Programování pro neprogramátory:
Nedosažitelný sen nebo blízká realita?

Tomáš Petříček, MFF UK, Praha

tomas@tomasp.net
@tomasp.net
https://tomasp.net

Charles University

Kdo jsem & co dělám

Programovací jazyky a nástroje ze všech možných pohledů!

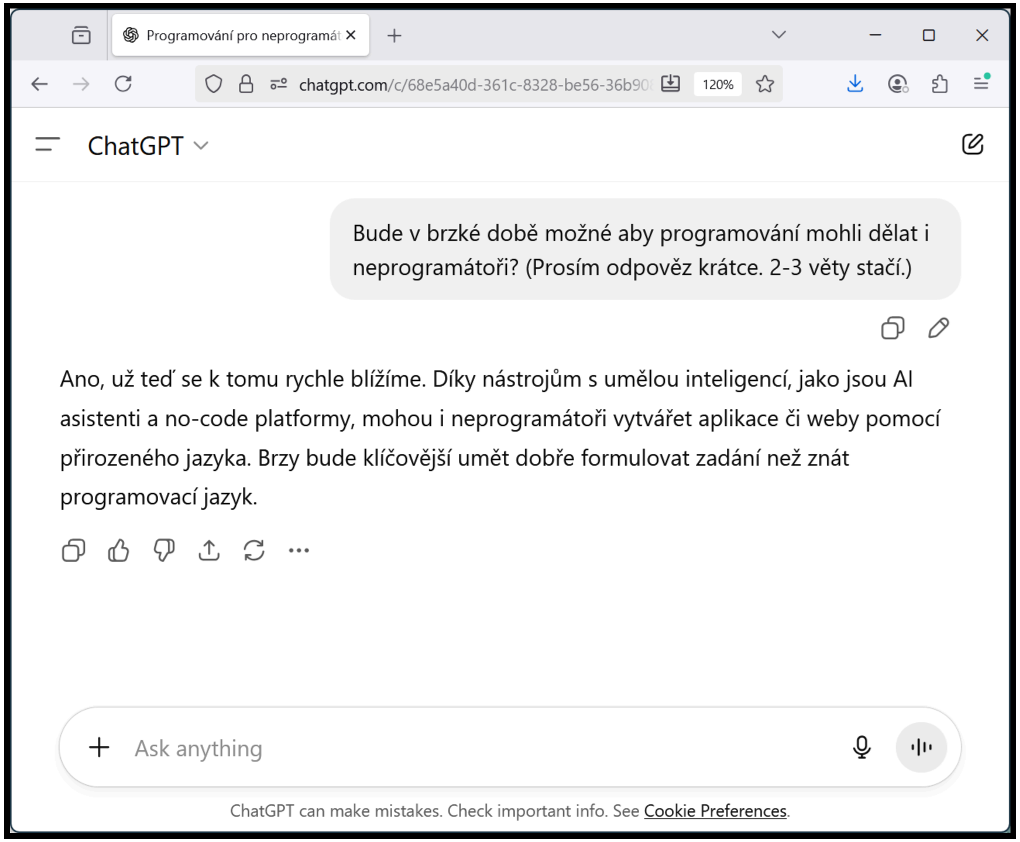
Absolvent MFF UK, doktorát z Cambridge, Microsoft Research & University of Kent

Přednáška bude česky, more or less...

