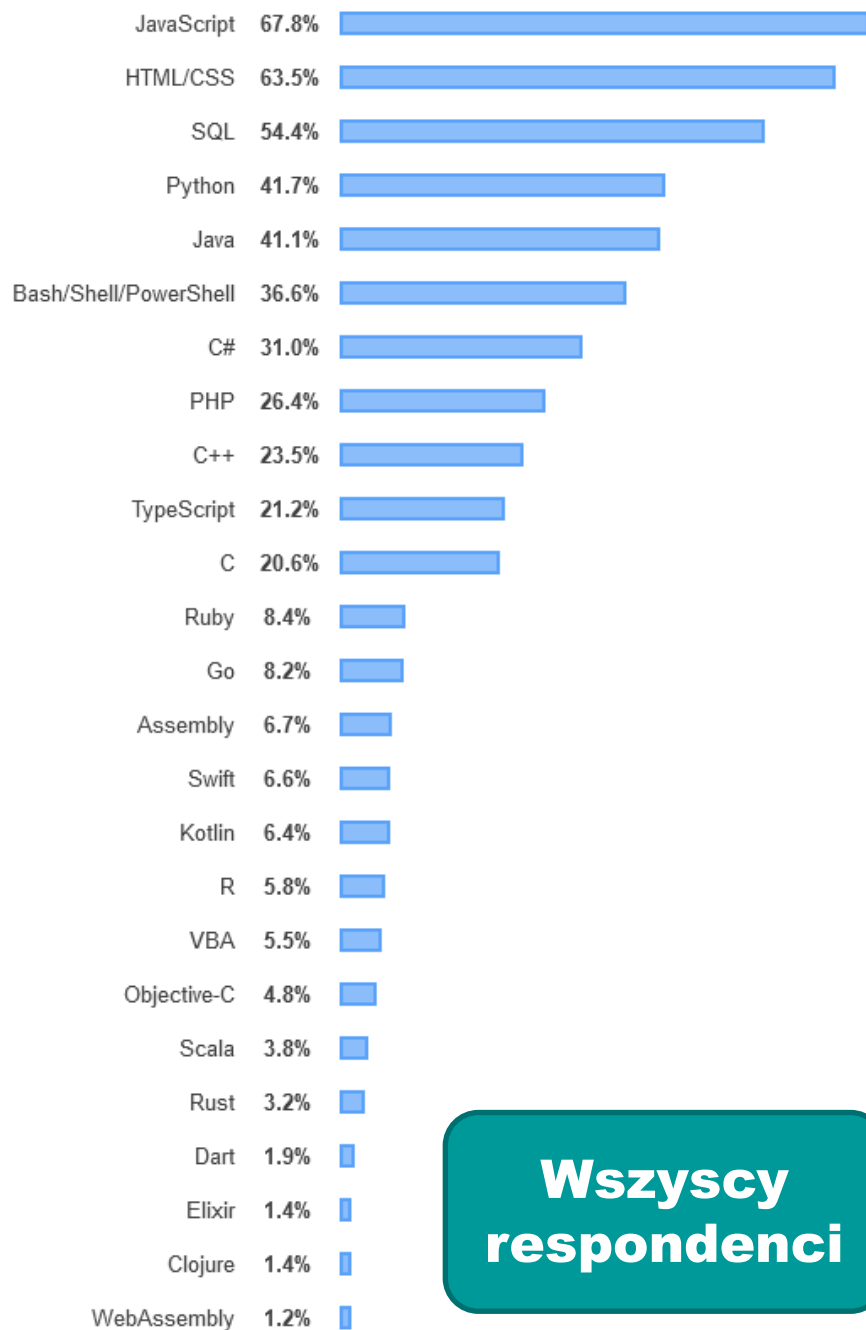
Nov 2019	Nov 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.246%	-0.50%
2	2		C	16.037%	+1.64%
3	4	▲	Python	9.842%	+2.16%
4	3	▼	C++	5.605%	-2.68%
5	6	▲	C#	4.316%	+0.36%
6	5	▼	Visual Basic .NET	4.229%	-2.26%
7	7		JavaScript	1.929%	-0.73%
8	8		PHP	1.720%	-0.66%
9	9		SQL	1.690%	-0.15%
10	12	▲	Swift	1.653%	+0.20%

