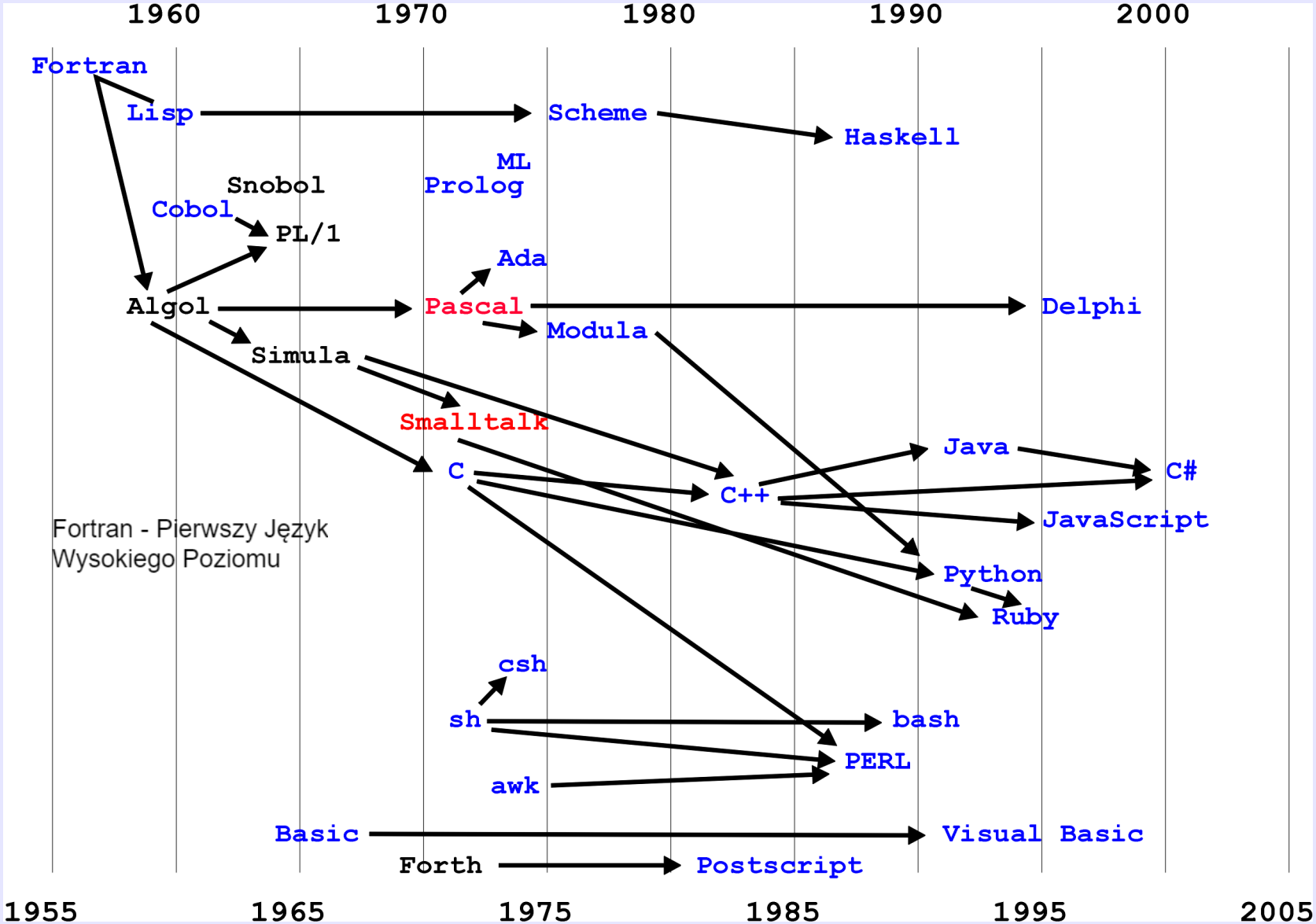
Tryb zaliczenia kursu

- Ostateczny wynik procentowy zostanie przeliczony na ocenę końcową zgodnie z następującą tabelą:

