

В. В. Шилов



ВВЕДЕНИЕ В ПРОГРАММНУЮ ИНЖЕНЕРИЮ

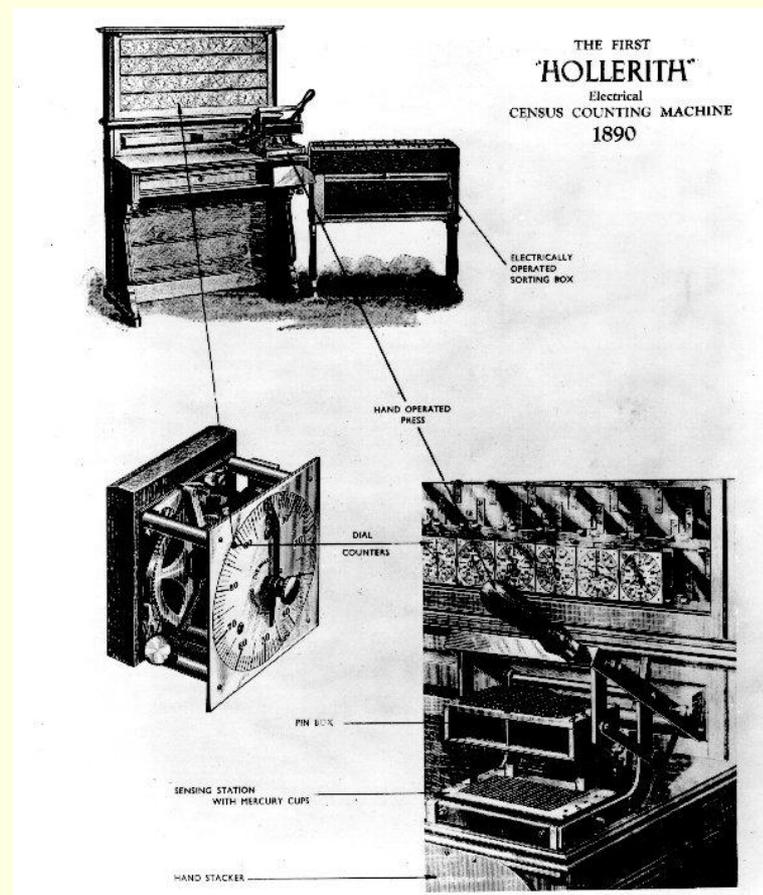
**От программирования
к программной инженерии**

Москва, 2018 год

Герман Холлерит

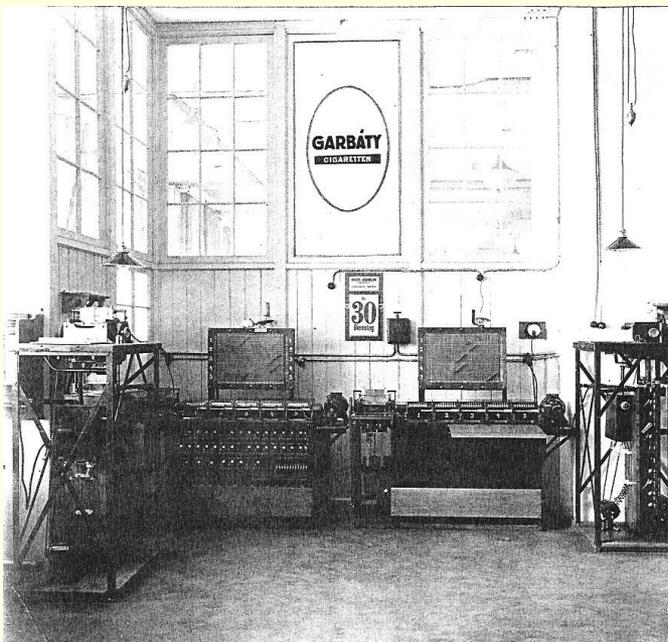


Герман Холлерит (Herman Hollerith, 1860-1929).



Счетно-табуляционные машины, предназначенные для обработки статистической информации.

Счетно-табуляционные машины



1911 г.

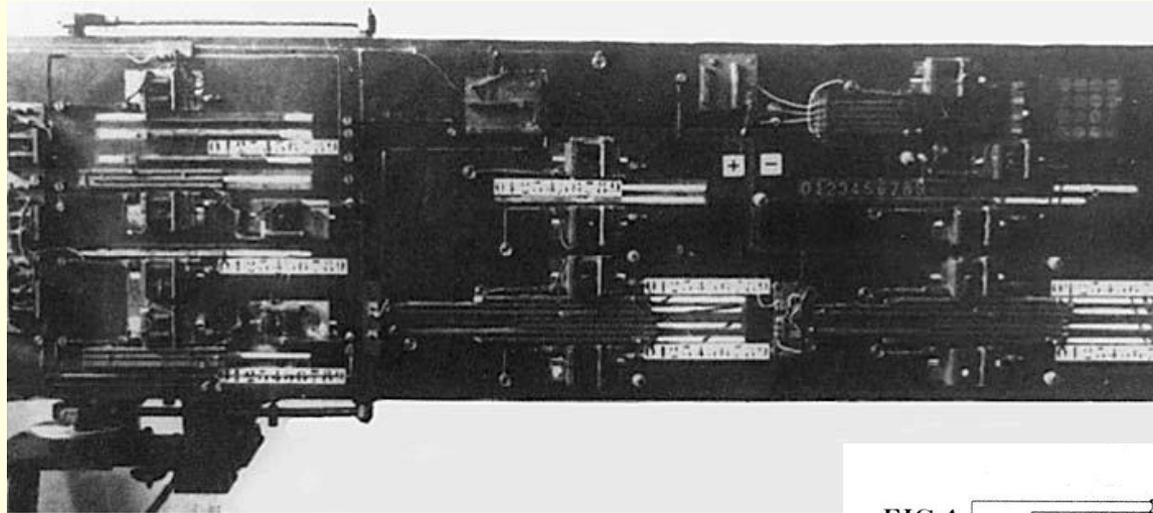
DEHOMAG с коммутационной доской, позволяющей перенастраивать машину на выполнение разных последовательностей действий.



1935 г. DEHOMAG D11.