Ranking języków w czasie

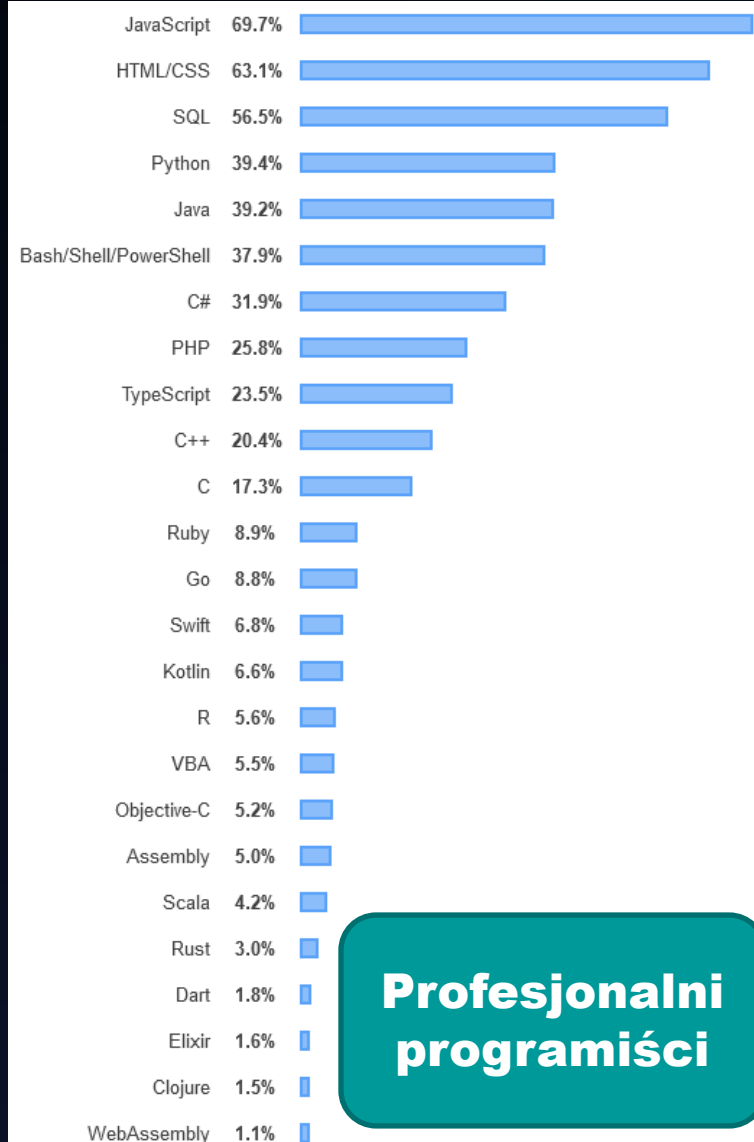
TIOBE Programming Community Index

Source: www.tiobe.com

