

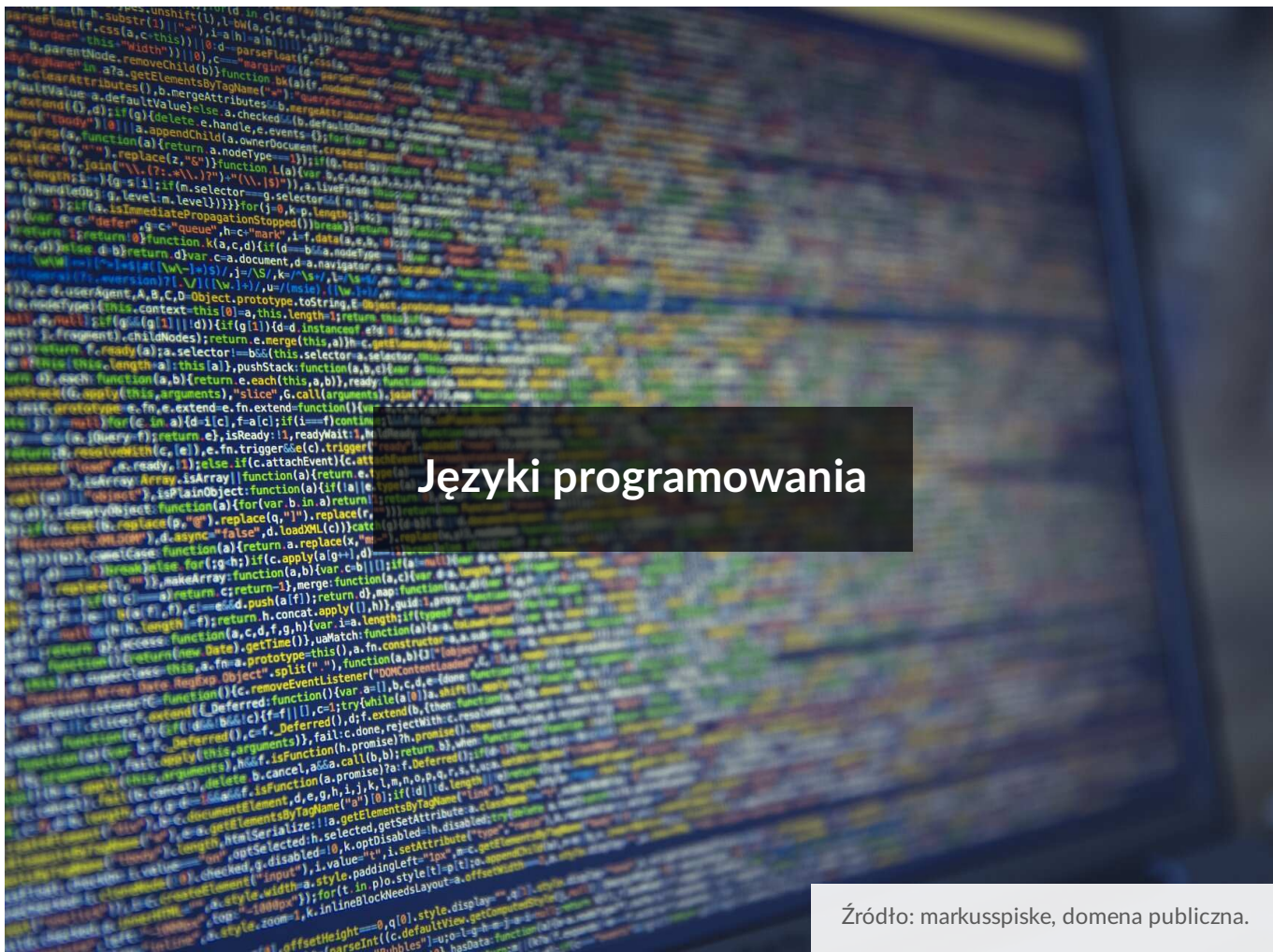
## Języki programowania

- Wprowadzenie
- Przeczytaj
- Audiobook
- Sprawdź się
- Dla nauczyciela

### Bibliografia:

- Źródło: Jarosław Deminet, *język programowania*, dostępny w internecie: [encyklopedia.pwn.pl](http://encyklopedia.pwn.pl) [dostęp 1.06.2021].
- Źródło: Edward Sapir, *Język*, [w:] tegoż, *Kultura, język, osobowość*, Warszawa 1978, s. 33.





## Języki programowania

Źródło: markusspiske, domena publiczna.

Jak możemy zdefiniować język? Teorii jest wiele. Jedna z nich mówi, że:

” Edward Sapir

### Język

(...) język jest w istocie doskonałym narzędziem ekspresji i komunikacji we wszystkich znanych społecznościach.

