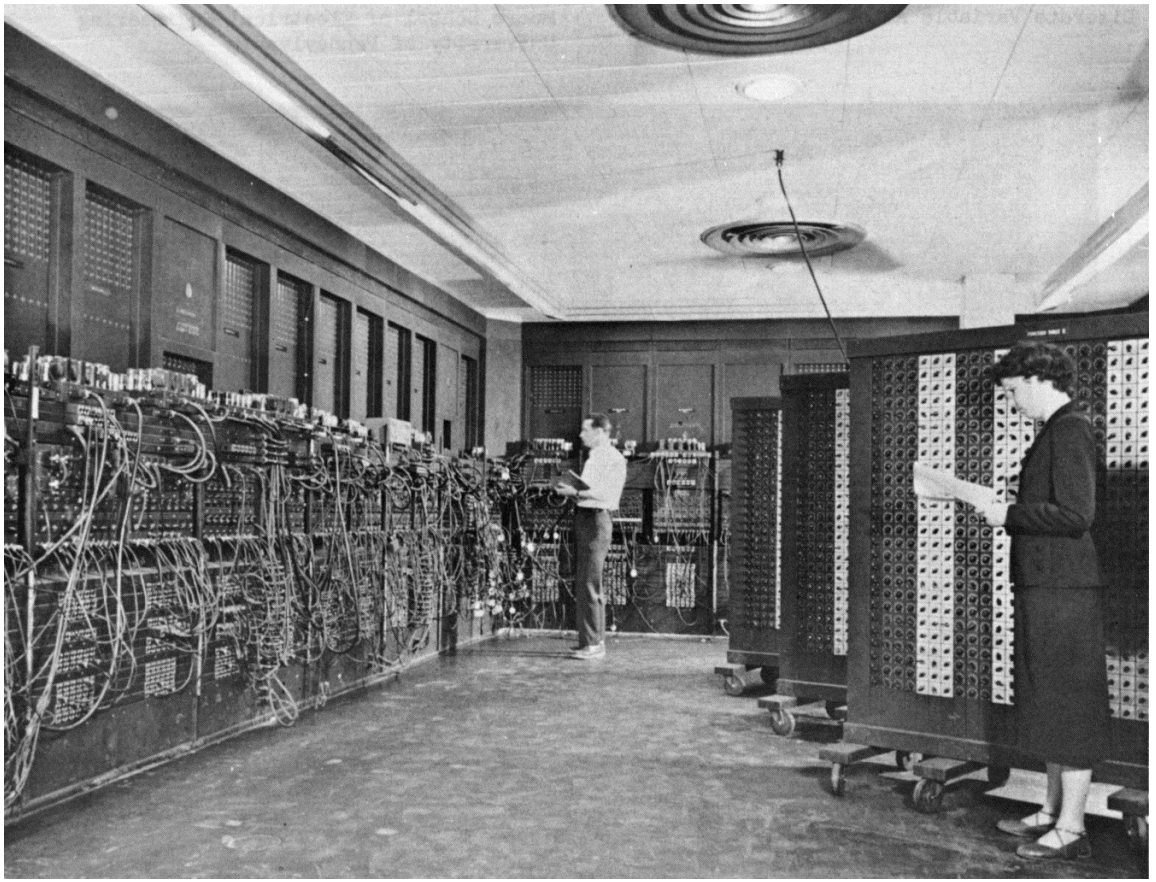
Geschichte und Ziele der

Objektorientierten Programmierung (OOP)