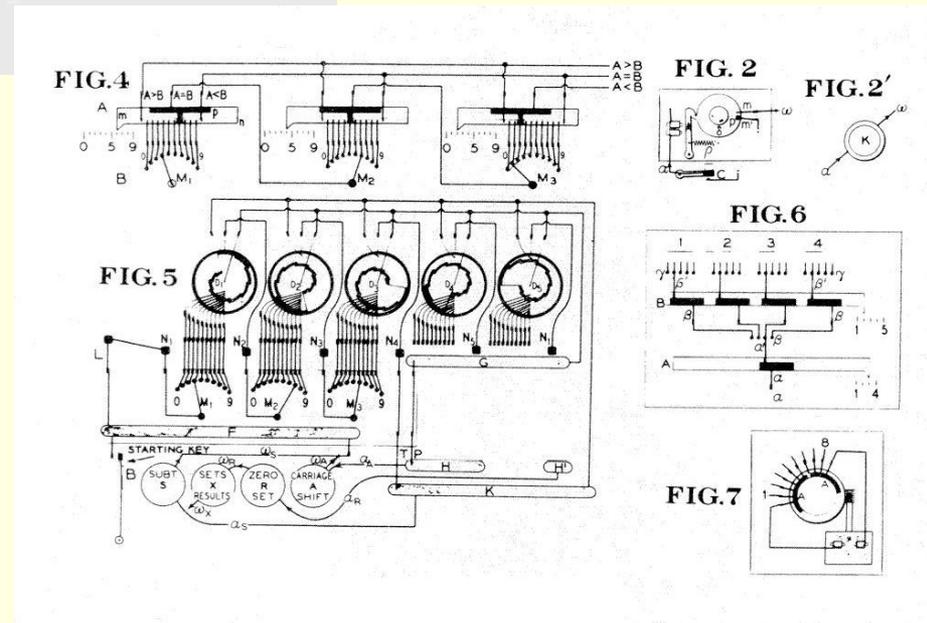
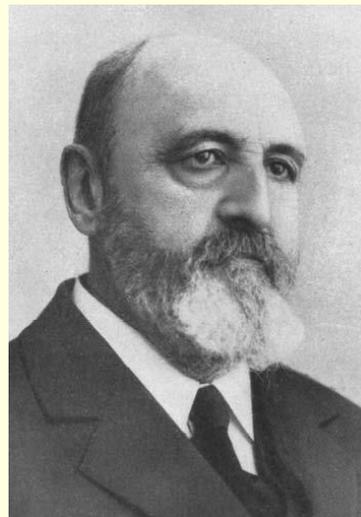
Возможность автоматически выполнять последовательность арифметических действий.

Специализированные машины



1914 г. Релейная машина для вычислений по формуле $p \times q - b$.

Леонардо Торрес-и-Кеведо
(Leonardo Torres-y-Quevedo,
1852-1936).



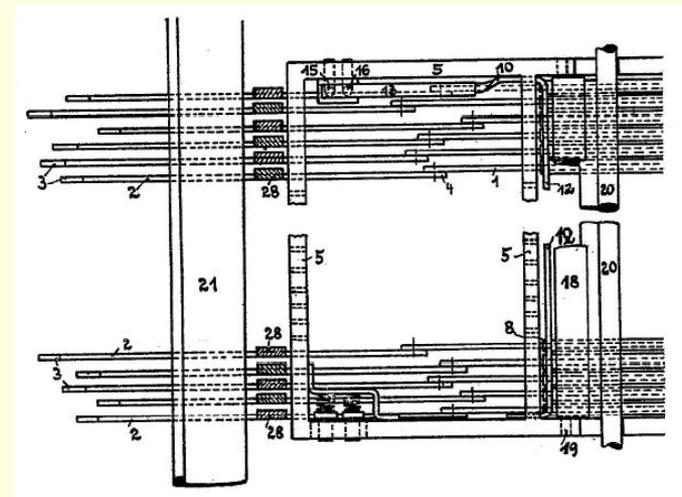
Появление памяти

Июнь 1923 г. Уроженец Испании Франсиско Кампос (Francisco Campos Campaña) получил в Германии патент (заявка была подана 2 марта 1920 г.) на механическую бухгалтерскую машину, предназначенную для раздельного ведения счетов нескольких клиентов.

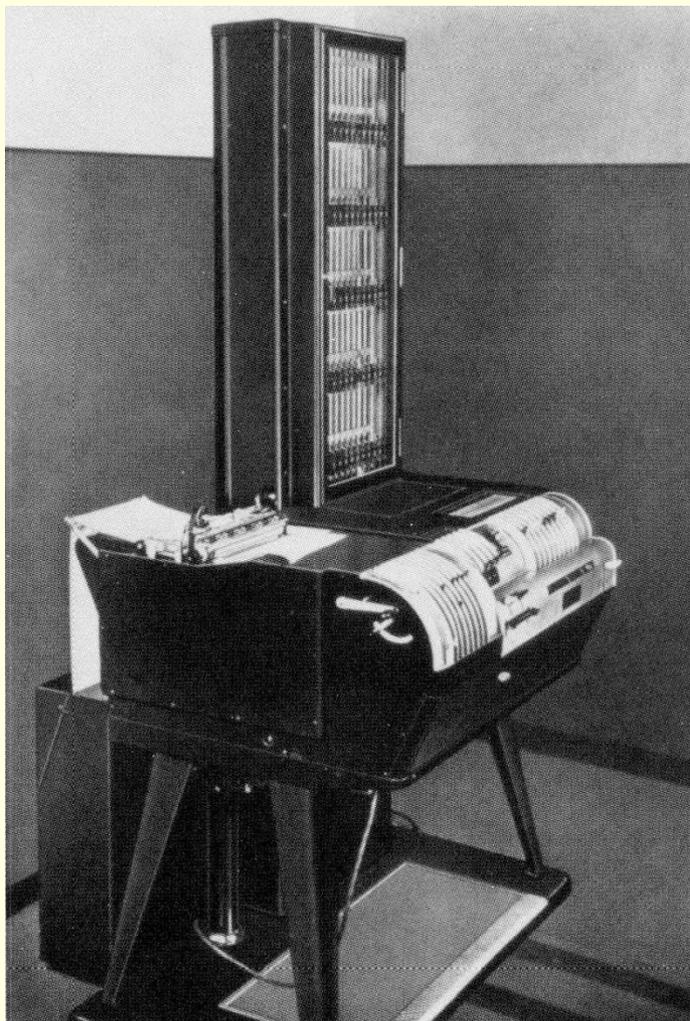
Машина могла складывать, вычитать, а также запоминать и хранить числа в памяти.

Она состояла из десяти пятизначных чисел, представленных наборами зубчатых реек (каждая рейка соответствовала одному разряду).

10 сентября 1924 г. Немецкий изобретатель Альфред Краузе (Alfred Krause) получил патент на механическую бухгалтерскую машину, содержащую устройство памяти для хранения результатов вычислений (судя по всему, в качестве прототипа своей разработки Краузе использовал патент Ф. Кампоса).