Charles University

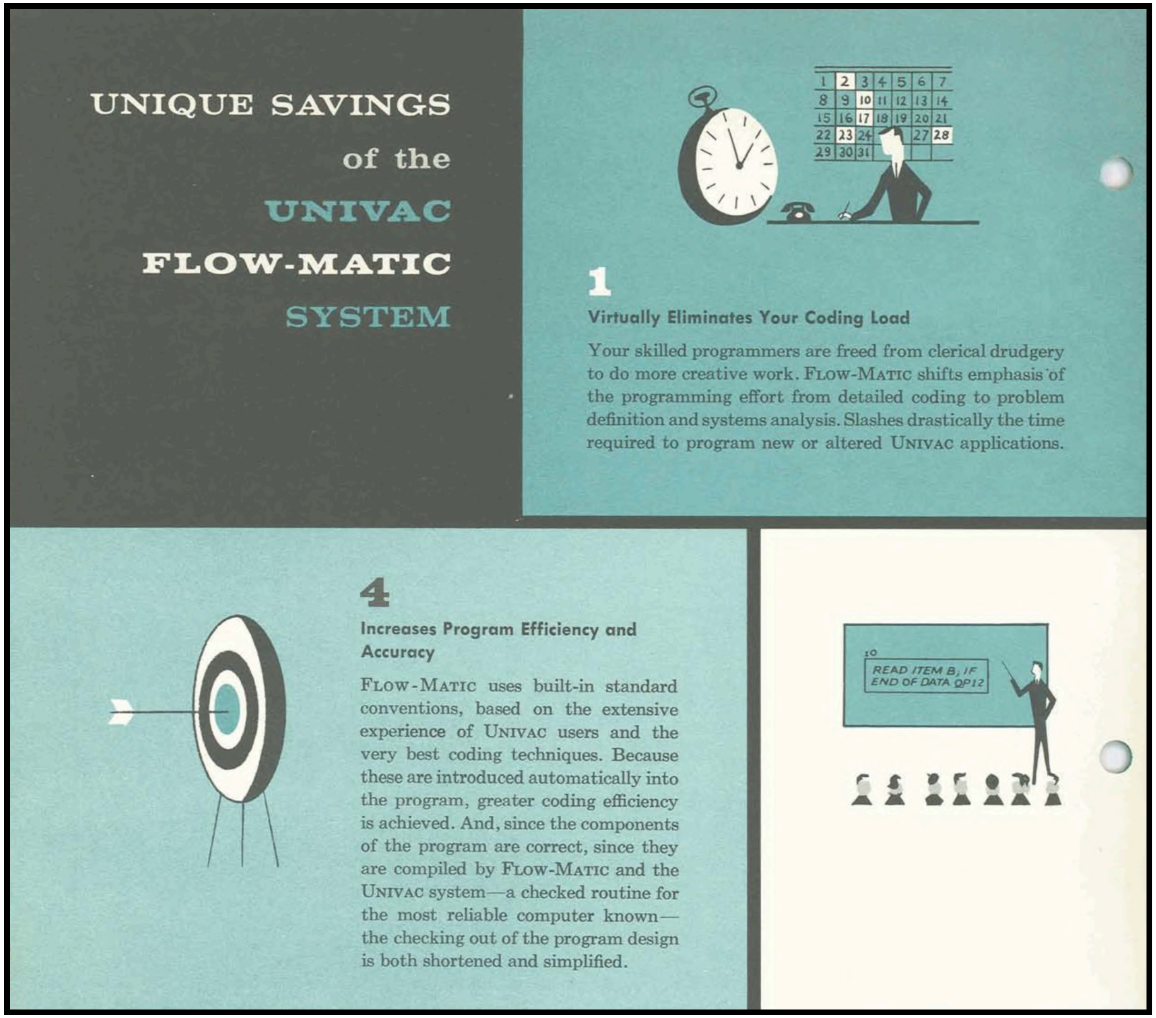
Programování pro neprogramátory

Bude v brzké době možné aby programování dělali i neprogramátoři?



Charles University

FLOW-MATIC


UNIQUE SAVINGS of the UNIVAC FLOW-MATIC SYSTEM

1

Virtually Eliminates Your Coding Load

Your skilled programmers are freed from clerical drudgery to do more creative work. FLOW-MATIC shifts emphasis of the programming effort from detailed coding to problem definition and systems analysis. Slashes drastically the time required to program new or altered UNIVAC applications.

4

Increases Program Efficiency and Accuracy

FLOW-MATIC uses built-in standard conventions, based on the extensive experience of UNIVAC users and the very best coding techniques. Because these are introduced automatically into the program, greater coding efficiency is achieved. And, since the components of the program are correct, since they are compiled by FLOW-MATIC and the UNIVAC system—a checked routine for the most reliable computer known—the checking out of the program design is both shortened and simplified.

Uhodnete z jakého roku je obrázek?

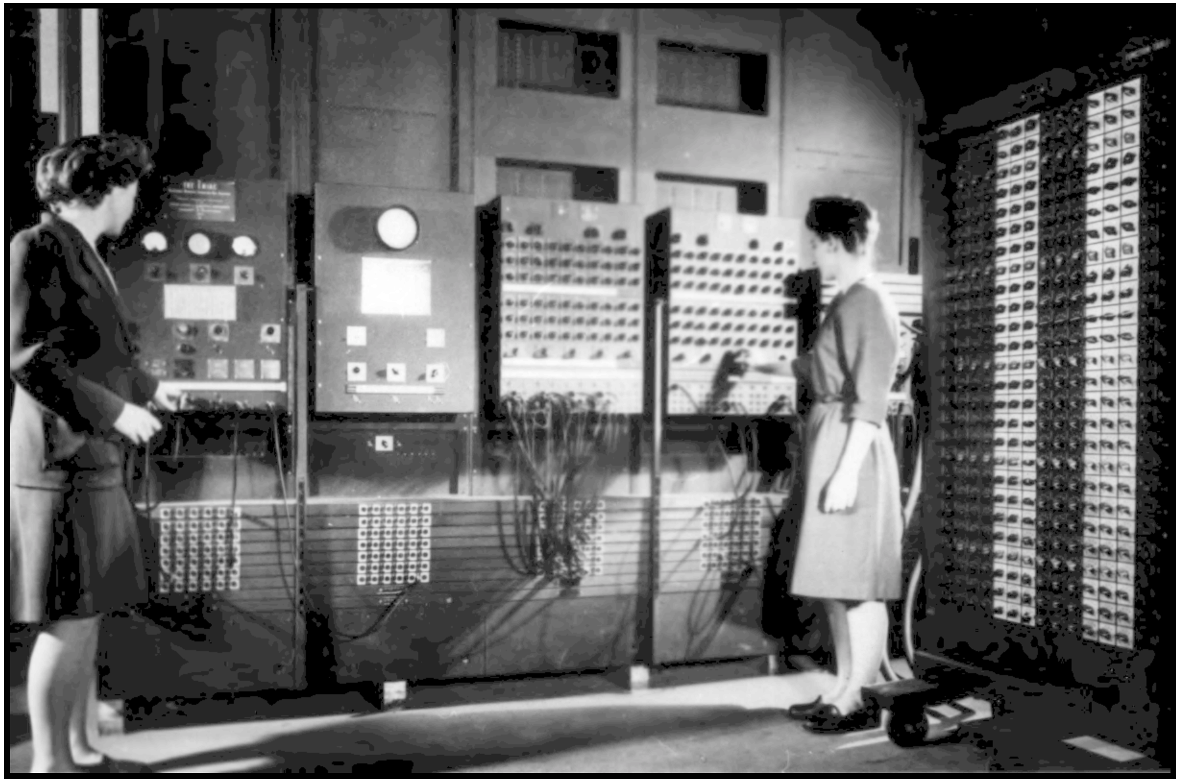
1) 2005 - 2025
2) 1985 - 2005
3) 1965 - 1985
4) 1945 - 1965