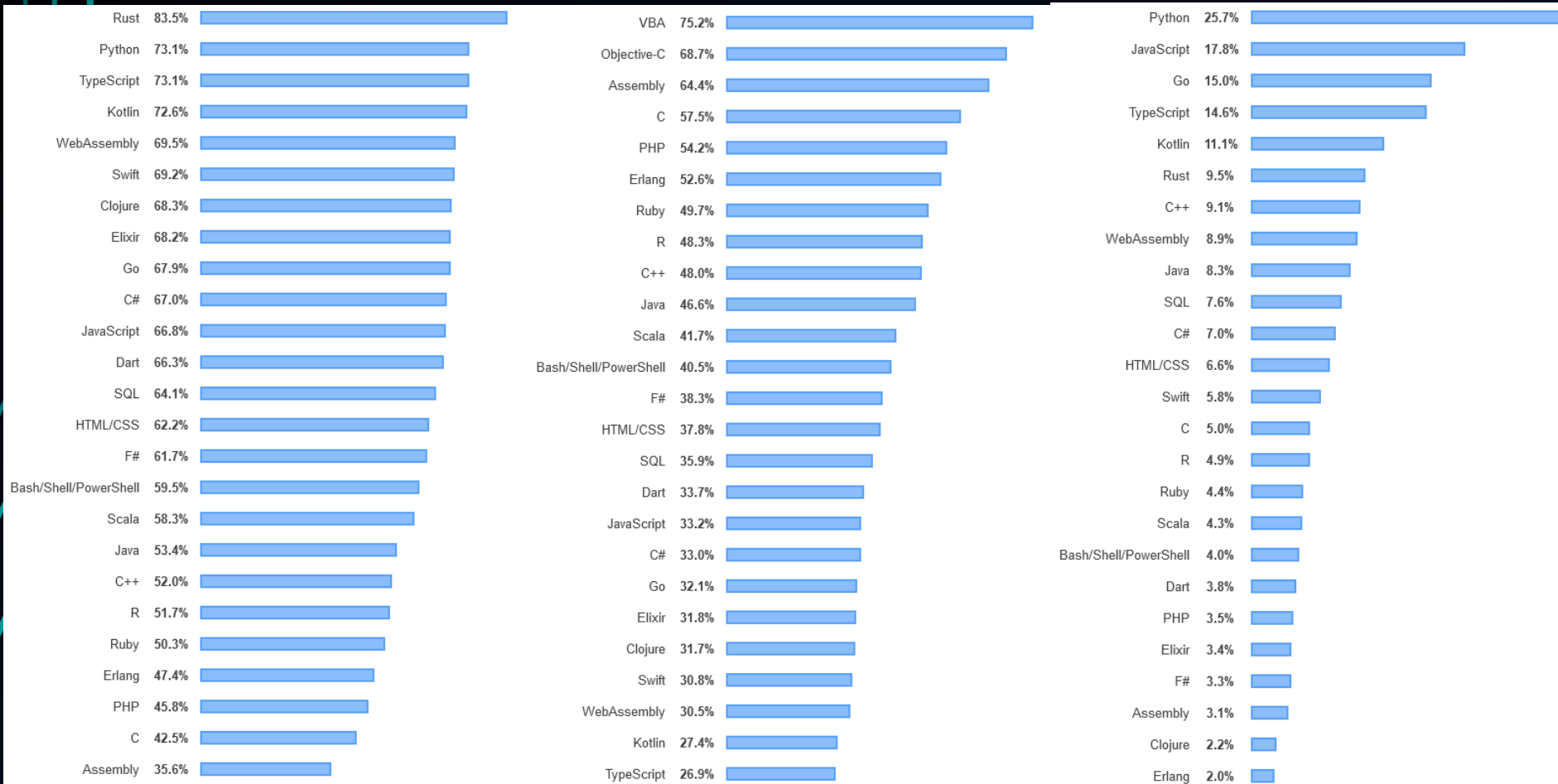




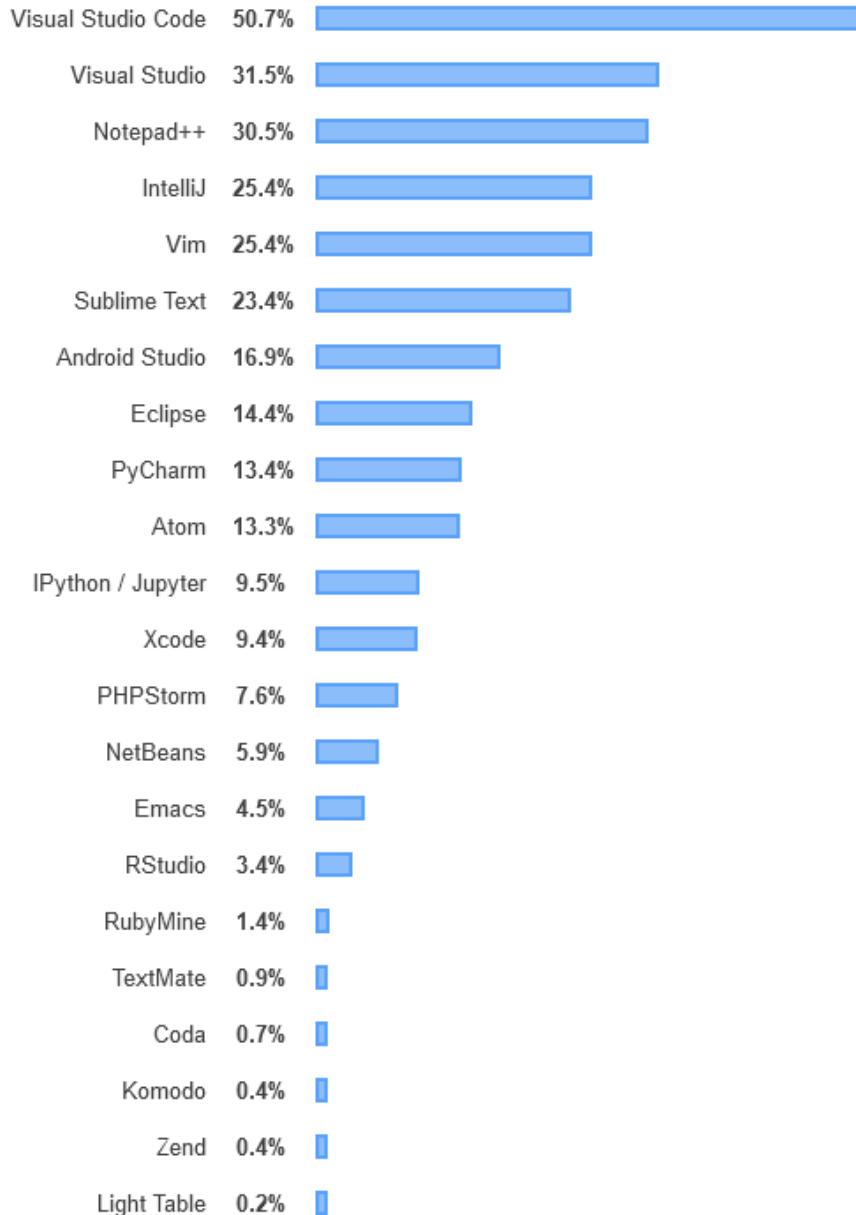
Ranking 2019



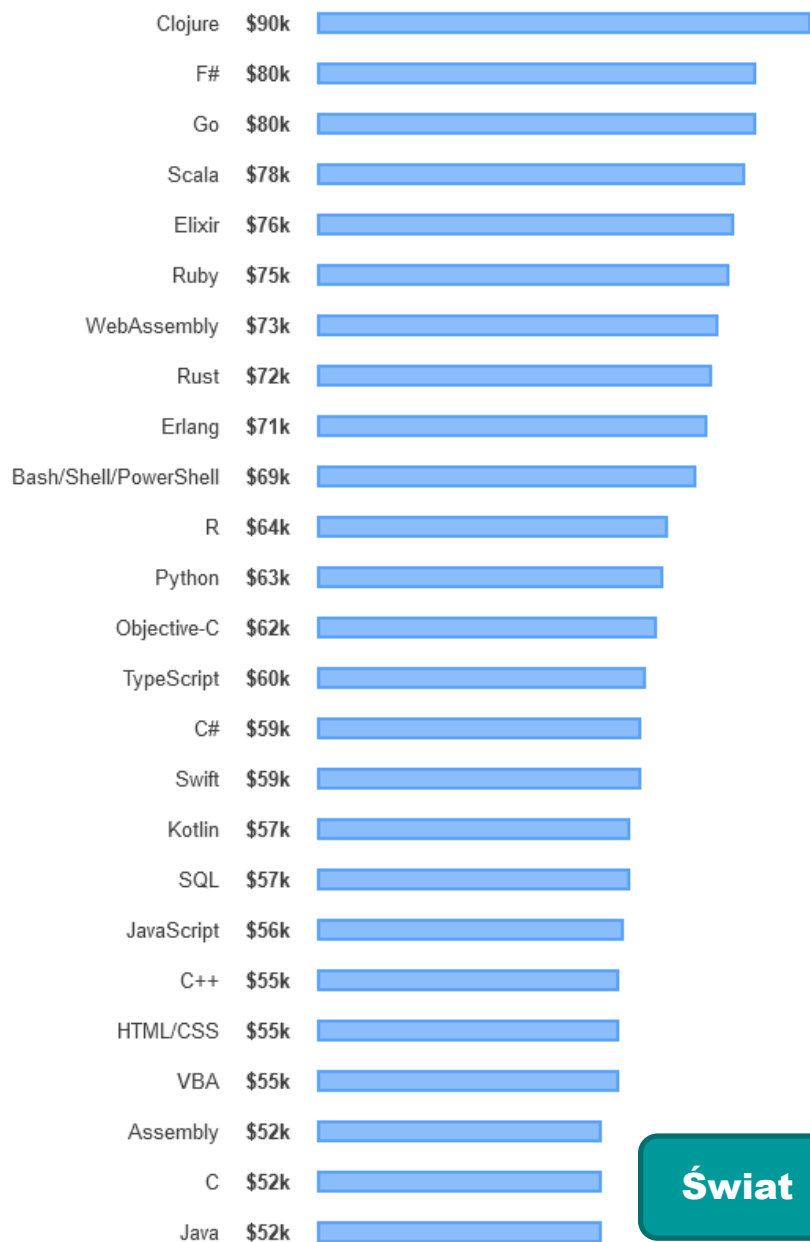