Появление памяти



1928 г. Ф. Кампос изготовил первый экземпляр своей механической бухгалтерской машины. По сравнению с описанным в патенте прототипом, Кампосу удалось добиться серьезного прогресса – его машина могла хранить и обрабатывать 1000 девятиразрядных десятичных чисел. Память была **адресуемой** (набрав номер счета, можно было распечатать его состояние). Машина Кампоса не имела ни предшественников, ни аналогов – это уникальное устройство стало первой в истории механической вычислительной машиной, имеющей память такого большого объема.

Появление памяти



Бухгалтерские машины модели "National 3000" компании NCR имели несколько регистров для запоминания промежуточных результатов и обладали возможностью передачи данных из регистра в регистр.

С их помощью можно было фактически имитировать работу разностной машины Бэббиджа.

Начало 1920-х гг.

Конрад Цузе



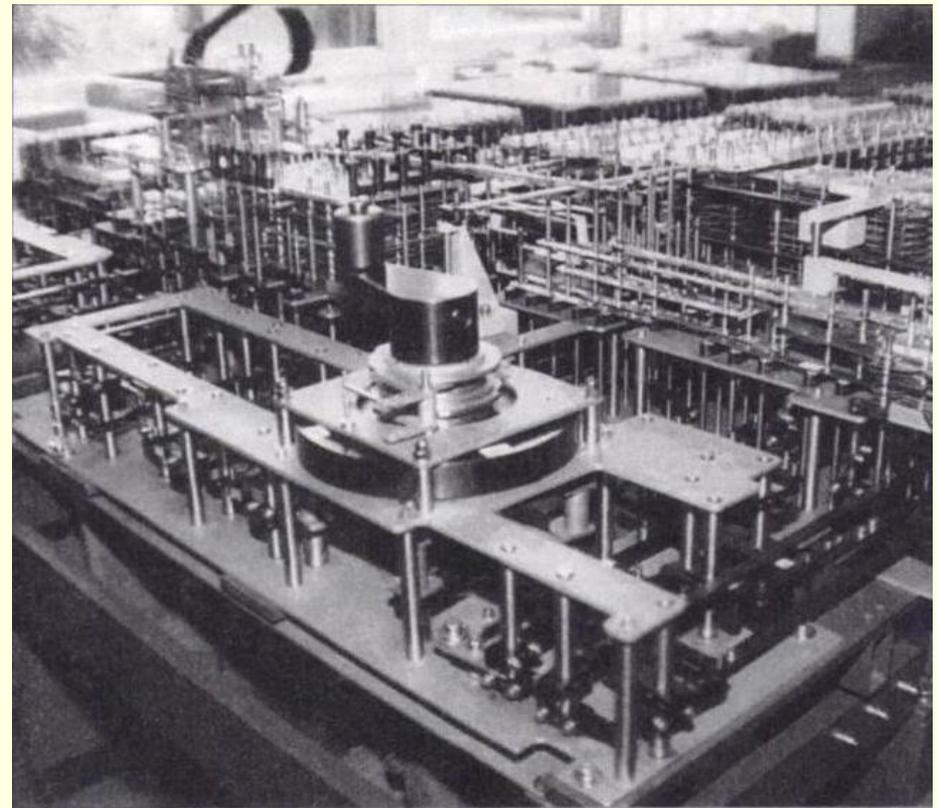
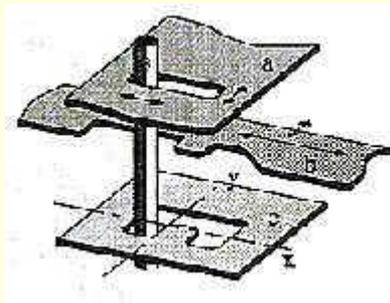
Конрад Цузе (Konrad Zuse, 1910-1995).

Компьютеры Конрада Цузе

1938 г. Механический управляемый программой компьютер Z-1.

Каждая ячейка памяти состояла из трех металлических пластинок, средняя из которых могла двигаться, смещая при этом штырь влево или вправо и тем самым определяя значение ячейки – 0 или 1.

Главным достоинством такой памяти Цузе считал ее компактность и расширяемость.



Первый электронный компьютер ENIAC

Джон фон Нейман
(John von Neumann, 1903-1957).

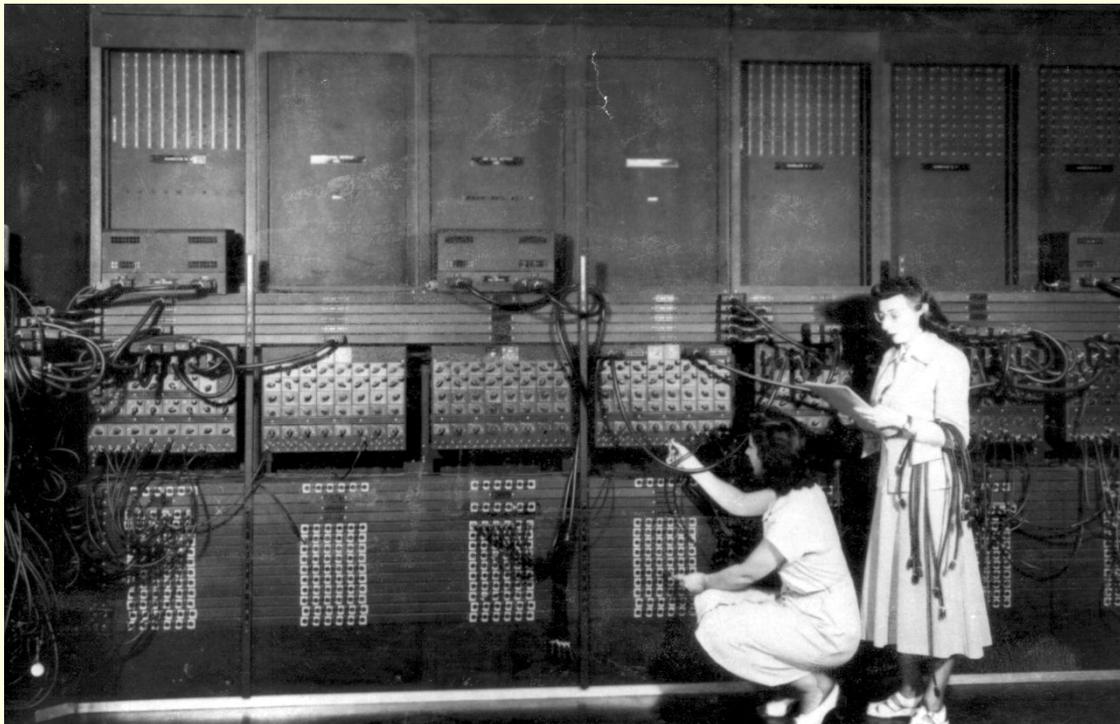


Джон Моучли (John William Mauchly, 1907-1980)