0-49 %	niedostateczny
50-59 %	dostateczny
60-69 %	plus dostateczny
70-77 %	dobry
78-85 %	plus dobry
86-94 %	bardzo dobry
95-100 %	celujący

Najważniejsze języki programowania



Algorytmy i programy 5

- **algorytm**: ściśle opisany sposób postępowania w celu rozwiązania pewnego problemu
- **program**: algorytm zapisany w sformalizowany sposób, umożliwiający wykonanie algorytmu przy pomocy komputera
- program
  - przyjmuje **dane wejściowe**, przetwarza je i produkuje **dane wyjściowe**
  - najczęściej ma postać ciągu **instrukcji**, określających sposób przetwarzania danych

Języki programowania 6

- **język programowania**: formalnie zdefiniowany sposób zapisywania algorytmów, umożliwiający pisanie programów
- język programowania określa
  - jakiego typu dane mogą być przetwarzane i jak te dane są zapisywane
  - jakie **instrukcje** mogą być stosowane i jakim podlegają regułom

Techniki programowania 7

- **technika programowania**: pewien zestaw mniej lub bardziej precyzyjnych reguł, których stosowanie ułatwia osiągnięcie określonych celów w trakcie tworzenia oprogramowania.
- konkretną technikę najłatwiej stosować w języku specjalnie dla niej zaprojektowanym

Rozwój technik programowania

Z perspektywy historycznej główne etapy rozwoju programowania to pojawianie się nowych technik programowania

- programowanie linearne
- programowanie proceduralne
- programowanie funkcyjne (funkcjonalne)
- programowanie strukturalne
- programowanie w języku logiki (logic programming)
- programowanie bazujące na obiektach
- programowanie obiektowo orientowane
- programowanie orientowane zdarzeniami
- programowanie generyczne
- programowanie wielowątkowe i rozproszone
- programowanie wieloparadygmatowe
- metaprogramowanie, w tym programowanie refleksywne i programowanie generatywne

Rozwój technik programowania 9

- nowa technika $->$ nowe języki
- techniki programowania i związane z nią klasy języków: na przykład języki proceduralne, funkcyjne czy obiektowe.
- programowania wieloparadygmatowe

Przyczyny rozwoju języków i technik programowania

Wszystkie tendencje i trendy w rozwoju języków programowania można streścić w dewizie:

Małe jest piękne!

Kryteria oceny technik i języków programowania

Użyteczność języków programowania oceniana jest przy użyciu różnych kryteriów. Najczęściej stosowane kryteria to:

- **Efektywność**: programy w danym języku wykonują się bardzo szybko i zajmują mało pamięci
- **Produktywność**: programy w danym języku można szybko napisać i uruchomić
- **Odporność na błędy**: struktura języka umożliwia wyłapywanie znacznej ilości błędów na etapie pisania programu, a przed etapem testowania programu
- **Stabilność**: język ułatwia tworzenie oprogramowania, w którym stosunkowo rzadko pojawiają się błędy w trakcie wykonywania, a jeśli się pojawiają, są dobrze udokumentowane, więc łatwo ustalić ich przyczynę
- **Analizowalność**: o programie relatywnie łatwo jest udowodnić, że robi rzeczywiście to, czego się od niego oczekuje
- **Klarowność**: język ułatwia pisanie programów, które w znacznym stopniu same się dokumentują
- **Trwałość**: raz przygotowane oprogramowanie może być długo użytkowane w różnych projektach bez konieczności przepisywania go od nowa.

Programowanie w języku wewnętrznym

Tak na prawdę każdy komputer rozumie tylko jeden język: język wewnętrzny swojego procesora.