Języki najbardziej lubiane, przerażające i poszukiwane



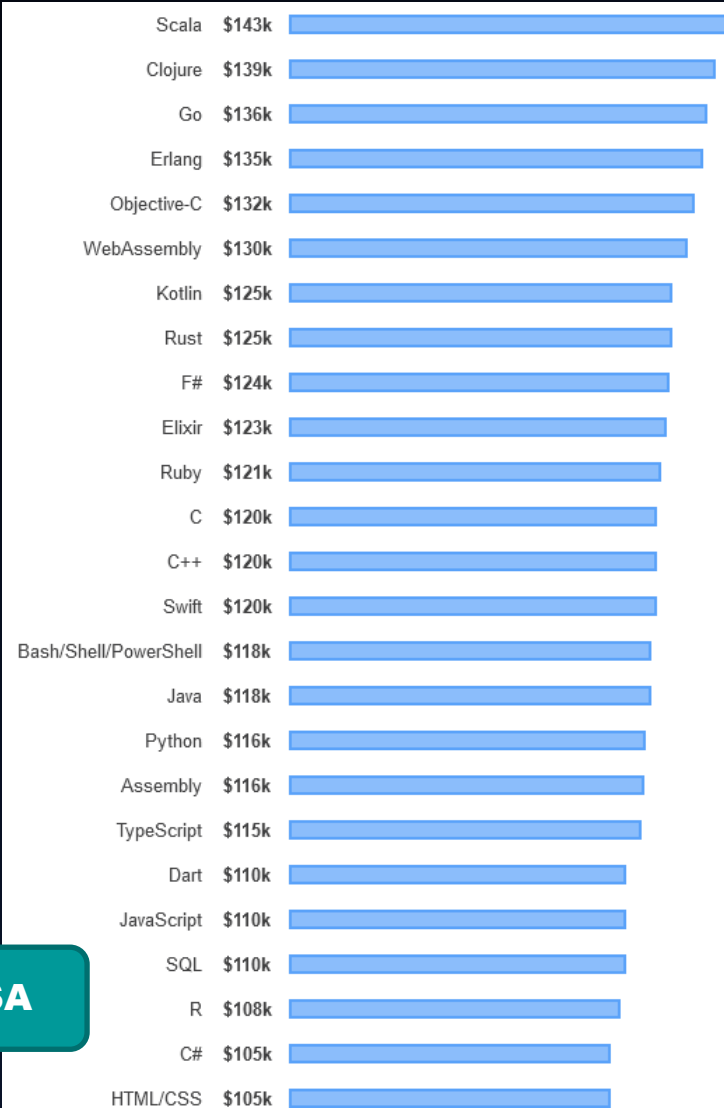
Najpopularniejsze środowiska programowania