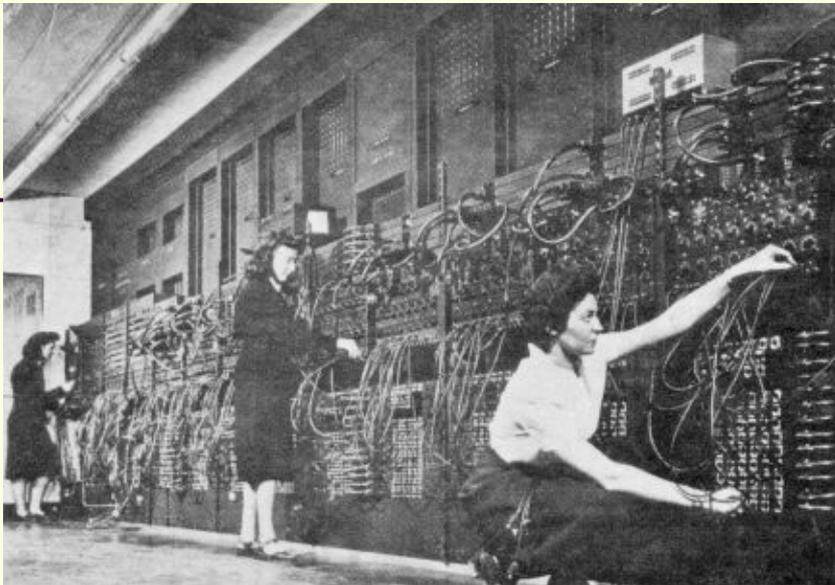
Джон Преспер Эккерт-мл. (John Presper Eckert, Jr, 1919-1995)

“Программирование” на ENIAC

Август 1942 г. Джон Моучли впервые использовал термин **to program** (**программировать**), но еще не в современном смысле, а для обозначения процесса соединения исполнительных устройств проектировавшейся им вычислительной машины с помощью штекерного коммутатора.

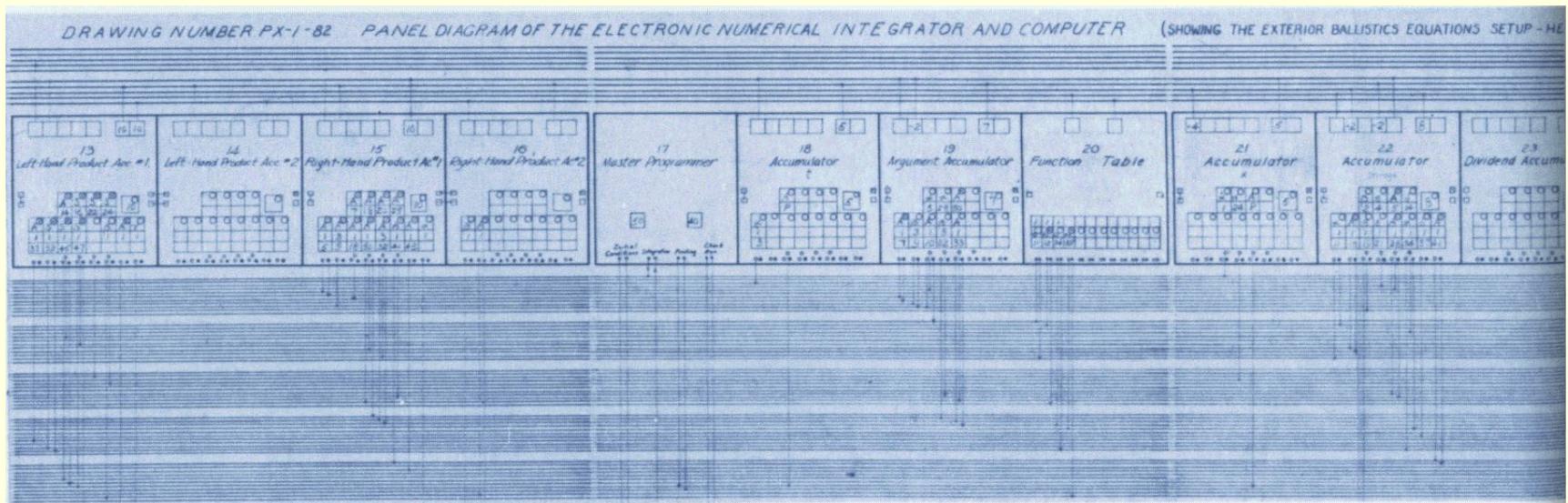


1945 г.



Так работали первые
“программисты”...

А программа записывалась так...



Принципы фон Неймана

Июнь 1945 г. Джон фон Нейман присоединился к группе разработчиков ENIAC и принял активное участие в обсуждении проекта их будущего компьютера EDVAC (Electronic Discrete VArIable Computer).

Отчет "First Draft of Report on the EDVAC":

1. **Программное управление работой компьютера.**
2. Принцип условного перехода.
3. **Принцип хранимой программы.**
4. Использование двоичной системы счисления.
5. Принцип иерархичности памяти.

Фактически эти принципы выработали Д. Моучли и П. Эккерт. Некоторые считают, что еще раньше их сформулировал Конрад Цузе.

Термин “программа”

Июль 1946 г. В своей лекции на конференции в Пенсильванском университете Преспер Эккерт охарактеризовал принципиально новую концепцию компьютерной архитектуры, основанную на хранении программы в памяти, и впервые использовал слово **программа** в современном значении.

Джон фон Нейман этот термин использовал крайне редко. В своей работе 1945 г., давая описание особенностей новой архитектуры, он говорит об **инструкциях, стандартных командах** (*standard orders*), **операциях и коде** (*code*) – но ни разу о программе! Год спустя в одной из статей словом **программа** он пользуется наряду со словом **код**, а еще через год – снова отказывается от него в пользу терминов **план** (*plan*) и **установка** (*set-up*).

Декабрь 1951 г. В отчете по работе над ЭВМ М-1, построенной под руководством И. С. Брука, четко говорится, что “Набор инструкций, необходимых для решения задачи, называется **программой**”.