Charles University

Historie

Programování v letech 1950–1980

ENIAC (1945)

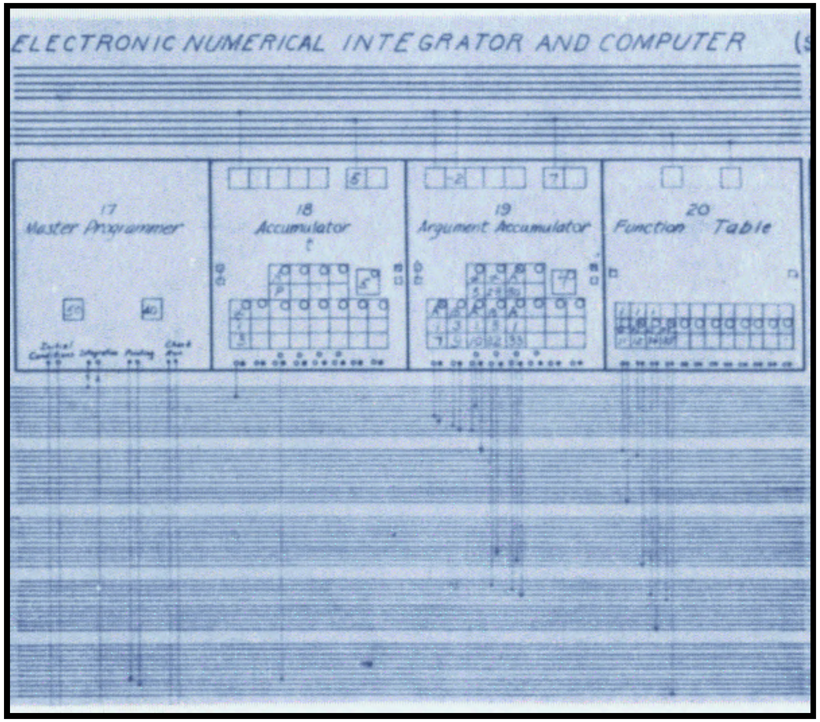


Frances Bilas, Betty Jennings

"The ENIAC was a son of a bitch to program."

Charles University

Simulace balistické křivky



Programování

Matematický popis problému

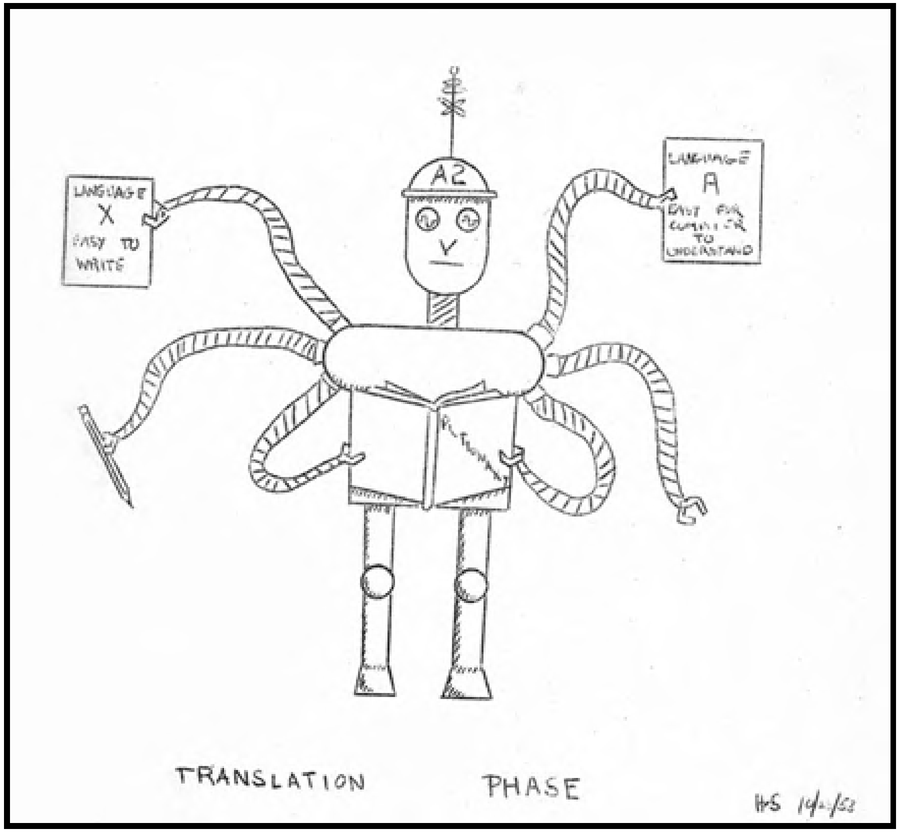
"Tradiční těžká práce"

Kódování

Převedení na řešení v počítači

"Lehký úkol pro sekretářku"

Charles University



Programovací jazyky

Překlad z jazyka snadného pro lidi do jazyka snadného pro počítače

FLOW-MATIC eliminuje problém kódování!

A2, FLOW-MATIC, MATH-MATIC, FORTRAN, COBOL, LISP, Algol, ...

NATO Software Engineering '68

Černá magie programování musí ustoupit vědeckému softwarovému inženýrství!

Softwarové inženýrství už je mužská práce...