Źródło: Edward Sapir, *Język*, [w:] tegoż, *Kultura, język, osobowość*, Warszawa 1978, s. 33.

Jak jednak odnieść taką definicję do języków programowania? Encyklopedia PWN tak opisuje język programowania:

” Jarosław Deminet

### język programowania

(...) narzędzie do formułowania programów dla komputerów; jest językiem formalnym, którego składnia określa zasady zapisu programów (w sposób jednoznaczny i łatwy do analizy), a semantyka przypisuje programom ich interpretację (określa efekty działania programu zapisanego w języku programowania).

Źródło: Jarosław Deminet, *język programowania*, dostępny w internecie: [encyklopedia.pwn.pl](http://encyklopedia.pwn.pl) [dostęp 1.06.2021].

W tym e-materiale scharakteryzujemy wybrane języki programowania oraz zapoznamy się z ich klasyfikacją.

### Twoje cele

- Wskażesz najpopularniejsze języki programowania wraz z ich zastosowaniem.
- Wyjaśnisz najważniejsze terminy służące do opisywania języków programowania.
- Wymienisz różnice pomiędzy językami interpretowanymi i kompilowanymi.

# Przeczytaj

---

## Klasyfikacja języków programowania

Języki programowania możemy podzielić według [paradygmatów](#), czyli zbiorów pewnych cech.

### Paradygmaty kompilowane i interpretowane

Kod w języku kompilowanym musi zostać przetworzony przez program zwany kompilatorem, aby można było go uruchomić; przykłady języków: C, C++, Java, Go.

Program napisany w języku interpretowanym może być wykonywany bezpośrednio na kodzie źródłowym przez program zwany interpreterem, który wykonuje kod linijka po linijce; przykłady języków: Python, Ruby, JavaScript, PHP.



Logo języka Python

Źródło: tylko do użytku edukacyjnego.

### Paradygmaty imperatywne i deklaratywne

W języku imperatywnym korzystamy z instrukcji, aby bezpośrednio zmienić stan programu – mówimy, jak coś zrobić.

Przykłady to C++, Java, Lua. W języku deklaratywnym deklarujemy oczekiwany



rezultat (wskazujemy, co ma być zrobione).  
Przykłady: SQL, regex, Prolog.

Logo języka C++  
Źródło: domena publiczna.

## Paradygmaty obiektowe, funkcjonalne, proceduralne



Logo języka Java  
Źródło: tylko do użytku edukacyjnego.

W języku obiektowym operujemy na obiektach, czyli strukturach danych z pewnymi informacjami oraz interakcjami między sobą; przykłady to Java, Kotlin, C#.

W języku funkcjonalnym programy składają się z nałożonych na siebie matematycznych funkcji, które operują na zwracanych przez siebie rezultatach; przykłady to Haskell, Elixir, Clojure.

W językach proceduralnych programy składają się z kolejnych kroków, które należy wykonać, aby otrzymać pożądaną rezultat; przykłady takich języków programowania to C, C++, Python.

**Ważne!**

Przedstawiony podział jest bardzo dynamiczny i wiele z języków programowania można umieścić w kilku różnych kategoriach. Przykładowo Python, choć zazwyczaj traktowany jako język interpretowany, może zostać skompilowany do programu niewymagającego interpretera. W języku C++ możemy stosować zarówno programowanie obiektowe, funkcjonalne i proceduralne, jak również deklaratywne i imperatywne. Języki, które pozwalają na użycie więcej niż jednego paradygmatu, nazywamy **wieloparadygmatowymi**.

## Najważniejsze języki programowania

### Ważne!

W szkole podstawowej poznaliśmy już kilka języków programowania, np. HTML, Scratch, Logo czy Visual Basic.

Aktualnie jednym z najpopularniejszych języków jest **Java**. Znajduje zastosowanie we wszelkiego rodzaju aplikacjach internetowych przeznaczonych do wyświetlania w przeglądarce, jak również aplikacjach mobilnych.

Wciąż popularnym językiem jest język **C**, oraz pokrewny mu język **C++**.

Język C często jest używany do programowania urządzeń takich jak satelity czy urządzenia szpitalne, które często pracują na ograniczonych zasobach i nie mogą ulegać awariom.

C++ jest wykorzystywany w niemal każdej gałęzi informatyki, od gier komputerowych, przez programowanie systemów operacyjnych, aż do zaawansowanych symulacji fizycznych.

Językiem, którego popularność nieustannie rośnie, jest **Python**. Wśród jego zalet możemy wymienić przejrzysty zapis. Python jest także głównie językiem interpretowanym, dlatego po wprowadzeniu zmian w kodzie można bardzo szybko uruchomić program, co ułatwia tworzenie większych aplikacji. Jest on wykorzystywany między innymi do tworzenia gier czy oprogramowania serwerów.