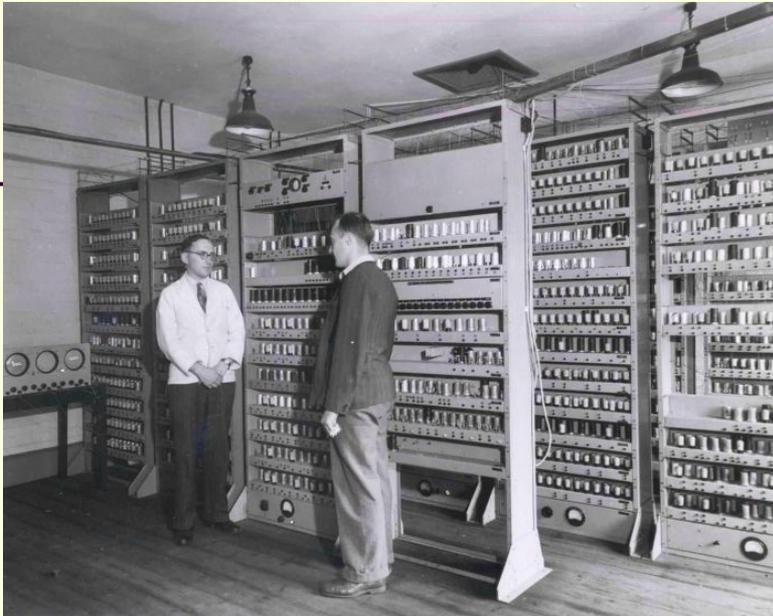
Что же такое программирование?

Программа. Последовательность операций, выполняемых компьютером для достижения поставленной цели или задачи. Описание на **языке программирования** действий, которые должен выполнить компьютер в соответствии с **алгоритмом** решения конкретной задачи. Упорядоченная последовательность **команд**, подлежащих обработке.

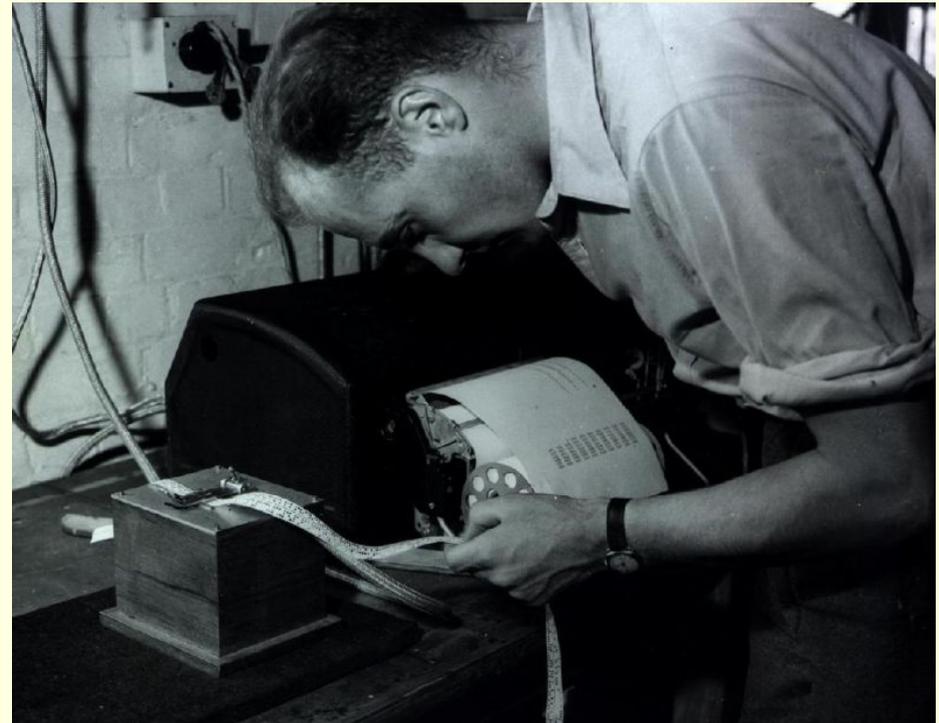
Программирование – совокупность процессов, связанных с разработкой программ и их реализацией.

В широком смысле к этим процессам относят все технические операции, необходимые для создания программ, в том числе анализ требований, все стадии разработки, а также реализации в виде готового продукта.

В узком смысле программирование – процессы выбора структуры, кодирования и тестирования программ.



**EDSAC, 1949 г. –
первый компьютер
с хранимой программой**



Программирование для EDSAC

```

31 T 158 S required by initial orders
enter → 32 E 61 S jump-over constants
33 P 8 S x1
34 P S x0
35 P 16 S x2
36 P S
37 P S
38 P S
39 P S
40 P S
41 P S
42 P S
43 P S
44 P 80 S 10·x1
X 45 P 1398 S 6x2/12; scaled by 211·10-1
46 P S
47 P S
48 P S
49 P S
X 50 P S
51 P 1 S
52 P 8 S 6x; scaled by 211·10
X 53 P S column count
54 P 5 S 5; initializer for column count
55 P 174 S
56 P 30 S
X 57 P 11633 S
58 P 228 S
59 P 185 S
60 P 12057 S

```

all scaled by 2⁻¹¹·10
constant needs to be double length, i.e.,

519845
P13988

interchange most-significant and least-significant half-words

Июль 1949 г.

Фрагмент программы вычисления интеграла Айри, – решения уравнения Айри: $y'' = xy$.

157 команд, около 20 ошибок!

Перечень ошибок в программе

Команда	Ошибка
62, 64, 68, 70	Incorrect left-shift order
72, 82	Incorrectly specified operand length
45, 50L, 53, 57	Incorrect constants
73, 144	Redundant instructions or constants
102-3, 143-4, 148-9	Missing orders
130, 141	Incorrect address in instruction
135, 149, 152	Incorrect or misspunched opcode