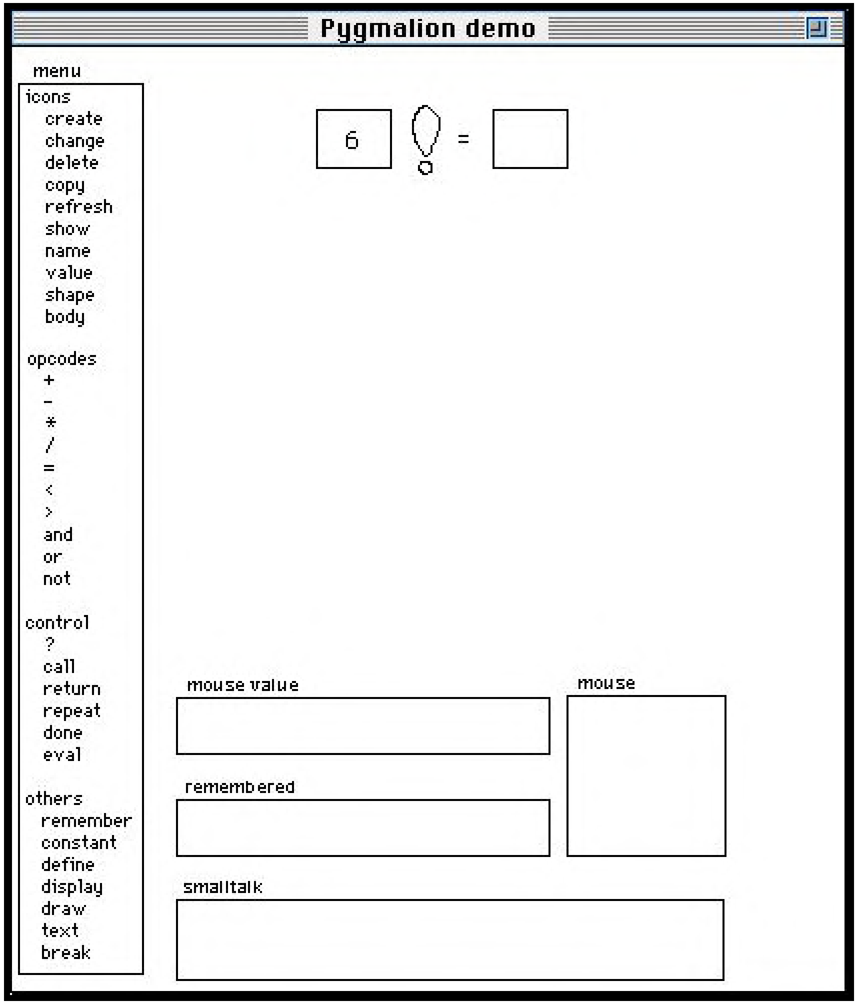
Charles University

Xerox PARC (1970s)

Osobní počítače
Grafické rozhraní
Okna a ikony
Objektové programování

Šlo by programovat interatktivně a vizuálně?

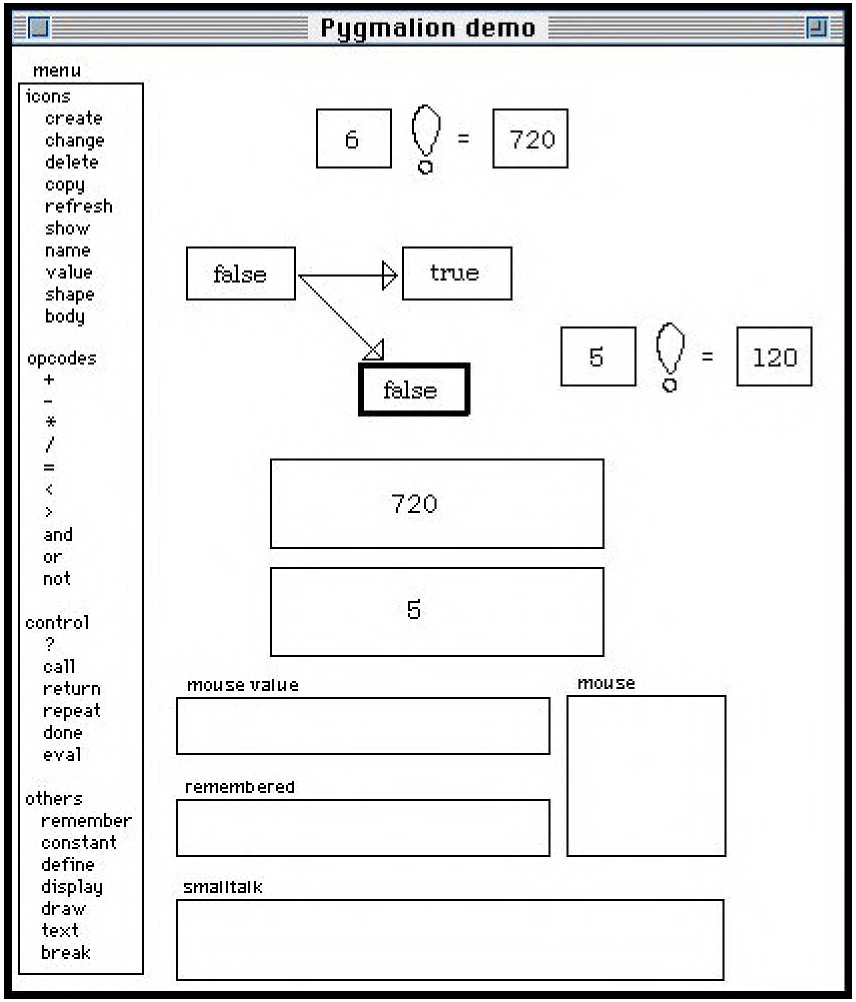
Pygmalion (1975)



Programming by demonstration

Ukážeme systému jak má postupovat na konkrétním příkladu...