- program w języku wewnętrznym: ciąg zer i jedynek kodujących rozkazy procesora oraz dane
- programowanie w języku wewnętrznym: bezpośrednie wprowadzanie zer i jedynek do pamięci

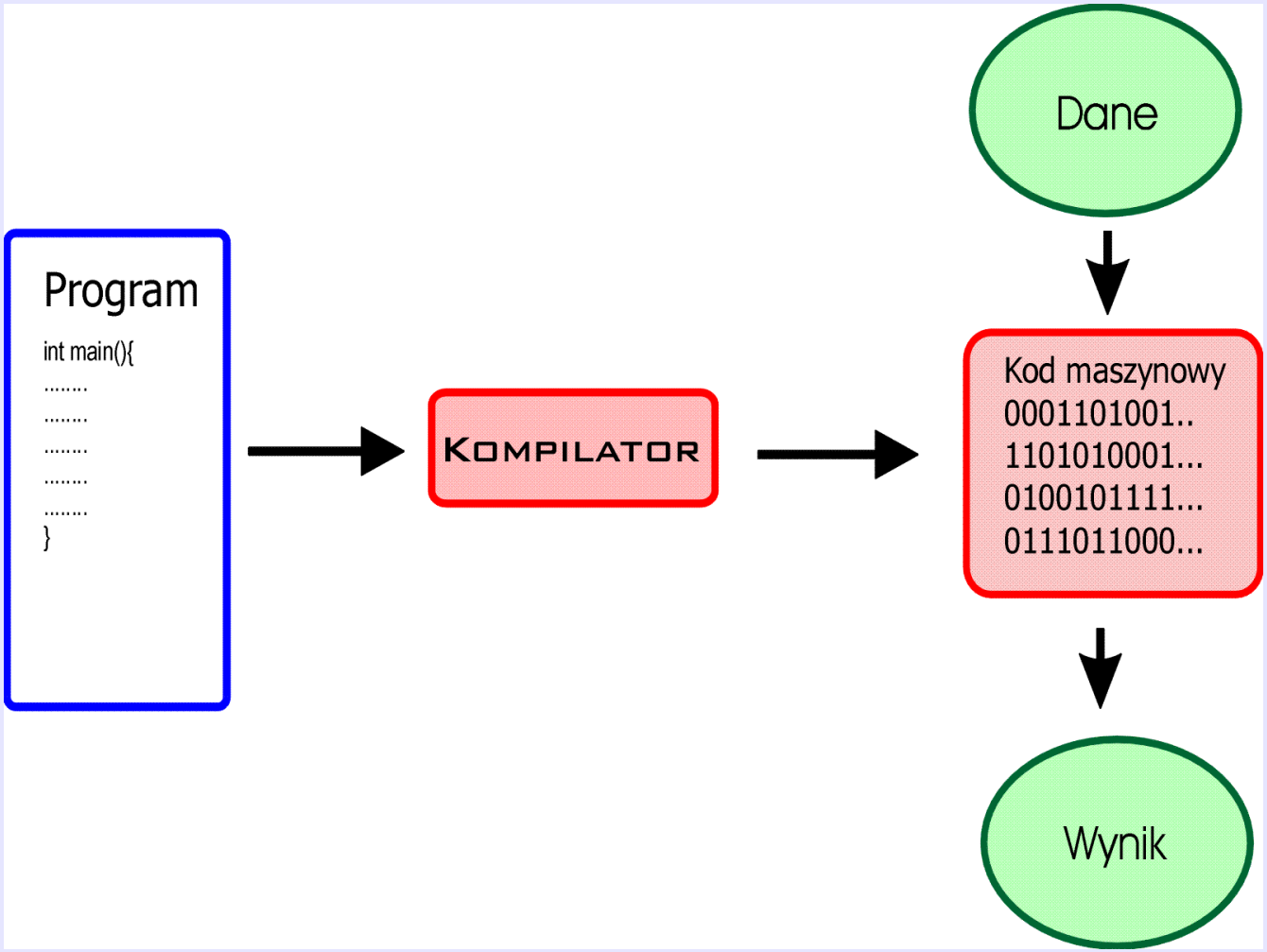
Programowanie linearne 15

Choć w zasadzie każdy język wewnętrzny w pewnym stopniu wspiera programowanie proceduralne, programowanie w języku wewnętrznym ze względu na stopień komplikacji w większości ograniczało się do programowania linearnego.