Jakie języki są powiązane z najwyższymi zarobkami?

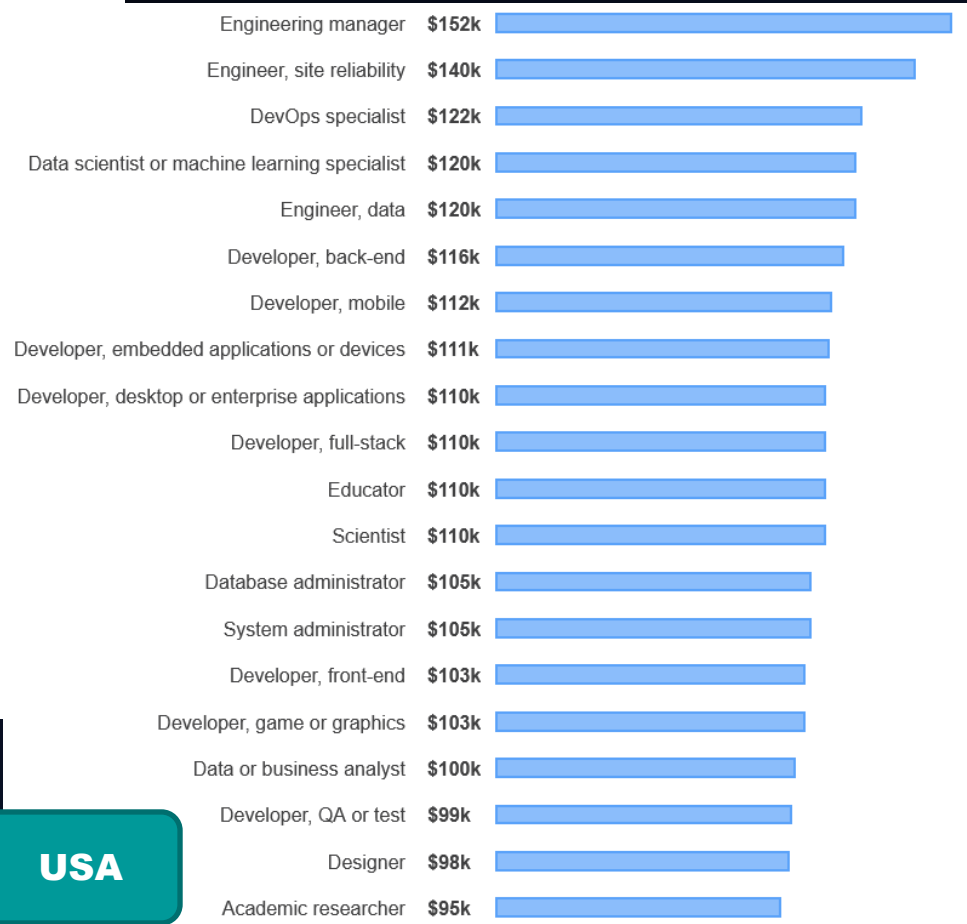
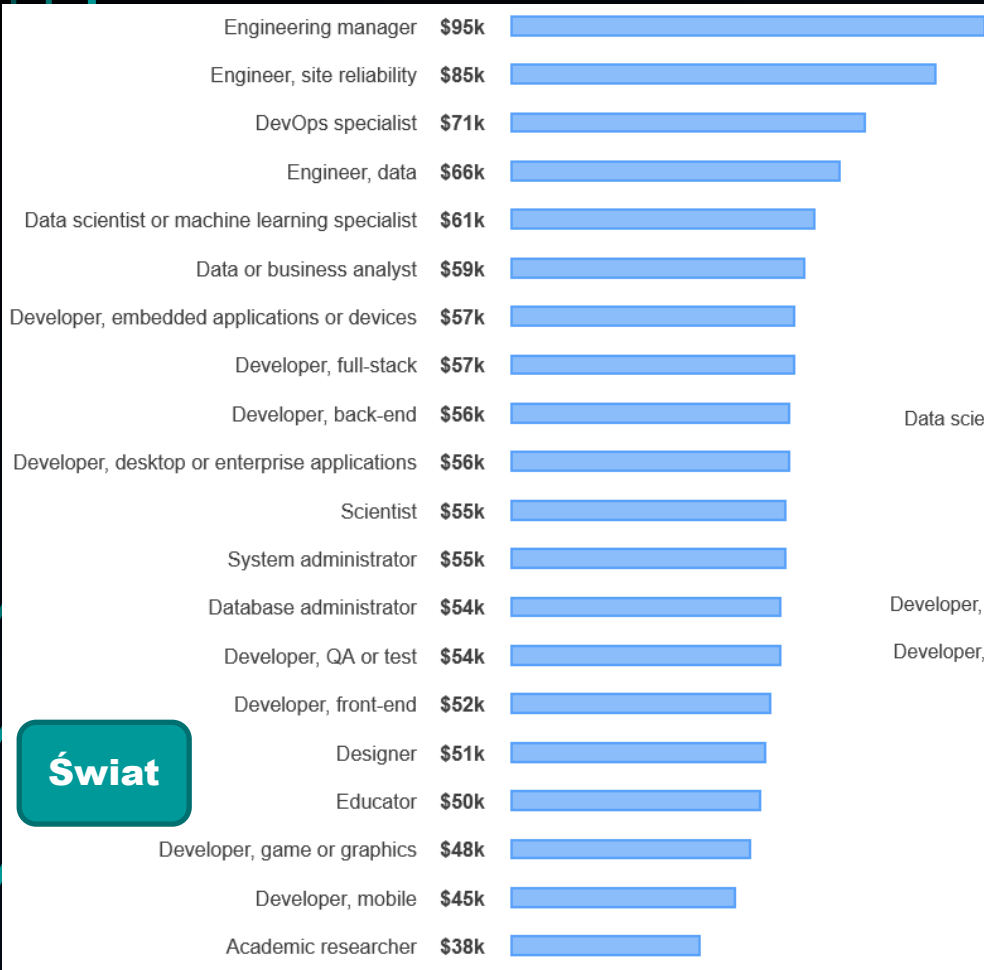