Отладка

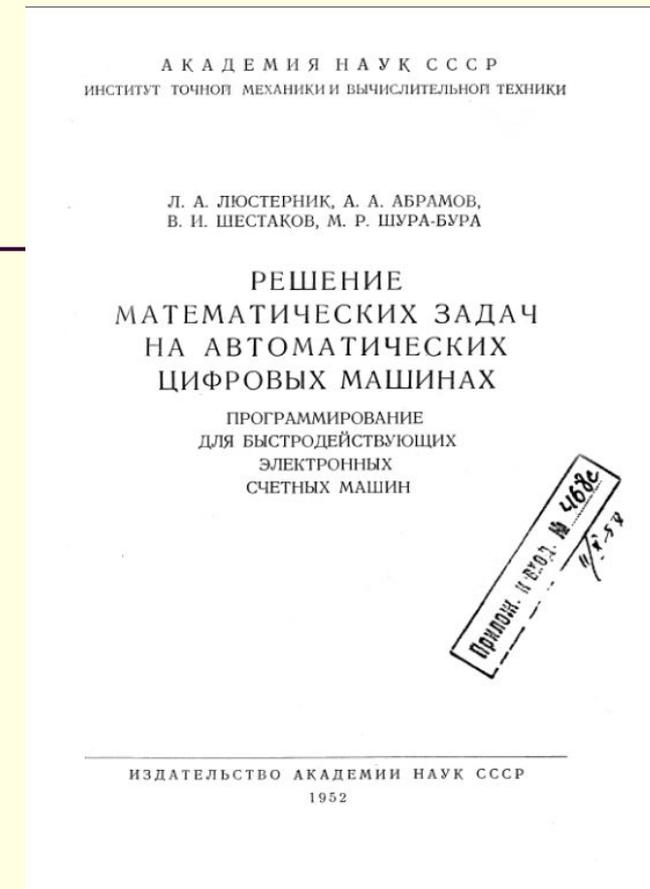
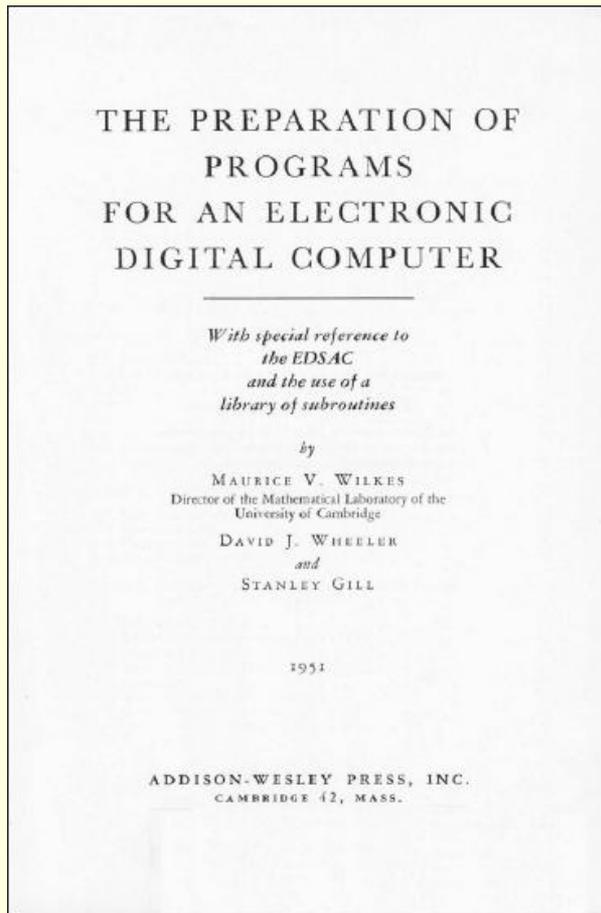
Морис Уилкс вспоминал позднее, какой ужас охватил его при мысли, что “большую часть оставшейся жизни придется провести в поисках ошибок в собственных программах”.

Именно тогда разработчики компьютеров впервые осознали необходимость создания эффективных методов обнаружения ошибок (**тестирования**) и **отладки программ**, а также более эффективных методов написания самих программ.



Дэвид Уилер (1927-2004)

Первые монографии



“The Preparation of programs for an electronic digital computer: with special reference to the EDSAC and the use of a library of subroutines”.

Maurice Wilkes, David Wheeler, Stanley Gill (1951)

Plankalkül Конрада Цузе – первый язык программирования

$$\begin{array}{l|l} V & Z + 1 \Rightarrow Z \\ & 1 \quad 1 \\ S & 1 \cdot n \quad 1 \cdot n \quad 1 \cdot n \end{array} \quad \text{– присваивание.}$$

$$\begin{array}{l|l} V & (V, V) \Rightarrow R \\ & 0 \quad 1 \quad 0 \\ S & \sigma \quad \sigma \quad 2\sigma \end{array} \quad \text{– объединение двух значений.}$$

$$\begin{array}{l|l} V & W \left[\begin{array}{l} \mu x(x \in V \wedge x \neq V) \Rightarrow Z \\ \quad \quad \quad 0 \quad 1 \quad 0 \end{array} \right. \left. \begin{array}{l} (R \wedge R17(Z)) \Rightarrow R \\ \quad \quad \quad 0 \quad 1 \quad 0 \quad 0 \end{array} \right] \\ S & \left[\begin{array}{l} \sigma \quad m\sigma \quad \sigma \quad \sigma \\ \quad \quad \quad 0 \quad 0 \quad \quad 0 \end{array} \right] \end{array} \quad \text{– оператор повторения.}$$

$$\begin{array}{l|l} P17 & R(V) \Rightarrow (R, R) \\ & V \quad 0 \quad 0 \quad 1 \\ & S \quad \quad \quad 0 \end{array} \quad \text{– заголовок подпрограммы.}$$

Язык был разработан Конрадом Цузе в 1945 г. Описание частично опубликовано в 1959 и 1972 годах.