Geschichte der OOP

1945
US Army, **ENIAC**

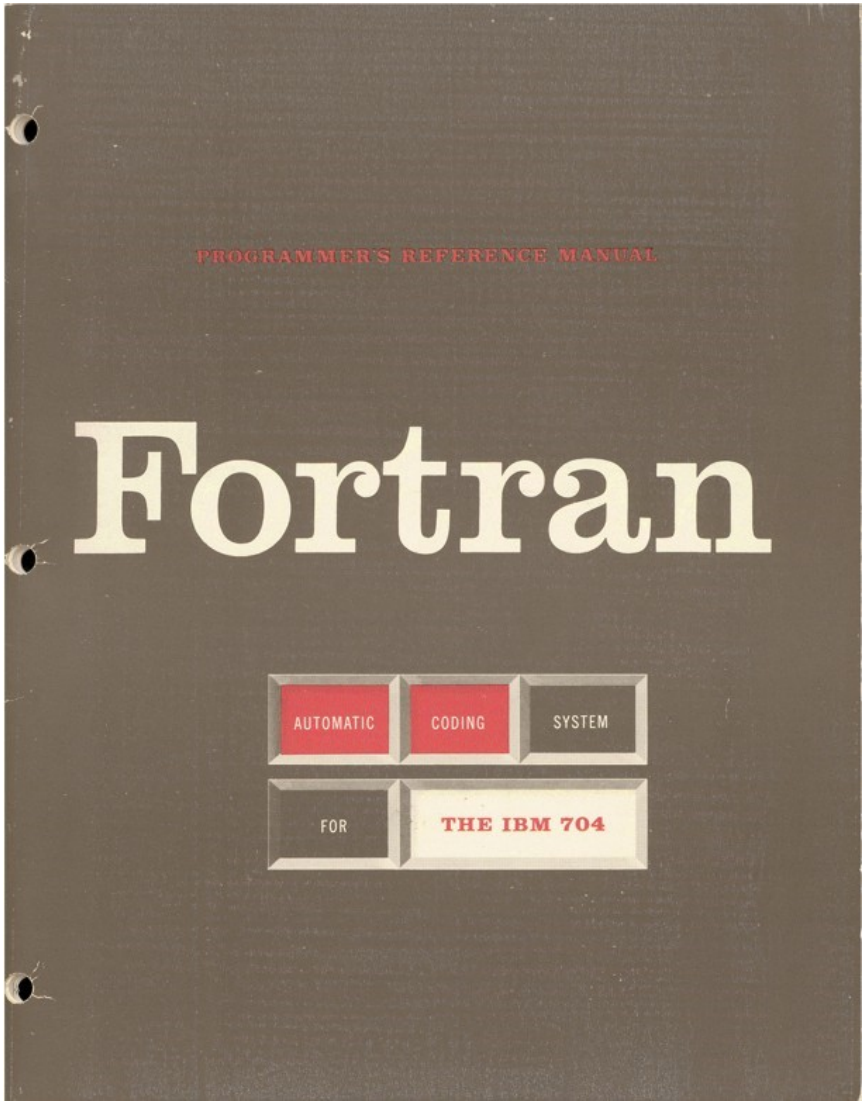
Erster elektronischer Computer



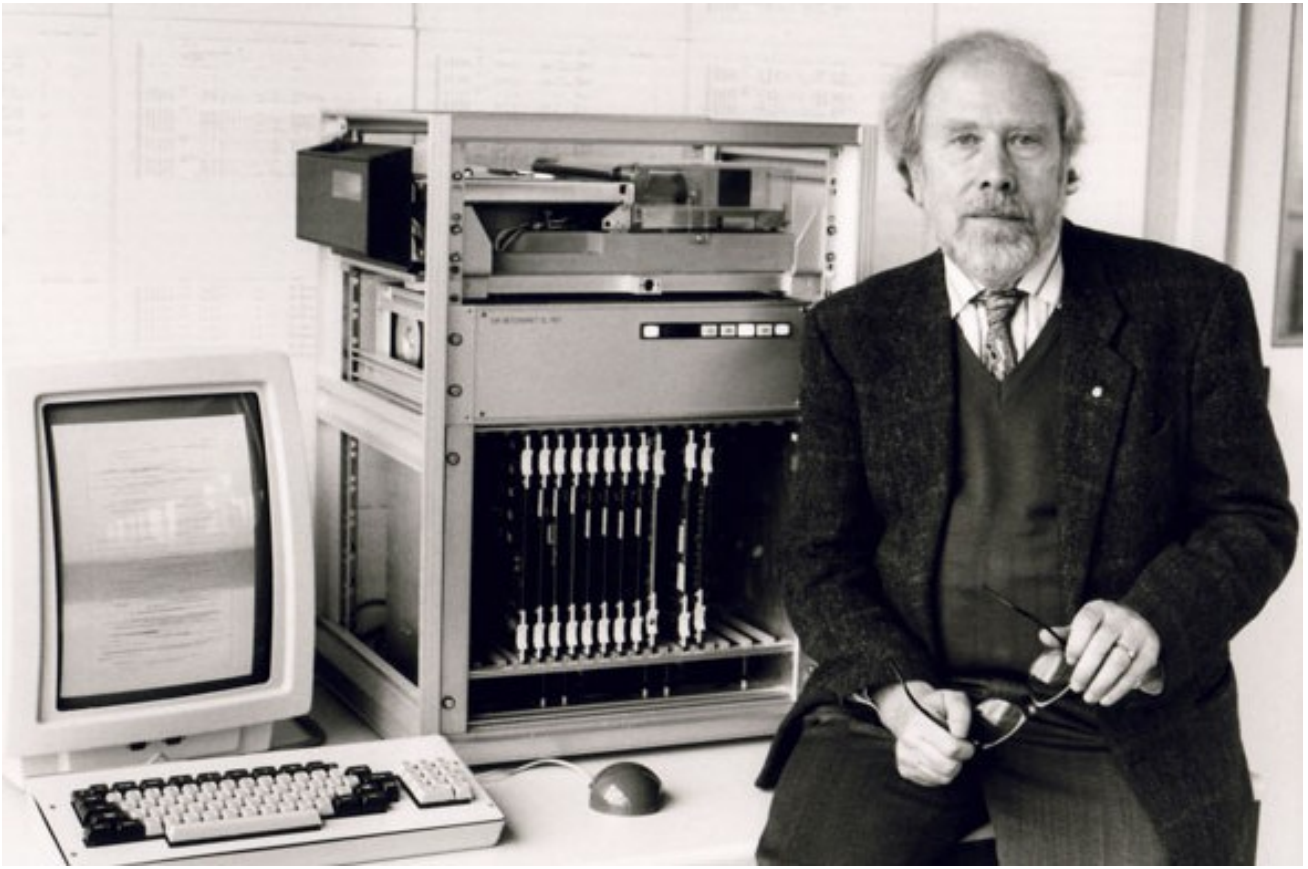
Geschichte der OOP

1967
IBM, **Fortran**

Erste imperative
Programmiersprache



Geschichte der OOP



1971 Niklaus Wirth, **Pascal**

Erste strukturierte Programmiersprache

Geschichte der OOP

1980
Xerox PARC
Smalltalk

Erste objektorientierte Programmiersprache

Geschichte der OOP

1985
Bjarne Soustrup
C++

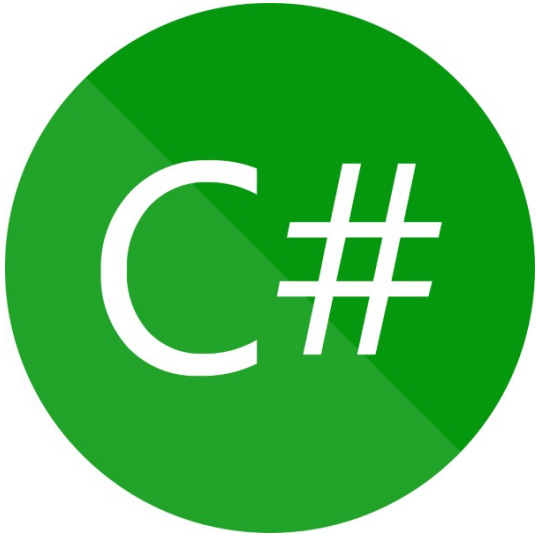
Weit verbreitete
OOP-Sprache

Geschichte der OOP

1995
SUN Microsystems
Java

Plattformunabhängige
OOP-Sprache

Geschichte der OOP



2000
Microsoft, **C# / .NET**

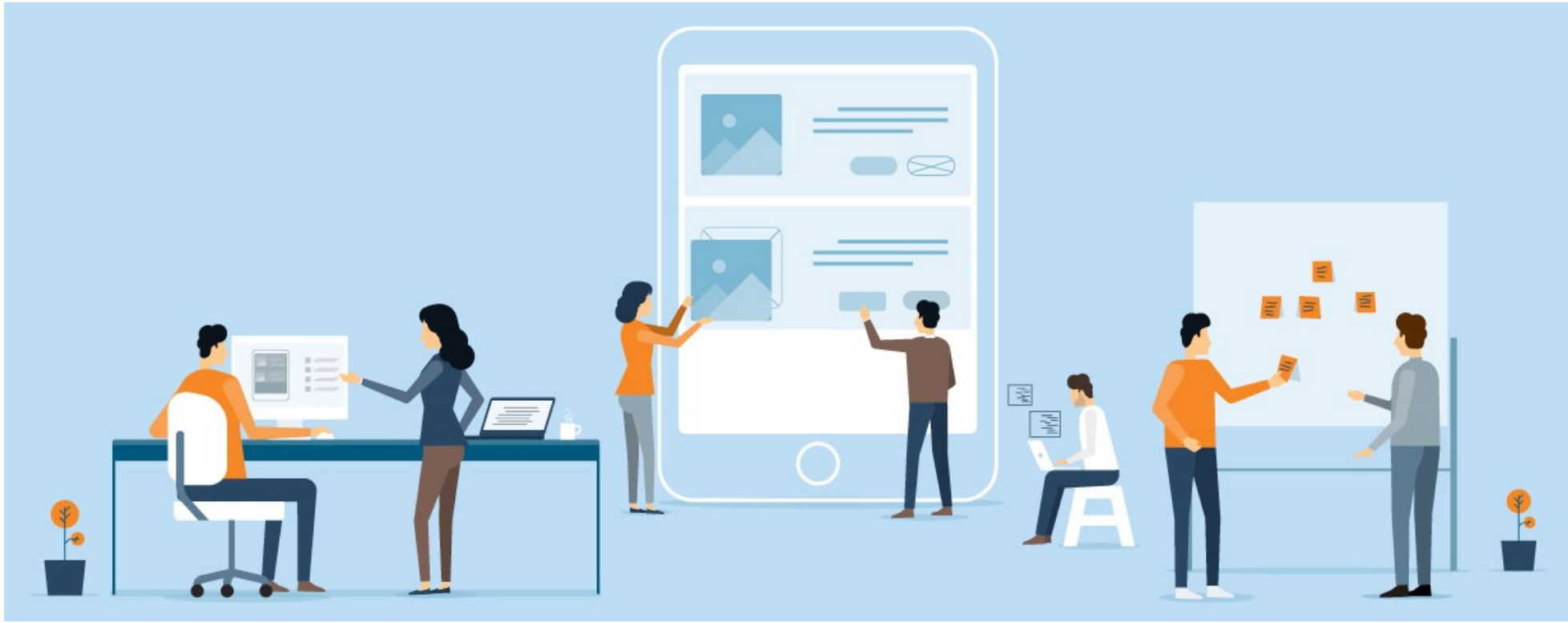
Weiterentwicklung von C++ und Java

Programmierung „früher“ (1960-1980)

Programmierung „früher“ (1960-1980)

- **Kleine Programme**
- **Wissenschaftliche** Anwendungen
  z.B. komplexe Gleichungen lösen