Świat

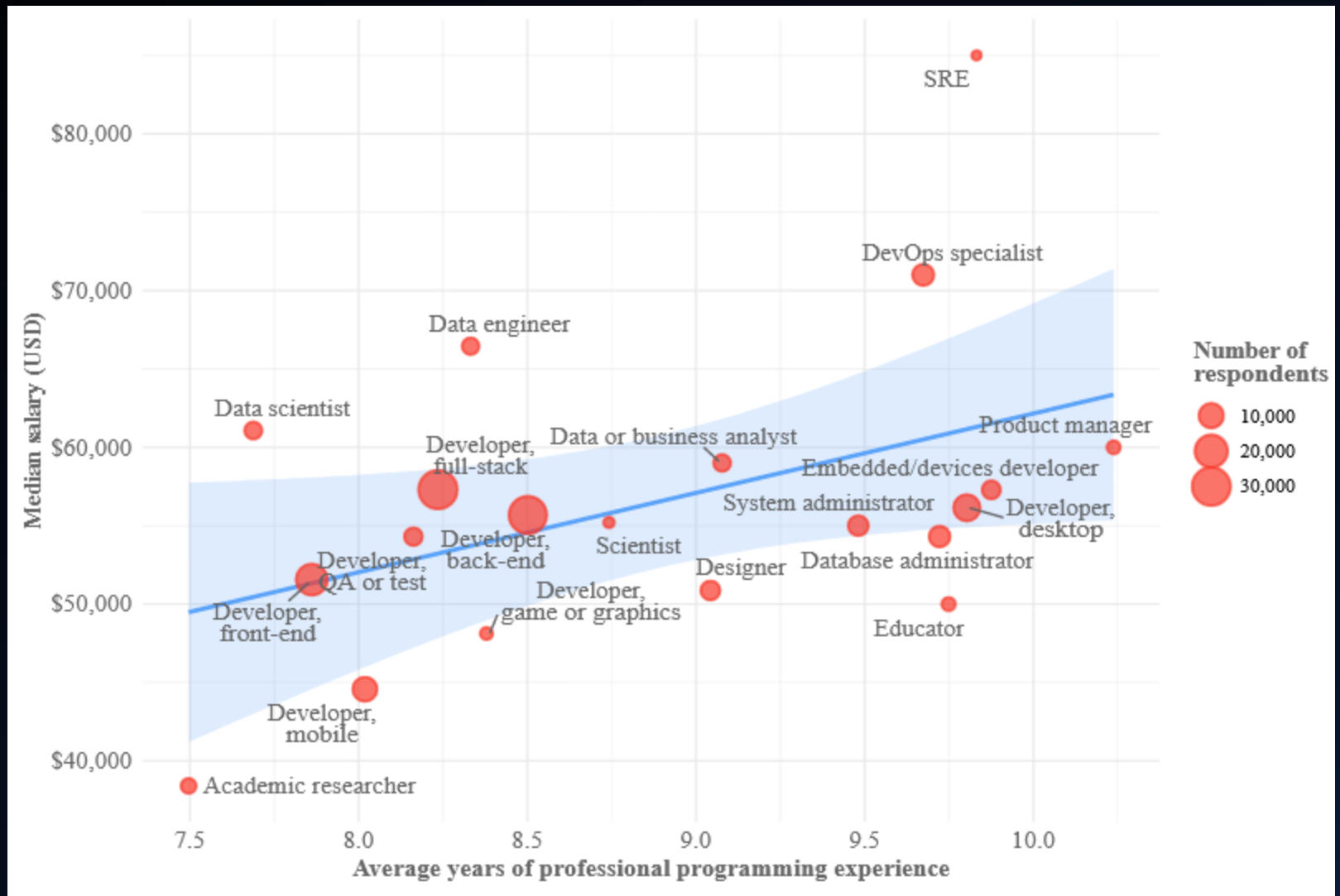


USA