Programowanie linearne to najprostsza technika programowania polegająca na przedstawieniu programu jako ciągu kolejno wykonywanych instrukcji

Języki wysokiego poziomu. Translatory

- **Język wysokiego poziomu** to język, który zaprojektowany jest z myślą o łatwości pisania w nim programów, a nie zgodności z architekturą konkretnego procesora.
- Aby wykonać program napisany w języku wysokiego poziomu potrzebny jest program pośredniczący, zwany **translatorem**.

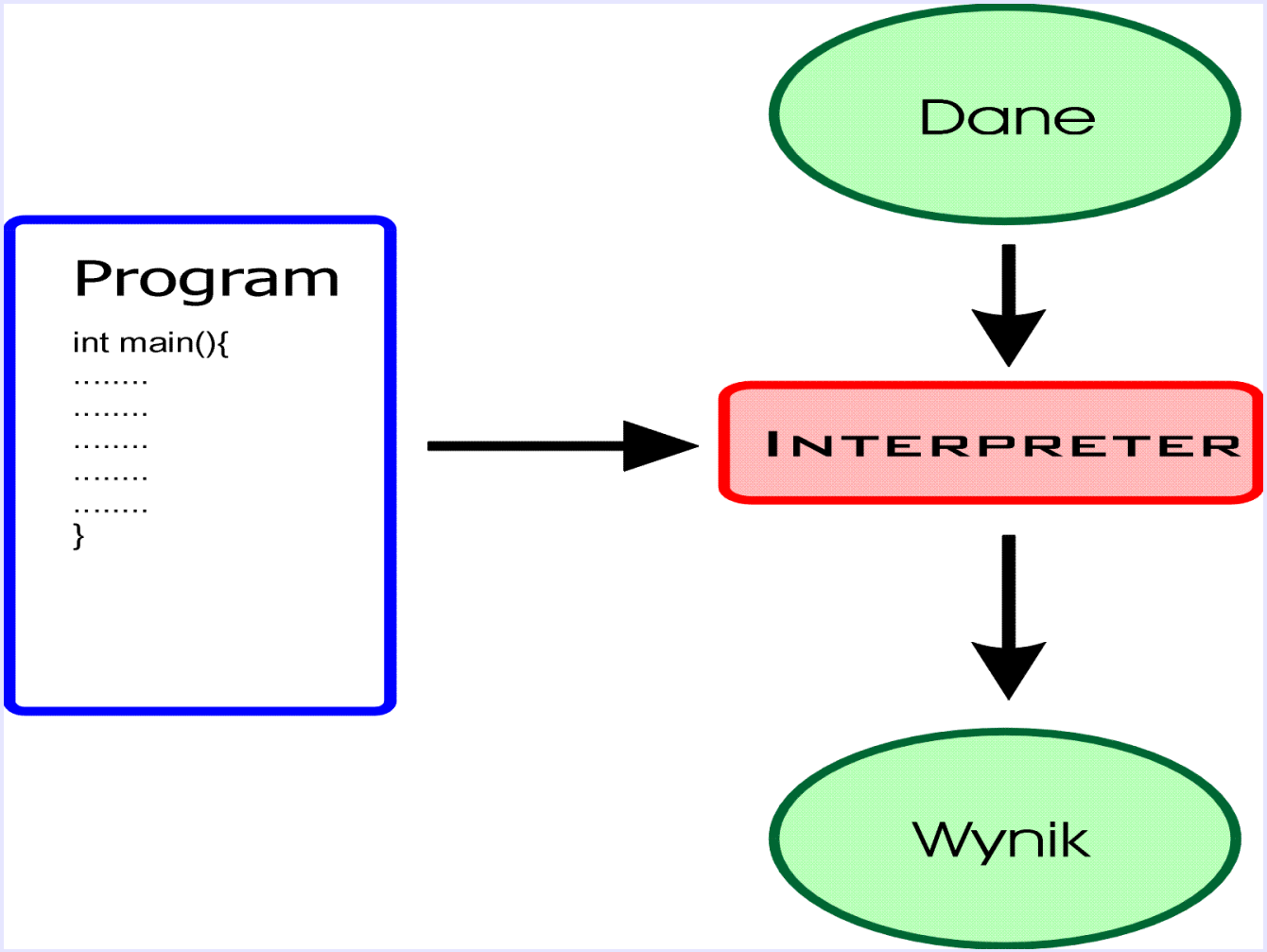


Działanie kompilatora

Kompilatory

Kompilator to program, który na wejściu przyjmuje tekst programu w języku wysokiego poziomu, a na wyjściu dostarcza jego odpowiednik zapisany w języku maszynowym.