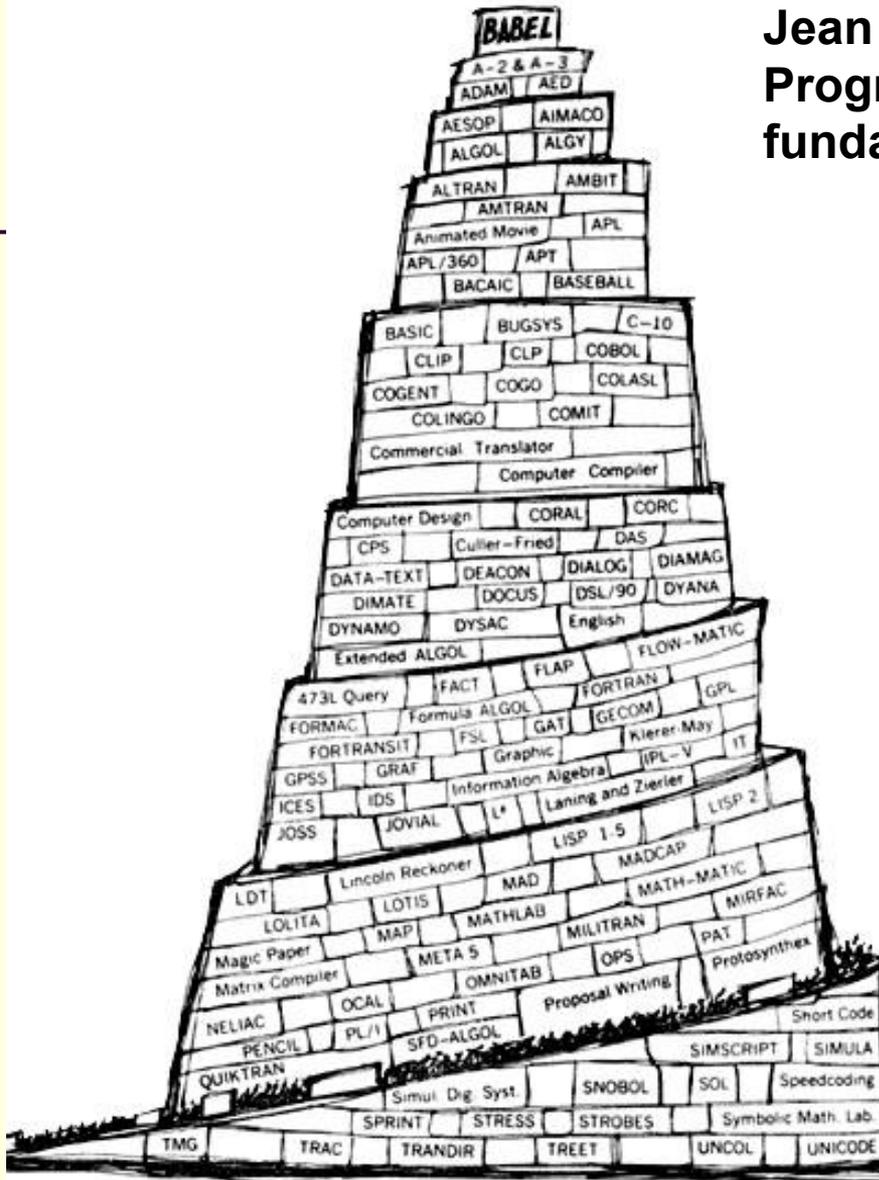
Plankalkül Конрада Цузе – первый язык программирования

		① $R(V) \Rightarrow R$					
V		0	0				
S		$m\sigma$	0				
		② $Az(V) \Rightarrow \& R$		$V \Rightarrow Z$	③ $0 \Rightarrow \varepsilon$		
V		0	0	0	0		
K		0		0			
S		σ	0	σ	σ		
						$1n$	
V	W	④ $\mu x \left[\begin{array}{cc} x \in V & \& x \neq V \\ 0 & 0 \end{array} \right] \Rightarrow Z$		⑤ $Sq(Z, Z) \Rightarrow \& R$			
K			0	1	0	1	0
S		$\left[\begin{array}{cc} \sigma & m\sigma \\ \sigma & \sigma \end{array} \right]$	σ		σ	σ	0
		⑥ $Kla(Z) \rightarrow (\varepsilon + 1 \Rightarrow \varepsilon)$		⑦ $Klz(Z) \rightarrow (\varepsilon - 1 \Rightarrow \varepsilon)$			
V		1		1			
S		σ		σ			
		⑧ $\varepsilon \geq 0 \Rightarrow \& R$		⑨ $Z \Rightarrow Z$			
V			0	1	0		
S			0	σ	σ		
		⑩ $Sz(Z) \Rightarrow \& R$		$\varepsilon = 0 \Rightarrow \& R$			
V		0	0		0		
S		σ	0		0		

Программа проверки синтаксической
правильности записи булевского
выражения.

2000 г. Первая реализация языка.

Jean Sammet.
Programming languages: history and
fundamentals. 1969



Вавилонская башня языков программирования



**Anthony G. Oettinger,
профессор Гарварда**

**The hardware-software complementarity //
Communications of the ACM. 1967. Vol. 10.
No 10. Pp. 604–606.**

The notion of software engineering is, thank goodness, beginning to be heard of more and more. It seems strange to classical engineers and to classical mathematicians alike, because, you know, why would a mathematician think of engineering with symbols and, by the same token, why would somebody who thinks of engineering in terms of things we do with pieces of metal or transistors or what have you, think of an operation that takes place on paper with pencils and erasers as engineering.

I want to give you an illustration of the engineering dimensions of this kind of a job. Writing down instructions to a computer is getting easier and easier as



Термин «**программная инженерия**» впервые широко зазвучал в 1968 году в Гармиш-Партенкирхене на конференции, организованной Научным комитетом НАТО. Его введение должно было стимулировать поиск решений для имевшего место в то время «**кризиса программного обеспечения**».

Результатом этого поиска со временем стало появление соответствующей области исследований и **профессии программного инженера**.

SOFTWARE ENGINEERING

Report on a conference sponsored by the
NATO SCIENCE COMMITTEE
Garmisch, Germany, 7th to 11th October 1968

Chairman: Professor Dr. F. L. Bauer
Co-chairmen: Professor L. Bolliet, Dr. H. J. Helms

Editors: Peter Naur and Brian Randell

January 1969

Труды конференции 1968 г.



Фридрих Л. Бауэр (1924-2015)

«Кризис программного обеспечения» — последствия быстрого роста вычислительной мощности компьютеров и сложности решаемых с их помощью задач.

Фактически это сложность написания работоспособного, понятного программного обеспечения с использованием верифицированных алгоритмов.

ЦЕНА ПРОГРАММНОЙ ОШИБКИ

1964 г.

Программа на языке Фортран

Вместо

DO 10 I=1,5

...

10 ...

Было написано

DO 10 I=1.5

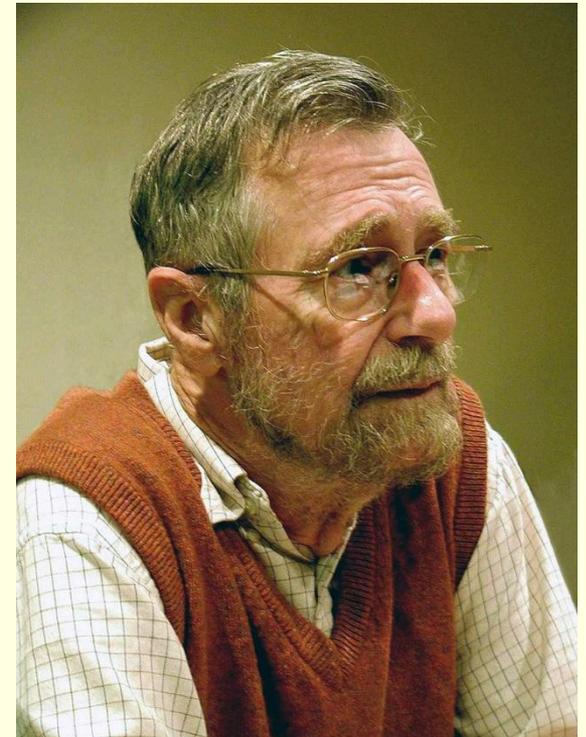
Компилятор воспринял как

DO10I=1.5

О пользе типизации данных...