Pygmalion (1975)



Programming by demonstration

Ukážeme systému jak má postupovat na konkrétním příkladu...

Charles University

Demo

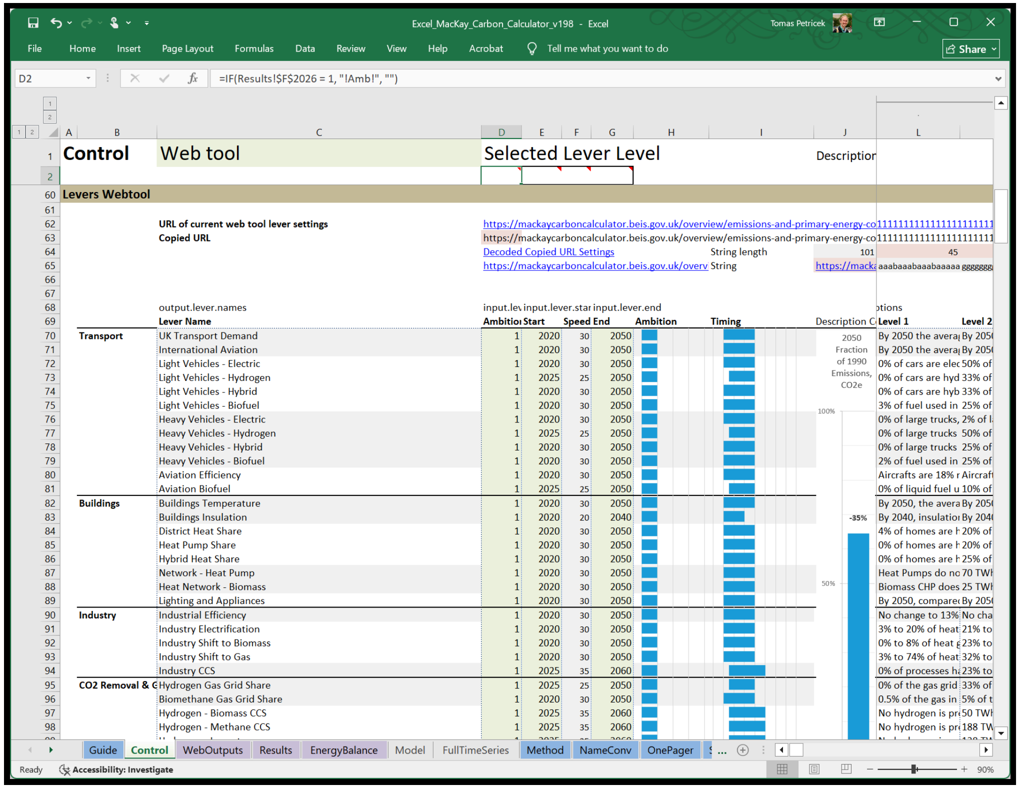
Reimplementace Pygmalionu

Současnost

Programování pro neprogramátory

Charles University

Uhodnete...



Co je nejpoužívanější programovací nástroj na světě?

Microsoft Excel!

Pro většinu lidí jen tabulka, ale umí více než byste čekali...

Charles University

Programování pro neprogramátory

"Excel"

Grafické rozhraní
Interaktivní přístup
Konkrétní hodnoty

"ChatGPT"

Generování kódu
Standardní jazyk
Další AI nástroje...

Charles University

Demo: Práce s daty

Letecké nehody v EU

Preview | Code | Blame

Search this file

	freq,unit,victim,c_regis,geo\TIME_PERIOD	2019	2020	2021	2022	2023
1	freq,unit,victim,c_regis,geo\TIME_PERIOD	2019	2020	2021	2022	2023
2	A,NR,KIL,EU27_2020,AT	0	0	0 p	1 p	0 p
3	A,NR,KIL,EU27_2020,BE	0	0	0 p	0 p	0 p
4	A,NR,KIL,EU27_2020,BG	0	0	0 p	0 p	0 p
5	A,NR,KIL,EU27_2020,CH	0	0	0 p	0 p	0 p
6	A,NR,KIL,EU27_2020,CY	0	0	0 p	0 p	0 p
7	A,NR,KIL,EU27_2020,CZ	0	0	0 p	0 p	0 p
8	A,NR,KIL,EU27_2020,DE	0	0	0 p	0 p	0 p
9	A,NR,KIL,EU27_2020,DK	0	0	0 p	0 p	0 p
10	A,NR,KIL,EU27_2020,EE	0	0	0 p	0 p	0 p
11	A,NR,KIL,EU27_2020,EL	3	0	0 p	1 p	0 p
12	A,NR,KIL,EU27_2020,ES	9	0	0 p	0 p	0 p
13	A,NR,KIL,EU27_2020,EU27_2020	21	3	0 p	17 p	2 p
14	A,NR,KIL,EU27_2020,EU28	21	:	:	:	:
15	A,NR,KIL,EU27_2020,FI	0	0	0 p	0 p	0 p

První sloupec obsahuje metadata

: znamená chybějící

p znamená předběžné

Řádky 13 a 14 jsou součty EU27 vs EU28 - Brexit!