Interpretery

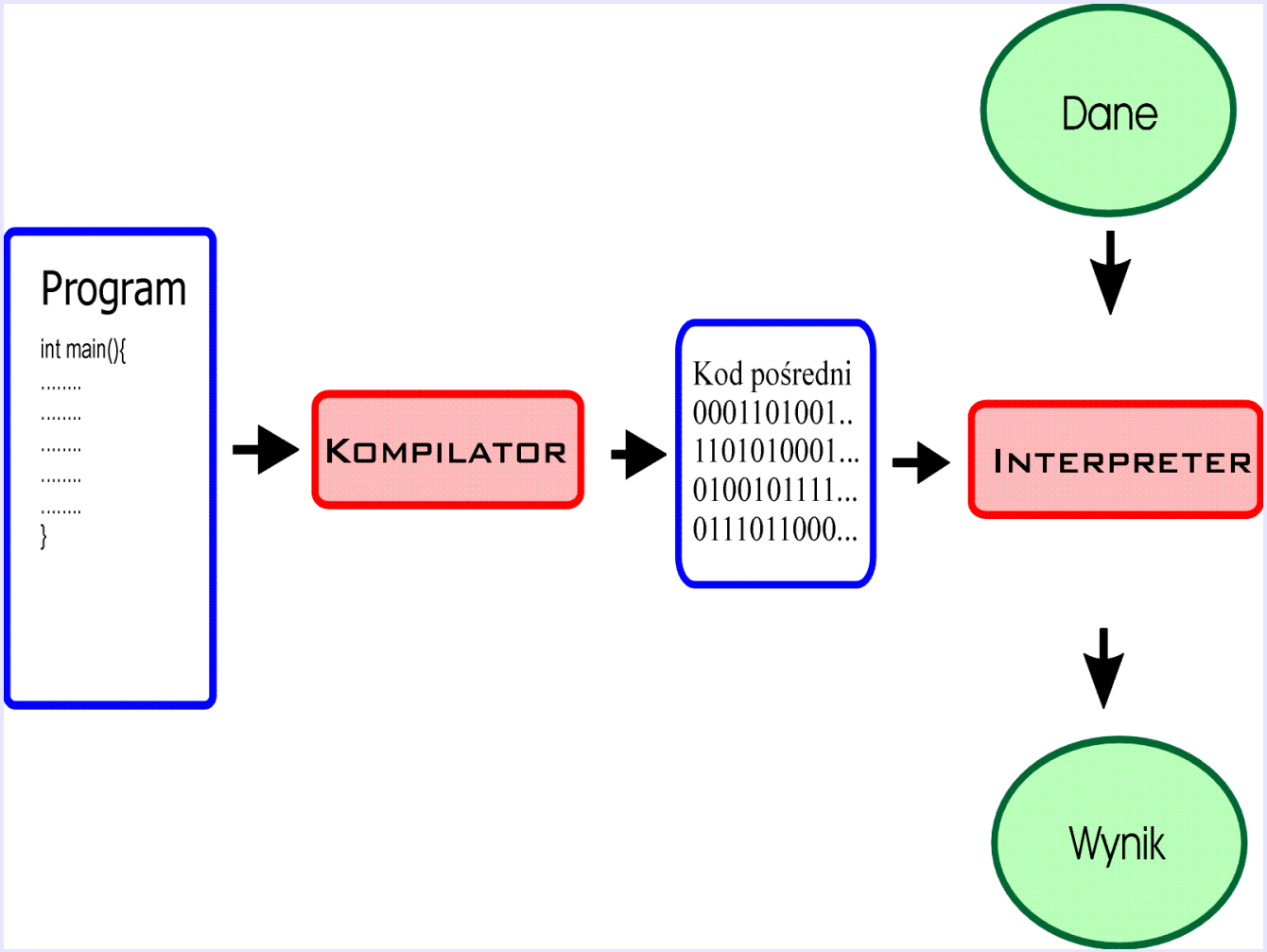


Działanie interpretera

Interpretery

Interpreter to program, który na wejściu przyjmuje tekst programu w języku wysokiego poziomu oraz dane wejściowe dla tego programu, a na wyjściu dostarcza wynik wykonania programu przeczytanego na wejściu.

Rozwiązania pośrednie



Rozwiązanie pośrednie

Rozwiązania pośrednie

W rozwiązaniu pośrednim program najpierw jest tłumaczony przez kompilator do formy pośredniej, a następnia forma pośrednia jest interpretowana przez interpreter.

Zalety i wady kompilatorów i interpreterów.

	Zalety	Wady
Kompilator	• Wysoka efektywność • Końcowy program może być wykonywany na komputerze, na którym nie ma kompilatora	• Relatywnie niska produktywność
Interpreter	• Relatywnie wysoka produktywność	• Niska efektywność • Na komputerze, na którym ma być wykonywany program musi być zainstalowany interpreter

Pierwszy język wysokiego poziomu: Fortran

Dla porzeb obliczeń naukowych stworzono **Fortran** (FORmula TRANslator).
Z historii Fortranu:

- IBM Laboratories, Johna Backus, opracowany dla IBM 704.
- projekt w 1954, kompilator (Fortran I) pojawił w 1957.