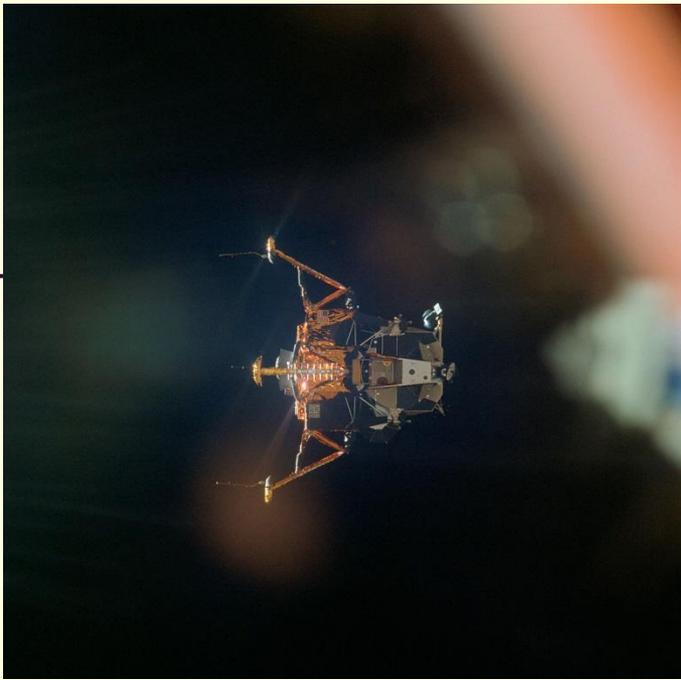
**«Отладка – процесс устранения ошибок в программе.
Программирование – процесс внесения ошибок».**

Эдсгер Дийкстра (1930-2002)



**«Craftsman or Scientist?»
(«Ремесленник или ученый?»), 1975
Конференция «ACM Pacific 75»**

16-24 июля 1969 г.



Margaret Hamilton, the Engineer Who Took the Apollo 11 to the Moon



Что такое инженерия?

Инженерия (инженерное дело, инжиниринг; англ. *engineering* ← от лат. *ingenium* — «искусность» и *ingeniare* — «изловчиться, разработать») — область технической деятельности, включающая ряд специализированных областей и дисциплин, направленная на практическое приложение и применение научных, экономических, социальных и практических знаний с целью обращения природных ресурсов на пользу человека.

Цели — изобретение, разработка, создание, внедрение, ремонт, обслуживание и/или улучшение техники, материалов или процессов.

Инженерия опирается на положения фундаментальной науки и результаты прикладных исследований.

Инженерия в программировании

Программная инженерия (*software engineering*) — приложение систематического, дисциплинированного, измеримого подхода к развитию, функционированию и сопровождению программного обеспечения, а также исследованию этих подходов; приложение дисциплины инженерия к программному обеспечению.

Программная инженерия интегрирует принципы математики, информатики и компьютерных наук с инженерными подходами, разработанными для производства материальных объектов.

Системный подход к анализу, проектированию, оценке, реализации, тестированию, обслуживанию и модернизации программного обеспечения, то есть применение инженерии к разработке программного обеспечения.

Curriculum 68

Computer Science

BI intro to computing
B2 computers and programming
B3 discrete structures
B4 numerical calculus

I1 data structures
I2 prog. languages
I3 comp. organization
I4 systems programming
I5 switching theory*
I6 num. analysis **I***
I7 num. analysis **II***

A1 formal languages
A2 advanced computer organization
A3 analog and hybrid computing
A4 systems simulation
A5 information retrieval
A6 computer graphics
A7 theory of computability
A8 large-scale information systems
A9 **AI**, heuristic programming

Curriculum 78

Computer Science

CS1 programming **I**
CS2 programming **II**
CS3 comp. systems
CS4 comp. organization
CS5 file processing

I1 data structures
I2 programming languages
I3 comp. organization
I4 systems programming
I5 switching theory
I6 num. analysis **I**
I7 num. analysis **II**

A1 formal languages
A2 advanced computer organization
A3 analog and hybrid computing
A4 systems simulation
A5 information retrieval
A6 computer graphics
A7 theory of computability
A8 large-scale information systems
A9 **AI**, heuristic programming

Computing Curricula 91

- Дискретные структуры (DS)
- Основы программирования (PF)
- Алгоритмы и теория сложности (AL)
- Архитектура и организация ЭВМ (AR)
- Операционные системы (OS)
- Распределенные вычисления (NC)
- Языки программирования (PL)
- Взаимодействие человека и машины (HC)
- Графика и визуализация (GV)
- Интеллектуальные системы (IS)
- Управление информацией (IM)
- Социальные и профессиональные вопросы программирования (SP)
- **Программная инженерия (SE)**

Последующие стандарты

Computing Curricula 2001

Computing Curricula 2005

Computer Science Curriculum 2008

Computer Science Curricula 2013

Программная инженерия – неотъемлемая часть!



Software Engineering 2004

Curriculum Guidelines for Undergraduate
Degree Programs in Software Engineering

A Volume of the Computing Curricula Series

August 23, 2004

The Joint Task Force on Computing Curricula
IEEE Computer Society
Association for Computing Machinery

This material is based upon work supported by the
National Science Foundation under Grant No. 0003263



Software Engineering 2014

Curriculum Guidelines for Undergraduate
Degree Programs in Software Engineering

A Volume of the Computing Curricula
Series

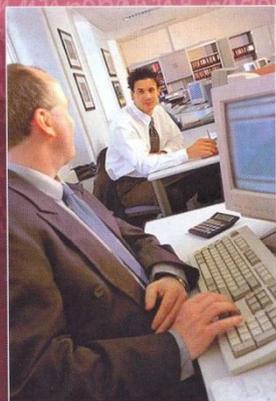
23 February 2015

Joint Task Force on Computing Curricula
IEEE Computer Society
Association for Computing Machinery

Software Engineering 2014

ЭНЦИКЛОПЕДИЯ ПРОФЕССИЙ

КАРЬЕРА в информационных технологиях



Аванта

Программная инженерия в списке
возможных занятий ИТ-специалиста
еще отсутствовала!

Карьера в информационных технологиях. М., 2004



**СПАСИБО
ЗА ВНИМАНИЕ!**