

# Když kód myslí: Programování s umělou inteligencí

---

Michal Vašinek

Katedra informatiky FEI

VŠB – Technická Univerzita Ostrava

“

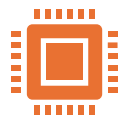
Někdy mám pocit, že mi čte myšlenky a překládá je na kód.



Některé myšlenky jsem ještě ani nevyslovil.

# Východiska

---



**59% vývojářů používá AI při své práci**

42% využívá chatboty k zodpovídání dotazů ohledně programování



**25% vývojářů přidává AI funkcionalitu do svých aplikací**



**13% vývojářů je zodpovědných za vývoj nových modelů nebo úpravu stávajících**



**29% vývojářů nepoužívá AI vůbec**



**27% vývojářů používá AI pomocníky ve vývojářských nástrojích**

Data za Q1 2024

[How developers Interact with AI Technologies](#)

# Základní nástroje

## Chatboti



## Asistenti kódování

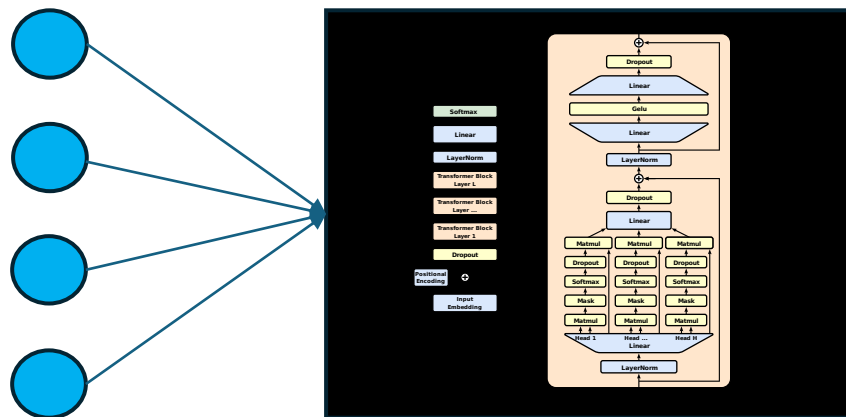


# Umělá inteligence – predikce dalšího slova

Umělá inteligence je obor ...

**jaké bude další slovo?**

Umělá  
inteligence  
je  
obor



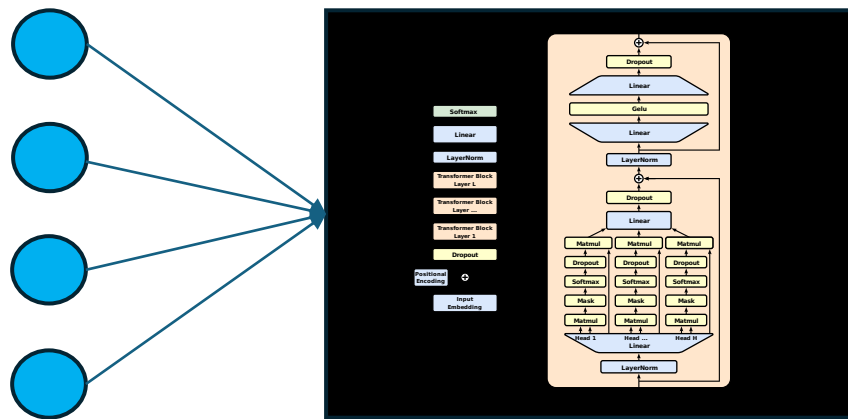
Neuronová síť složená z  
Transformer bloků

# Umělá inteligence – predikce dalšího slova

Umělá inteligence je obor ...