Język **SQL** jest jednym z głównych języków służących do obsługi baz danych. Służy on zarówno do zapisywania danych, jak i do ich odczytywania, przy czym oferuje możliwości filtrowania według podanych kryteriów.

Język **JavaScript** służy do dodawania interaktywnych elementów do stron internetowych. Bardzo ważnym aspektem użycia JavaScript jest środowisko **Node.js**, za pomocą którego można stworzyć serwer w tym samym języku, co aplikację po stronie przeglądarki – bez Node.js JavaScript nie oferuje możliwości serwera.

**PHP** to język służący do tworzenia oprogramowania do serwerów. Chociaż w znacznej mierze został zastąpiony przez języki takie jak JavaScript czy Python, nadal jest wykorzystywany w projektach takich jak Wikipedia czy WordPress.

## Asemblery i kody maszynowe

### Co dzieje się, kiedy chcemy uruchomić program w języku kompilowanym?

Na początku, po sprawdzeniu kodu pod względem poprawności składni, kompilator tłumaczy go do **języka asemblera** (często nazywanego po prostu asemblerem), który jeszcze nie może zostać uruchomiony. Ten proces nazywa się **kompilacją**.

#### Ważne!

**Asembler** to program tłumaczący kod z języka asemblera na kod maszynowy. Technicznie rzecz biorąc, za język asemblera możemy uznać każdy język programowania, który bardzo ściśle odpowiada pojedynczym instrukcjom, jakie wykonuje procesor.

Dopiero po procesie zwanym **asemblacją** otrzymujemy kod maszynowy, który może zostać bezpośrednio odczytany i wykonany przez procesor. Ponieważ sam język asemblera jest bardzo nieczytelny, powstały kod maszynowy jest dla ludzi praktycznie niezrozumiały, jednak dla komputera jest to idealna forma. Wynika to z tego, że kod maszynowy to postać programu, która jest przeznaczona do bezpośredniego lub prawie bezpośredniego wykonania przez procesor i jest dopasowana do konkretnego typu procesora (wyrażona w postaci zrozumiałych dla niego kodów rozkazów i argumentów).

# Od kodu do programu

## 1 Napisanie kodu w języku programowania

Programista pracuje w edytorze tekstu lub w zintegrowanym środowisku programistycznym.

2

## Sprawdzenie poprawności kodu

Kod jest weryfikowany pod względem błędów składniowych przez program zwany parserem.

## 3 Kompilacja

Kompilator przekłada kod napisany przez programistę do języka odpowiadającego instrukcjom procesora.

4

## Asemblacja

Program (**assembler**) tworzy kod maszynowy, który może zostać bezpośrednio wykonany przez komputer.

## 5 Gotowy program

Program jest gotowy do uruchomienia.

## Słownik

### **paradygmat programowania**

zbiór cech charakteryzujących grupy języków programowania

### **język wieloparadygmatowy**

język, który może zostać przyporządkowany do kilku różnych paradygmatów

### **assembler**

program służący do tłumaczenia kodu z języka assemblera na kod maszynowy

### **kompilacja**

proces tłumaczenia kodu z języka programowania na język assemblera

### **język assemblera**

każdy język, który jest bardzo podobny do poszczególnych instrukcji wykonywanych przez procesor

### **kod maszynowy**

kod, który jest wykonywany przez procesor; jest to ostateczna forma programu w języku kompilowanym, która może zostać uruchomiona

### **asemblacja**

proces tłumaczenia kodu z języka assemblera na kod maszynowy

### **język wysokiego poziomu**

język, którego składnia i słowa kluczowe mają ułatwić rozumienie kodu programu przez człowieka; nim procesor komputera zrozumie taki kod, należy poddać go

procesowi kompilacji; do języków wysokiego poziomu zalicza się m.in. Pascal, C, C++, Logo i BASIC

### **język niskiego poziomu**

język programowania, w którym jednej operacji elementarnej odpowiada najczęściej jedna operacja wykonywana przez procesor

### **język skryptowy**

język programowania służący do kontrolowania danej aplikacji i interpretowany wewnątrz niej

# Audiobook

---

## Polecenie 1




Zapoznaj się z audiobookiem i wykonaj ćwiczenie.

## Ćwiczenie 1

## Polecenie 2

# Sprawdź się

---

Pokaż ćwiczenia:   

## Ćwiczenie 1



Dokończ zdanie.

**W języku proceduralnym...**

- nakładamy na siebie funkcje matematyczne, których wynik jest rezultatem programu.
- opisujemy kolejne kroki, jakie program musi wykonać.
- operujemy na obiektach.

## Ćwiczenie 2



Wskaż poprawną odpowiedź.

**Wszystkie języki programowania mogą być przyporządkowane tylko do jednego paradygmatu.**

- Fałsz.
- Prawda.