- opracowanie kompilatora zajęło 18 roboczo−lat.
- "tradycyjny" zapis formuł matematycznych oraz przechowywanie zmiennych w tablicach
- instrukcja pętli.
- formatowane instrukcje wejścia−wyjścia
- w Fortranie II (1958) możliwość niezależnej kompilacji podprogramów.
- Fortran IV (1962): deklaracje typu zmiennej, użycie nazwy podprogramu jako parametru innego podprogramu.
- Fortran IV powszechnie używany do 1978, potem Fortran 77, Fortran 90, Fortran 95, Fortran 2003

Programowanie proceduralne

Programowanie proceduralne polega na

- wyodrębnieniu w problemie szeregu mniejszych, powtarzających się podproblemów
- przedstawieniu rozwiązania każdego z podproblemów w postaci samodzielnego małego programu, tak zwanego **podprogramu**
- rozwiązaniu właściwego problemu poprzez odwołania do podprogramów

- Programowanie proceduralne jest powszechnie stosowane do dziś, współistniejąc z innymi, bardziej zaawansowanymi technikami programowania.
- Obok nazwy podprogram stosuje się również nazwę **procedura** oraz **funkcja**.

Programowanie funkcyjne

Język Lisp jest pierwszym językiem programowania wspierającym technikę programowania funkcyjnego.