Charles University

Demo

Práce s daty v nástroji Denicek

Charles University

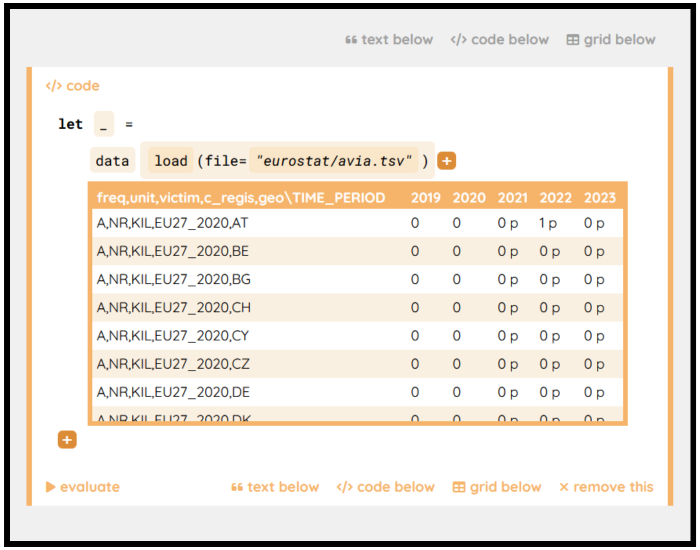
Práce s daty v nástroji Denicek

Způsob práce

- Načteme konkrétní data
- Ruční editace dat
- Program je zobecněná verze

Co to znamená

- Data máme stále před sebou
- Musíme přemýšlet co děláme
- Neumí to dělat složitější věci..



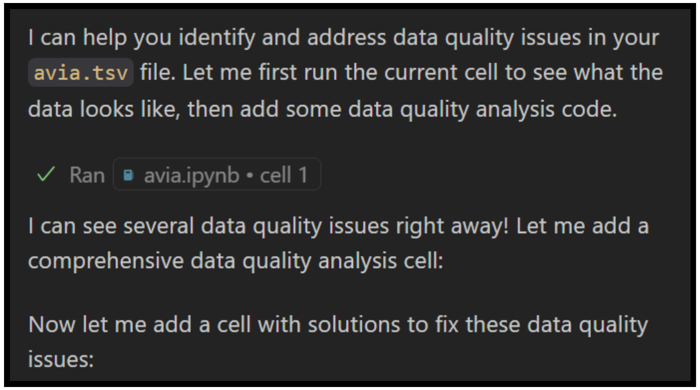
Demo

Práce s daty pomocí Copilot

Práce s daty pomocí Copilot

Způsob práce

- AI vygeneruje kód pro nás
- Lze zkoušet, dovysvětlit, měnit
- Konvenční programovací jazyky



Co to znamená

- Funguje i na celé aplikace!
- Nemusím umět programovat?
- Je to dost kódu. Je správně?

Programování

Co je vlastně programování (1/2)

Charles University

Překlad mezi jazyky

Neformální představa

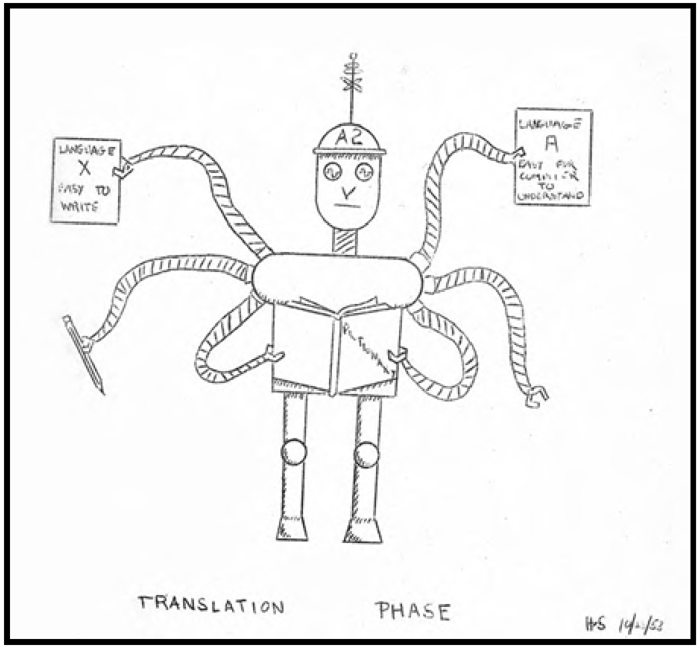
- Více či méně přesné zadání
- Chybí spousta detailů!

