1.3 Geschichte der Programmiersprachen

50er Jahre

- erste Definition höherer Programmiersprachen
- Effizienz maßgebliches Designziel
- FORTRAN (Backus) als Sprache für wissenschaftliches Rechnen
  (komplexe Berechnungen, einfachen Daten), Arrays, Schleifen, IF-THEN-ELSE
- COBOL als Sprache für Business-Anwendungen
  (einfache Berechnungen, große Datenmengen)
  Records, Trennung von Datenstrukur/Ausführung, Ausgabeformatierung
- ALGOL (Naur, Backus) Vorläufer aller imperativer Sprachen
  Blockstruktur, Rekursion, Wertparameter, Typdeklaration bei Variablen
- Nachweis der technischen Machbarkeit und des Nutzens höherer Sprachen
- Neu: Modularisierung durch Unterprogrammtechnik, Zugriff auf globale Daten
  durch Umgebung (FORTRAN), Blockstruktur (ALGOL 60), Files (COBOL)
- FORTRAN(90) und COBOL noch immer sehr verbreitet!

1.3 Geschichte der Programmiersprachen

Frühe 60er Jahre

- erste Versuche mit nicht-imperativen Sprachen
- LISP (McCarthy) basiert auf Theorie rekursiver Funktionen, erste funktionale Programmiersprache
- APL für Vektor- und Matrixmultiplikationen
- SNOBOL4 (Griswold) gestattet Stringmanipulationen und Pattern Matching, Unterstützung für deklaratives Programmieren
- Diese Sprachen erfordern komplexe dynamische Ressourcenverwaltung, erfolgreich in spezifischen Anwendungen
- Erkennen der Bedeutung von Symbolmanipulation

1.3 Geschichte der Programmiersprachen

Späte 60er Jahre

- PL/I – Versuch von IBM erfolgreiche Sprachkonstrukte zu vereinen
  Module, Blöcke, dynamische Datenstrukturen
  Neu: Exception Handling, Multitasking
  Nachteile: Mangel an Uniformität, unausgereifte Teile, langsam
- ALGOL 68 – Nachfolger von ALGOL 60
  Orthogonalitätsprinzip (vollständige Integration von Sprachkomponenten)
  erste Sprache mit formaler Spezifikation, kaum populär, wenig nutzerfreundlich
- SIMULA 67 (Nygaard, Dahl) – erste Sprache mit Klassenkonzept
  Hierarchien zur Strukturierung von Daten und Prozeduren
  beeinflusste alle modernen Sprachen
- BASIC (Kemeny, Kurtz, Beginners All-Purpose Instruction Code)
  keine neuen Konstrukte,
  beliebt wegen Einfachheit und Effizienz, interaktiver Programmierstil
- Pascal (Wirth) – dient dem Erlernen strukturierter Programmierung
  Verfügbarkeit auf PCs und Einfachheit erzeugten großen Erfolg

1.3 Geschichte der Programmiersprachen

70er Jahre

- Fokus auf Zuverlässigkeit und Wartbarkeit
- abstrakte Datentypen, Module, Typisierung, Korrektheitsbeweise, Exceptions
- CLU – Datenabstraktion durch Cluster-Mechanismus
  Mesa – Pascal-Erweiterung um Module
  Concurrent Pascal, Euclid – Pascal-Fortentwicklung mit abstrakten Datentypen
- C (Ritchie)
  große Mächtigkeit, Effizienz für Systemprogrammierung, Verfügbarkeit auf vielen Plattformen
  noch immer eine der am meisten verwendeten Sprachen!
- Scheme – heute weit verbreiteter LISP-Dialekt
- PROLOG (Colmerauer, Marseille) – erste logikorientierte Programmiersprache
  Heute in KI und Wissensverarbeitung verbreitet

1.3 Geschichte der Programmiersprachen

80er Jahre

- Modula-2 (Wirth) – spezifische Konstrukte für modulares Programmieren
- Weiterentwicklungen bei funktionalen Sprachen:
  Scheme (Sussmann, Steele, MIT)
  Miranda (Turner)
  ML (Milner) – Typüberprüfung
- Ada (Pentagon) – Sprache für Militärzwecke
  State-of-the-art moderner Programmiersprachen, Packages, Tasks, Exceptions
  (Ada 95 – Erweiterung um objektorientierte Konzepte)
- Common Lisp – Standard für Lisp mit vielen Erweiterungen
- Durchbruch der objektorientierten Sprachen:
  Smalltalk (Kay, Ingalls, Xerox Parc)
  C++ (Stroustrup) – Erweiterung von C um Objektorientierung
- Eiffel (Meyer) – objektorientiert
- Modula-3 , Oberon (Wirth) – Pascal-Tradition

1.3 Geschichte der Programmiersprachen

90er Jahre

- Erkenntnis: Es gibt nicht DIE Programmiersprache
- Sprachen der 4. Generation: Anwendungsgeneratoren, fensterbasierte Programmieroberflächen, Spezialsprachen für Datenbanken (SQL)
- Wachsende Bedeutung von C++
- Netzorientiertes Programmieren, Code-Mobilität durch Java

Sprachgenerationen:

1. Assembler
2. unstrukturierte Hochsprachen
3. Prozedurale Sprachen (Pascal, C)
4. einfache Spezialsprachen
5. Deklarative Sprachen (Prolog, funktionale Sprachen)

1.3 Geschichte der Programmiersprachen

seit 2000

- Stärkere Nutzung von Java
- C# - Microsoftvariante eines objektorientierten C, Java-Ähnlichkeit
- aktuell : Neue Systeme vorwiegend in C++, C, Java, C# aber Pflege vieler Cobol und Fortran-Systeme