Programowanie funkcyjne to technika programowania, w której

- funkcja traktowana jest na równi z danymi i może być argumentem innych funkcji
- poza t.zw. wyrażeniem warunkowym, jedyną stosowaną instrukcją jest instrukcja wyliczenia funkcji

Programowanie funkcyjne

- programowanie funkcyjne cechuje się m.in.
  - brakiem zmiennych
  - traktowaniem programu jako funkcji skonstruowanej z funkcji elementarnych poprzez superpozycję, wyrażenie warunkowe i definicje rekurencyjne.

- najważniejszą zaleta: wysoki poziom analizowalności programu
- technika ciągle rozwijana tam gdzie kluczowa jest pewność poprawnej implementacji algorytmu

Programowanie strukturalne 37

Algol zapoczątkował technikę programowania strukturalnego

Programowanie strukturalne to technika programowania, w której uwypukla się strukturę logiczną programu i danych poprzez stosowanie instrukcji strukturalnych, instrukcji złożonych, zmiennych loklanych oraz grupowanie danych w t.zw. strukturach.

Niepodważalny jest wpływ Algolu na swoich następców: Pascal i C, które filozofię programowania strukturalnego przejęły z właśnie z Algolu.

Pierwszy język wysokiego poziomu dla profesjonalistów: C

Z początkiem lat 70−tych XX wieku Dennis Ritchie Z Bell Labs firmy AT&T opracował **język C**.
Cechy języka C:

- w pełni strukturalny,
- bardzo zwięzły; blok oznaczany przez { i } w miejsce **begin** i **end**,
- szereg instrukcji bliskich typowym instrukcjom języka maszynowego, np. i++, ++i, i+=5,
- daje prawie nieograniczone możliwości dostępu do pamięci, poprzez wprowadzenie **arytmetyki wskaźników**.

Pierwszy język wysokiego poziomu dla profesjonalistów: C

Olbrzymi sukces języka C w znacznej części spowodowany był faktem, że Ritchie przepisał w języku C system operacyjny Unix, a kompilator języka C był rozpowszechniany razem z Unixem.

Programowanie w języku logiki

Programowanie w języku logiki to specyficzna technika programowania, która opiera się na deklarowaniu faktów, a do rozwiązywania problemów wykorzystuje mechanizmy automatycznego dowodzenia twierdzeń.

Prolog

- Język stworzony w 1972 roku przez francuskiego informatyka Alaina Colmerauera.
- Zaprojektowany od podstaw do programowania w języku logiki.
- Pierwotnie przeznaczony do analizy zdań w językach naturalnych.

Programowanie obiektowe 44

Język Simula stworzył podwaliny pod programowanie obiektowe, ugruntowane później w językach Smalltalk i C++.

Programowanie obiektowe to technika programowania, która umożliwia rozszerzanie języka programowania tak, by w naturalny sposób można było w nim odzwierciedlić tak dane jak i operacje na nich pojawiające się w rozwiązywanym problemie.

- W technice programowania obiektowego wyróżnia się dwie odmiany:
  - elementarna: programowanie bazujące na obiektach
  - zaawansowana: programowanie obiektowo orientowane

- Programowanie obiektowe bazujące na obiektach charakteryzuje się gromadzeniem danych i operujących na nich funkcji w t.zw. obiektach oraz łączeniem obiektów o wspólnych cechach w tak zwane klasy.
- W programowaniu obiektowo orientowanym dodatkowo wykorzystuje się t.zw. polimorfizm dynamiczny, czyli zróżnicowane zachowanie obiektów w zależności od ich miejsca w tak zwanej hierarchii dziedziczenia.

C++

W roku 1986 Bjarne Stroustrupa pracujący w AT&T Bell Laboratories opracował język C++.

Bjarne Stroustrupa można odwiedzić pod adresem
http://www.research.att.com/~bs/