Formální high-level popis

- Přesný program pro počítač
- Některé detaily nemusíme řešit

Low-level instrukce

- Jednotlivé kroky počítačě
- Kóduje vše do naprostého detailu



Charles University

Překlad mezi jazyky

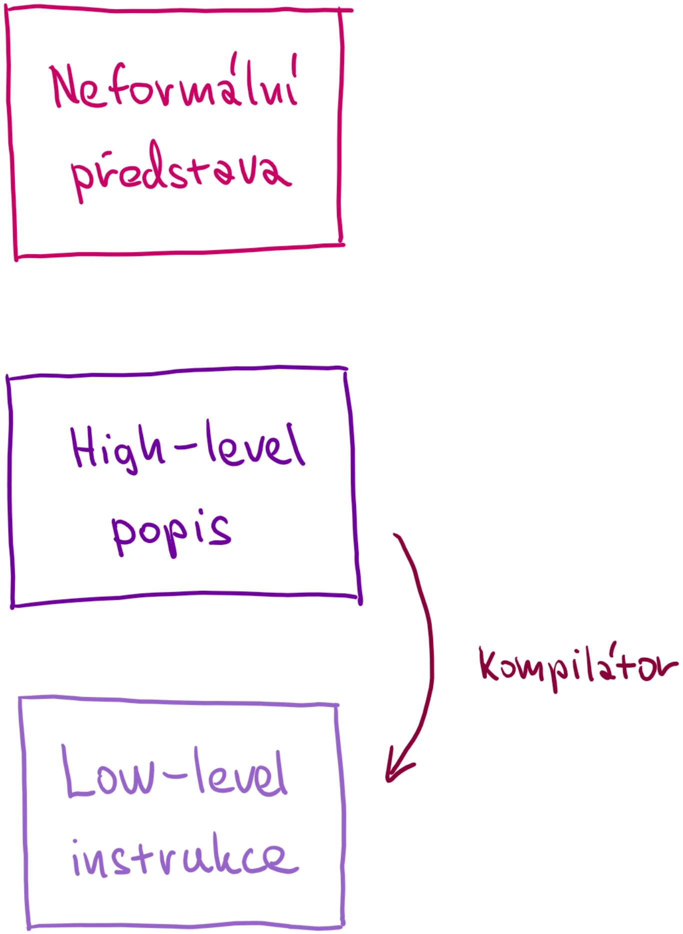
Neformální představa

High-level popis

Low-level instrukce

Charles University

Překlad mezi jazyky

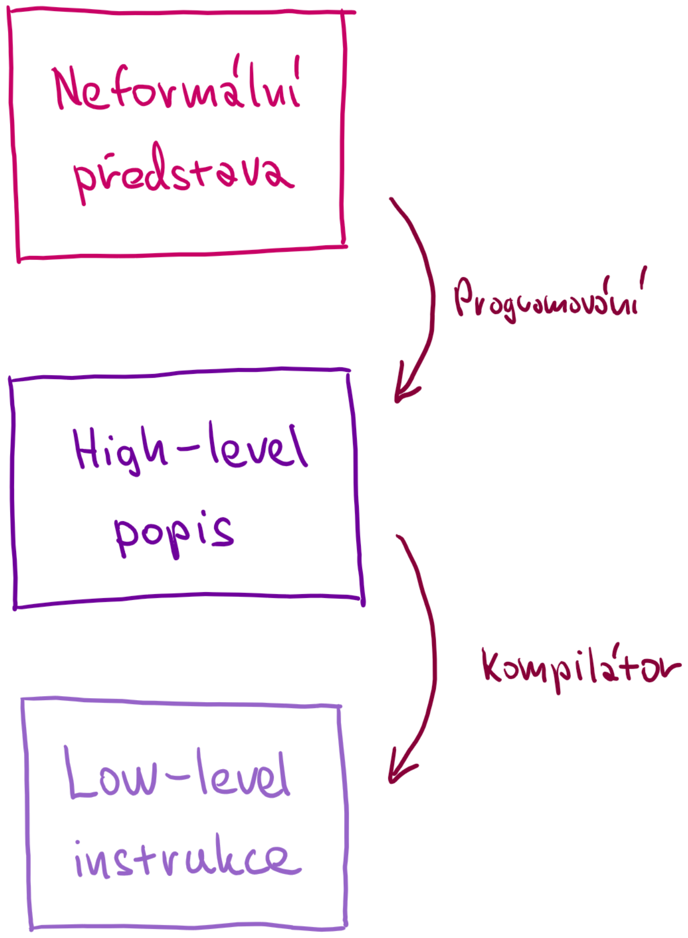


Dva jazyky s formální syntaxí a sémantikou

Kompilátor

Musí zachovat význam

Více či méně efektivní

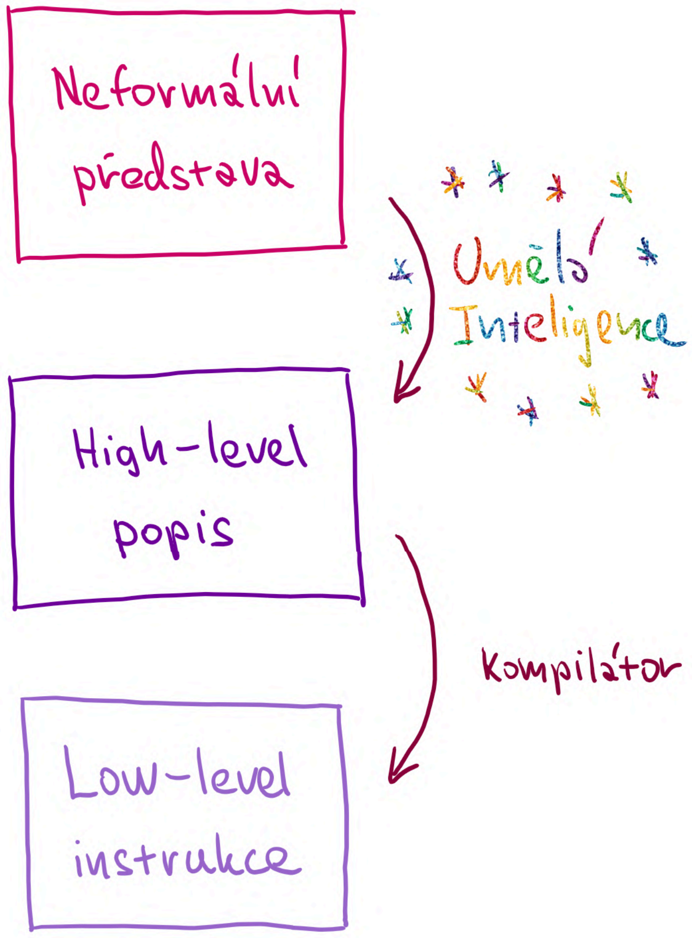


Dva přístupy k softwarovému inženýrství

Napsat specifikaci
Doplnit a domyslet vše co není formální

Postupný proces
Postupně zjišťujeme více, iterativní vývoj

Vyřeší to AI za nás?