### Ćwiczenie 3



Dokończ zdanie.

Środowisko, które umożliwia tworzenie oprogramowania serwerowego w języku JavaScript to...

Node.js.

SQL.

PHP.

### Ćwiczenie 4



Uzupełnij tekst.

Kod, który jest wykonywany przez procesor, to .

Program tłumaczący kod na język assemblera to .

Proces tłumaczenia kodu z języka assemblera na kod maszynowy to .

paradygmat

aseblacja

kompilacja

assembler

kod maszynowy

kompilator

### Ćwiczenie 5



Ułóż we właściwej kolejności kroki prowadzące do uruchomienia kodu w języku kompilowanym.

kompilacja



aseblacja



uruchomienie programu



napisanie kodu źródłowego



## Ćwiczenie 6



Wskaż, które zdania są prawdziwe, a które fałszywe.

Stwierdzenie	Prawda	Fałsz
Kod maszynowy jest łatwy w zrozumieniu dla człowieka.	<input type="radio"/>	<input type="radio"/>
Procesor może bezpośrednio wykonać kod maszynowy.	<input type="radio"/>	<input type="radio"/>
Kod maszynowy to każdy język programowania.	<input type="radio"/>	<input type="radio"/>
Kod maszynowy jest wynikiem procesu asemblacji.	<input type="radio"/>	<input type="radio"/>

## Ćwiczenie 7



Wskaż poprawne możliwości dokończenia zdania.

Język SQL jest wykorzystywany do...

tworzenia oprogramowania serwerowego.

zapisywania informacji w bazach danych.

odczytywania informacji w bazach danych.

dodawania elementów interaktywnych do stron internetowych.

## Ćwiczenie 8



Dokończ zdanie.

Jedną z zalet języka C++ jest...

łatwość w nauce.

przenośność.

efektywne zarządzanie zasobami komputera.

# Dla nauczyciela

---

**Autor:** Zespół autorski [Contentplus.pl](http://Contentplus.pl) sp. z o.o.

**Przedmiot:** Informatyka

**Temat: Języki programowania**

**Grupa docelowa:**

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres podstawowy

**Podstawa programowa:**

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres podstawowy. Uczeń:

2) do realizacji rozwiązań problemów prawidłowo dobiera środowiska informatyczne, aplikacje oraz zasoby, wykorzystuje również elementy robotyki;

## **Kształtowane kompetencje kluczowe:**

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

## **Cele operacyjne (językiem ucznia):**

- Wskażesz najpopularniejsze języki programowania wraz z ich zastosowaniem.
- Wyjaśnisz najważniejsze terminy służące do opisywania języków programowania.
- Wymienisz różnice pomiędzy językami interpretowanymi i kompilowanymi.

## **Strategie nauczania:**

- konstruktywizm;
- konektywizm.

## **Metody i techniki nauczania:**

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- metody aktywizujące.

## **Formy pracy:**

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

## Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda.

## Przebieg lekcji

### Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Języki programowania”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj”.

### Faza wstępna:

1. Przedstawienie tematu i celów zajęć.
2. **Rozpoznanie wiedzy uczniów.** Uczniowie tworzą pytania dotyczące tematu zajęć, na które odpowiedzą w trakcie lekcji.

### Faza realizacyjna:

1. **Praca z tekstem.** Nauczyciel ocenia, na podstawie informacji na platformie, stan przygotowania uczniów do zajęć. Jeżeli jest ono niewystarczające prosi o ciche zapoznanie się z treścią w sekcji „Przeczytaj”.
2. **Praca z multimedium.** Nauczyciel wyświetla zawartość sekcji „Audiobook”. Uczniowie wspólnie zapoznają się z treścią audiobooka. Każdy z uczniów zapisuje w zeszycie trzy przykłady projektów, jakie chciałby zrealizować w przyszłości oraz propozycje języków programowania, których chcą użyć do ich realizacji. Wybrani uczniowie przedstawiają swoje propozycje na forum klasy i uzasadniają wybór języków programowania.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują indywidualnie ćwiczenia nr 1-4, a następnie dzielą się wynikami swojej pracy z kolegą lub koleżanką.

### **Faza podsumowująca:**

1. Nauczyciel wyświetla na tablicy temat lekcji i cele zawarte w sekcji „Wprowadzenie”. W kontekście ich realizacji podsumowuje przebieg zajęć, a także wskazuje mocne i słabe strony pracy uczniów.
2. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności, omawia ewentualne problemy podczas rozwiązania ćwiczeń.

### **Praca domowa:**

1. Uczniowie wykonują ćwiczenia 5-8 z sekcji „Sprawdź się”.

### **Wskazówki metodyczne:**

- Treści w sekcji „Przeczytaj” można wykorzystać jako podsumowanie i utrwalenie wiedzy uczniów.