

# Neprogramátor

Jiří Vlasák

Listopad 2024

# Kamarád gymnazista

- Vystudoval gymnázium
- Pracuje v archivu
- Chce se naučit programovat

# Skripta pro začátečníky

```
using System;

namespace Namespace
{
    /// <summary>
    /// Summary description for Class1.
    /// </summary>
    class Class1
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main(string[] args)
        {
            //
            // TODO: Add code to start application here
            //
        }
    }
}
```

## How to Design Programs

# Podmínkové výrazy

- if ještě jakž-takž
- cond, to už je moc

# Jaký problém řešíme?

Jak vysvětlit programování lidem s nízkým technickým povědomím?

... že programování je čarování, že programátoři zvládnou udělat věci, které normální lidé nezvládnou... (z dokumentu The Internet's Own Boy)

- Slovní zásoba a skladba věty, syntax
- Význam slov a vět, sémantika

# Český jazyk

## Syntax

---

Podstatná jména ("kniha")	Příslovce
Přídavná jména ("kouzelná")	Předložky
Zájmena	Spojky
Číslovky (0 a 3.14)	Částice
Slovesa (sečti a spoj)	Citoslovce

---

- Každá věta začíná velkým písmenem
- Následují slova oddělená mezerami (detaily skladby věty zanedbáme)
- Věta končí tečkou ., vykřičníkem !, nebo otazníkem ?

## Sémantika

- Učíme se od narození
- Zjednodušíme: význam slov chápeme intuitivně



# Příklad vět v češtině

Třeba:

- Sečti 0 a 3.14!
- Spoj "kouzelná" a "kniha".

Což ale zní dost kostrbatě, spíše bychom použili složitější větnou skladbu:

- Prosím, sečti pro mě čísla nula a 3.14.
- Co vznikne, když ke slovu "kouzelná" přidáš slovo "kniha"?

# Kouzelný jazyk Racket

## Syntax

- 
- Podstatná jména ("kniha", "kouzelná", 0 a 3.14)
  - Slovesa (+ nebo string-append)
- 

- Každá věta začíná závorkou (
- Hned po závorce následuje sloveso
- Potom jsou podstatná jména oddělená mezerami
- A úplně nakonec zase závorka )

## Sémantika

- Přesně daná specifikací jazyka
- Můžeme odhadnout

# Příklad vět v Racket

- `(+ 0 3.14)`
- `(string-append "kouzelná" "kniha")`

# Pozor na význam!

V češtině:

- Prosím, sečti pro mě nulu a "kniha".

V Racket:

- (+ 0 "kniha")

Jaký je tedy rozdíl mezi čarodějem a neprogramátorem?

- Oba používají jazyky
- Každý jazyk má slova a skladbu věty, syntax
- Slova a věty mají význam, sémantiku

# Kouzelná hůlka

Vykoná kouzelnou větu:

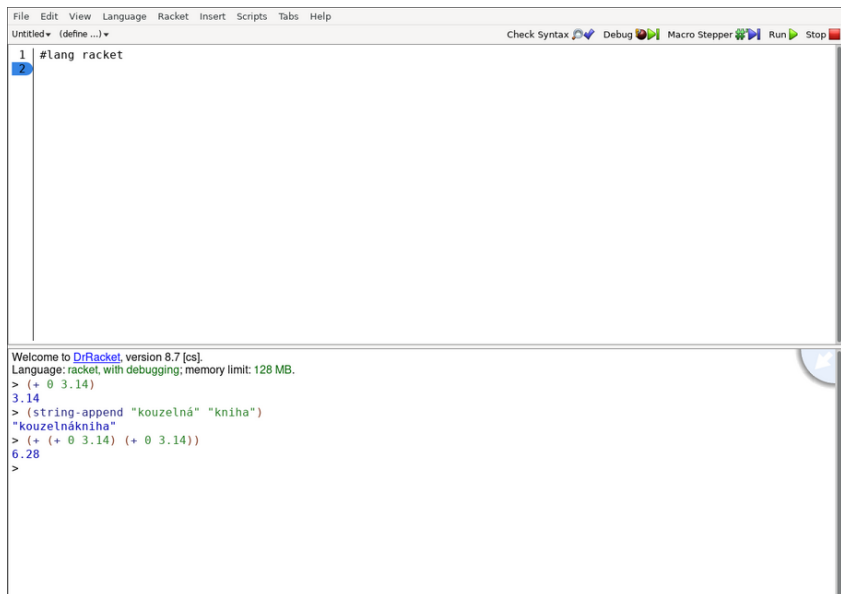
- `(+ 0 3.14) => 3.14`
- `(string-append "kouzelná" "kniha") => "kouzelnákniha"`

Výsledek vykonání kouzelné věty je opět kouzelné slovo (podstatné jméno nebo sloveso) – dá se tedy znovu použít:

- `(+ (+ 0 3.14) (+ 0 3.14)) => 6.28`

ChatGPT není kouzelná hůlka pro český jazyk!

# Kouzelná hůlka DrRacket



The screenshot displays the DrRacket IDE interface. The top menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The title bar shows 'Untitled' and '(define ...)'. The toolbar contains icons for Check Syntax, Debug, Macro Stepper, Run, and Stop. The code editor shows the following code:

```
1 #lang racket
2
```

The console window at the bottom displays the following output:

```
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (+ 0 3.14)
3.14
> (string-append "kouzelná" "kniha")
"kouzelnákniha"
> (+ (+ 0 3.14) (+ 0 3.14))
6.28
>
```

# Neprogramátor

## Dialog s kouzelnou hůlkou

- Podstatná jména: primitivní datové typy
- Pojmenování informací: definice proměnných
- Vykonat či nevykonat: predikáty a podmínkové výrazy
- Nová slovesa: definice funkcí

## Dopis pro kouzelnou hůlku

- Čarodějnické řemeslo: zdrojové kódy a repozitář
- Kouzelné lektvary: spustitelné soubory a testy

## Nekonečně kouzelné jazykolamy

- Kouzelné sloveso je kouzelné sloveso: rekurze
- Nekonečná podstatná jména: seznamy