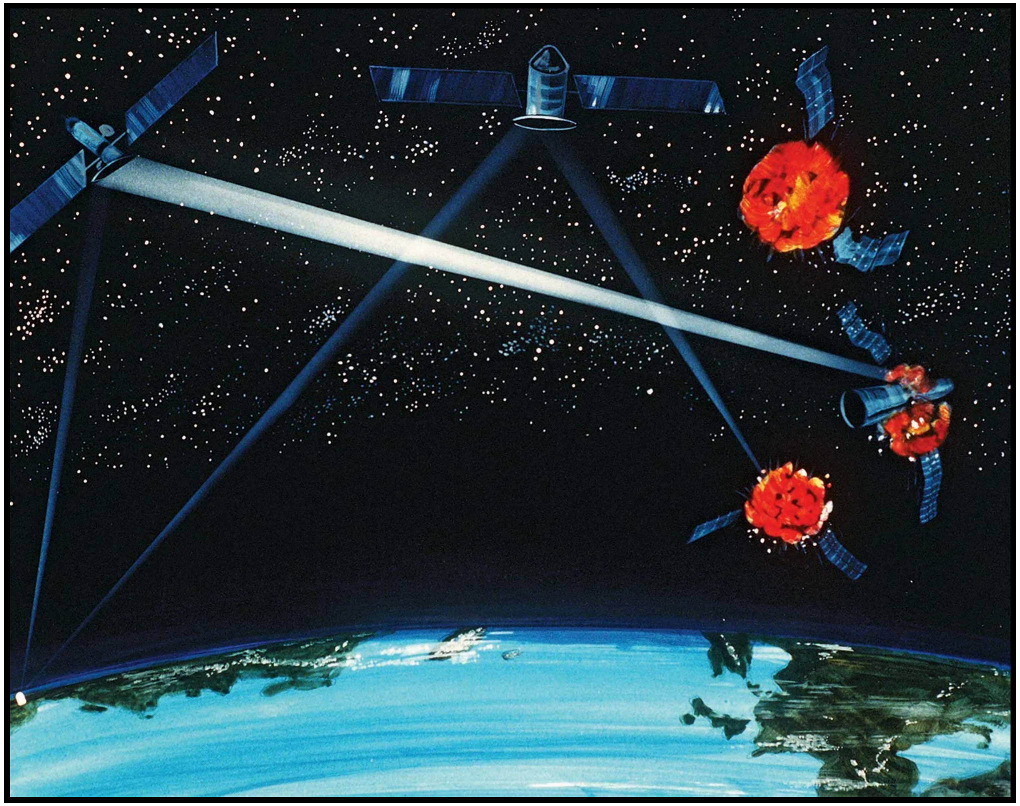
Těžko říct ale...

1) Krok zahrnuje dost lidského přemýšlení

2) Formální detaily se musí nějak doplnit

3) Svět je složitý!

Charles University



Strategic Defence Initiative (1980s)

Je možné napsat software na obranu proti balistickým raketám?

Ne, protože prostředí se mění rychleji...

Charles University

Programování

Co je vlastně programování (2/2)

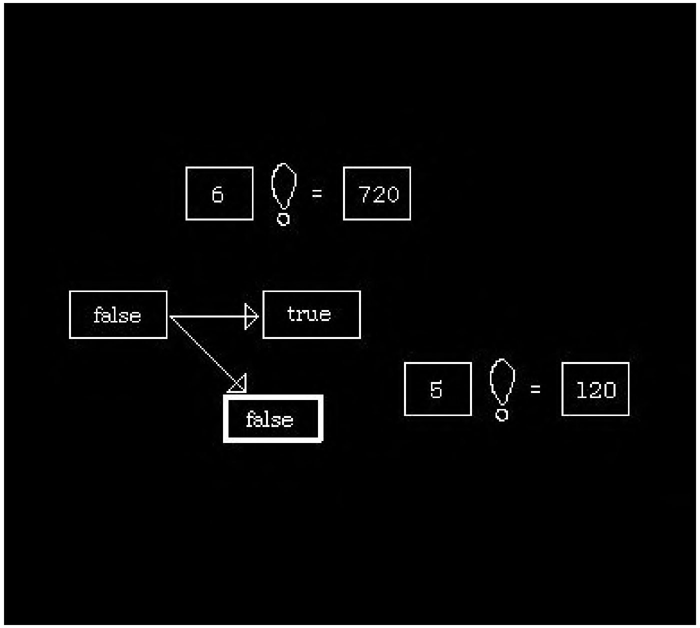
Demo

Projekt Sculpin

Interakce s prostředím

Iterativní vývoj

- Postupná adaptace běžícího a fungujícího systému
- Kombinace kódu a interakcí



Pro neprogramátory?

- Interakce jednodušší než kód
- AI se stejně hodí, ale jinak...
- Zbývá dost práce 😊

Charles University

Programování pro neprogramátory:
Nedosažitelný sen nebo blízká realita?

- Snažíme se o to už nějakých 70 let...
- Programování je o pochopení ne o kódu
- Pochopení bude potřeba vždycky...
- Problém je dobrý způsob interakce!

Tomas Petricek, MFF UK, Praha
tomas@tomasp.net | tomasp.net | @tomasp.net