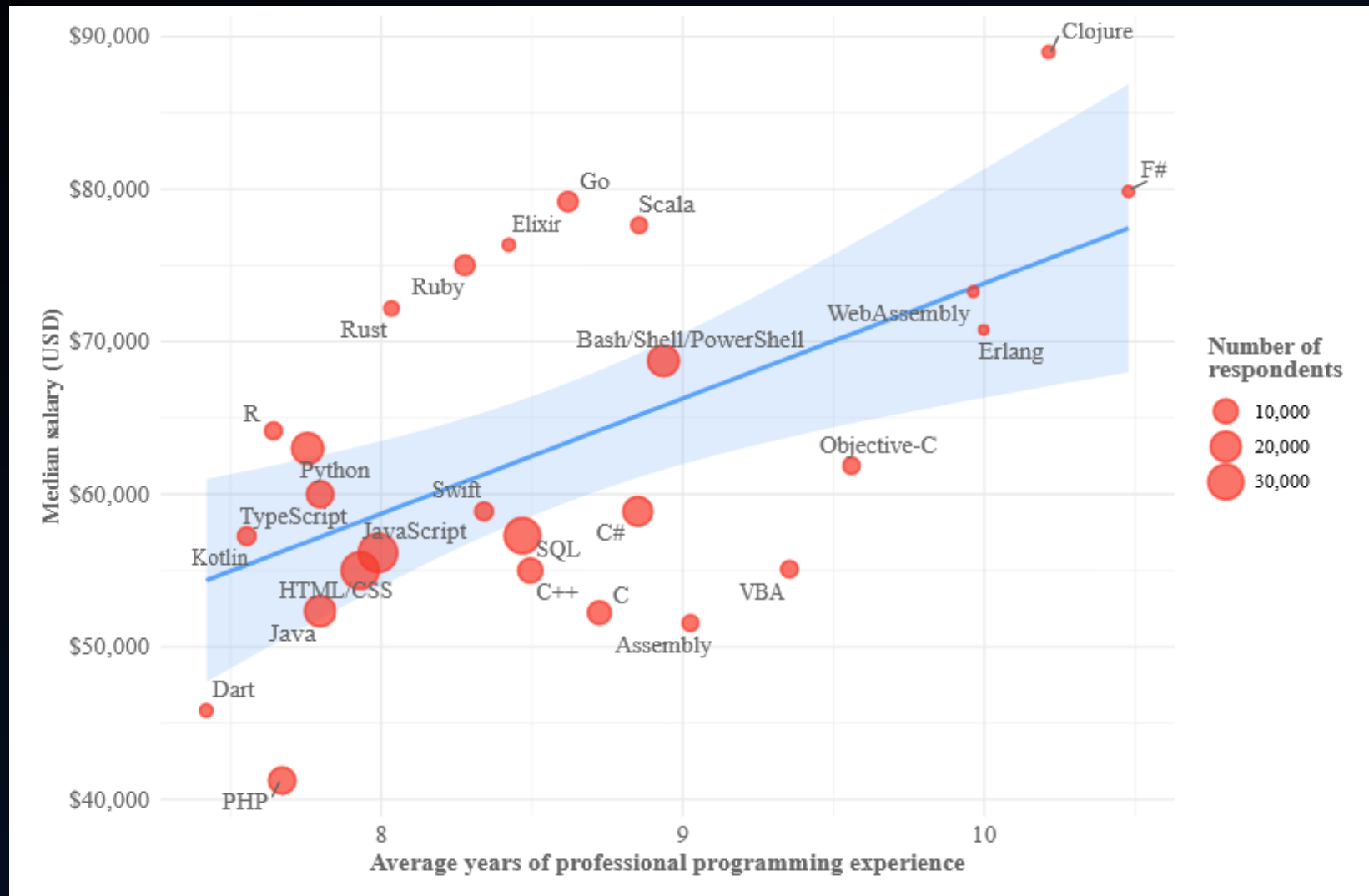
Wynagrodzenie według stanowiska programisty