- **Wenige Entwickler** (1 – 2)
- Entwickler selbst Nutzer des Programms
  → Benutzerfreundlichkeit nicht notwendig
  → nur Texteingabe / -ausgabe
- **Imperative** Programmiersprachen

Programmierung „heute“



Programmierung „heute“

- **Komplexe Programme**
- **Viele Nutzer** (manchmal Millionen)
- Große **Entwicklerteams**
  unterschiedliche Rollen
- **Jahrelange Entwicklung**
  und Weiterentwicklung
- **Objektorientierte** Programmiersprachen

Rollen in Softwareentwicklung

- **Systemanalytiker** beschreibt Anforderungen („was soll das System können?“)
- **Softwarearchitekt** entwirft Struktur des Systems
- **Programmierer** entwickeln Teile des Systems
- **Softwaretester** prüfen Teile und Gesamtsystem
- **Grafiker** erstellen Grafiken für die Software
- **Projektmanager** planen und steuern Arbeitsablauf
- **Support-Team** hilft bei Installation und Problemen

Vorteile der OOP

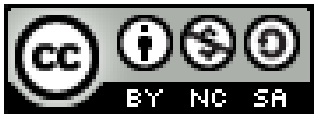
OOP-Sprachen unterstützen:

- **Modellierung** komplexer Anwendungsbereiche
  → Realität im Programm abbilden
- **Arbeitsteilung** für viele Programmierer
  → Schnelles Verständnis der Arbeit anderer
  → Fehler vermeiden
- **Änderungen** und Erweiterungen zulassen
  → Anpassung neue Anforderungen
- **Wiederverwendung** von Quellcode
  → Zeit und Geld zu sparen

Autor / Quellen

Autor:

- Christian Pothmann (cpothmann.de)
  Freigegeben unter CC BY-NC-SA 4.0, März 2021



Grafiken:

- ENIAC: U.S. Army, gemeinfrei
- Fortran: en.wikipedia.org, gemeinfrei
- N. Wirth: Copyright ETH Zürich
- Smalltalk: de.wikipedia.org C++ Logo: isocpp.org, gemeinfrei
- Java Logo: de.wikipedia.org C# Logo: Jason Groce, gemeinfrei
- .NET Logo: Microsoft, Lizenz CC0 1.0
- IBM-Programmierer: Bundesarchiv, Lizenz CC BY-SA 3.0
- Softwareentwicklung: technofaq.org, Lizenz CC BY-NC-SA 4.0