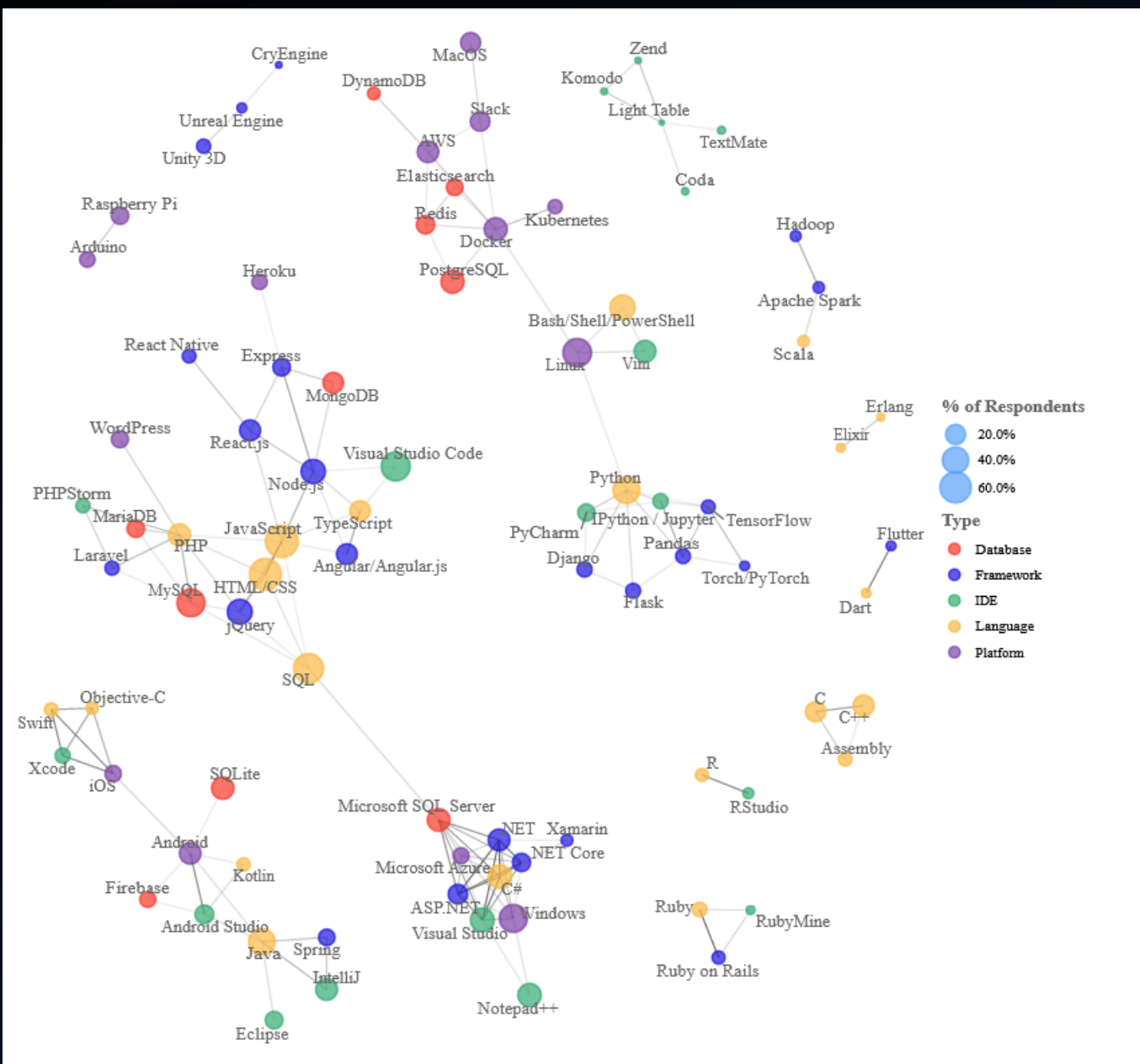
Wpływ doświadczenia i stanowiska programisty na wynagrodzenie



Wpływ doświadczenia i języka programowania na wynagrodzenie



Powiązanie języków i technologii



Muzyka dla skupienia się podczas kodowania