**jaké bude další slovo?**

Umělá  
inteligence  
je  
obor



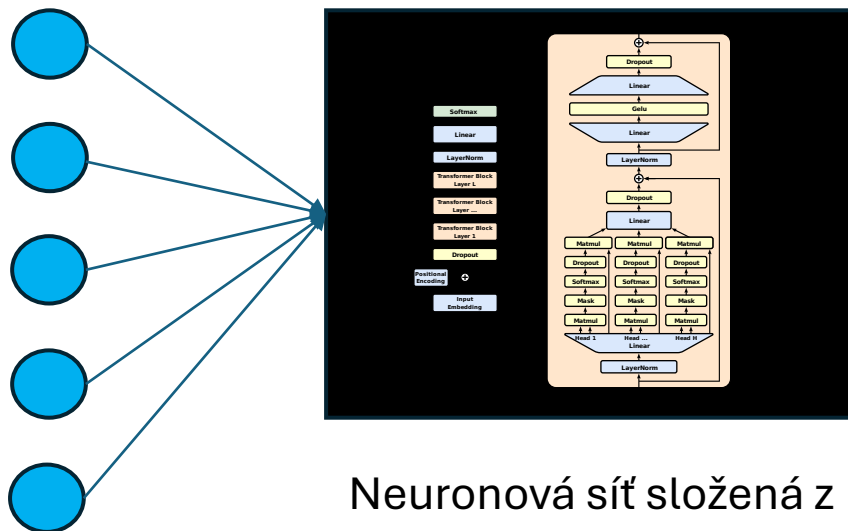
Neuronová síť složená z  
Transformer bloků

Slovo	P(slovo vstup)
informatiky	0,45
matematiky	0,12
lingvistiky	0,000007
veterinářství	0,0000005
počítačových	0,02
koťátko	0,000000001
...	...

# Umělá inteligence – predikce dalšího slova

Umělá inteligence je obor **informatiky** ... **jaké bude další slovo?**

Umělá  
inteligence  
je  
obor  
**informatiky**

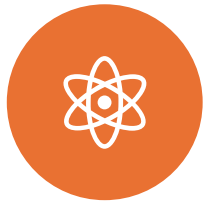


Neuronová síť složená z  
Transformer bloků

Slovo	P(slovo vstup)
zabývající	0,2
který	0,07
ve	0,02
,	0,12
.	0,05
koťátko	0,000000001
...	...

# Chatboti

---



GENEROVÁNÍ KÓDU

=

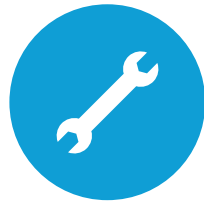
VYTVOŘ MI  
NAPIŠ MI



ÚPRAVA KÓDU

=

NÁVRH EFEKTIVNĚJŠÍ  
IMPLEMENTACE



DEBUGGING A  
DIAGNOSTIKA CHYB

=

PROGRAM MI VRACÍ  
CHYBU ..., JAK JI MOHU  
OPRAVIT



NÁVRH

=

NAVRHNI MI TŘÍDY PRO  
APLIKACI, JAKÉ JSOU  
KNIHOVNY PRO ŘEŠENÍ  
PROBLÉMU, POPIŠ MI  
JEJICH VÝHODY A  
NEVÝHODY



VYSVĚTLENÍ KÓDU

=

CO DĚLÁ TENTO KUS  
KÓDU



DOKUMENTACE

=

JAKÉ PARAMETRY MÁ  
TATO FUNKCE



# GitHub Copilot

- Asistent pro kódování poháněný umělou inteligencí má 1,3 milionu platících předplatitelů. (odhad cca 29 mil. vývojářů na světě)
  - Satya Nadella (chairman, CEO – Microsoft)
- Našeptává kód
- [Pro studenty a učitele zdarma.](#)
- Ostatní 10\$ měsíčně.



# GitHub Copilot

whisper.py > Zprava

```
1  """
2  Trida pro reprezentaci zprav v aplikaci
3  """
4  ✨
5  class Zprava:
    def __init__(self, text, uzivatel):
        self.text = text
        self.uzivatel = uzivatel

    def __str__(self):
        return f"{self.uzivatel}: {self.text}"

    def __repr__(self):
        return f"Zprava({self.text}, {self.uzivatel})"
```

# GitHub Copilot

whisper.py > Zprava

```
1  """
2  Trida pro reprezentaci zprav v aplikaci
3  """
4  ✨
5  class Zprava:
    def __init__(self, text, uzivatel):
        self.text = text
        self.uzivatel = uzivatel

    def __str__(self):
        return f"{self.uzivatel}: {self.text}"

    def __repr__(self):
        return f"Zprava({self.text}, {self.uzivatel})"
```

Víte, jak poznáte  
klávesnici vývojáře  
používajícího umělou  
inteligenci při psaní  
kódu?

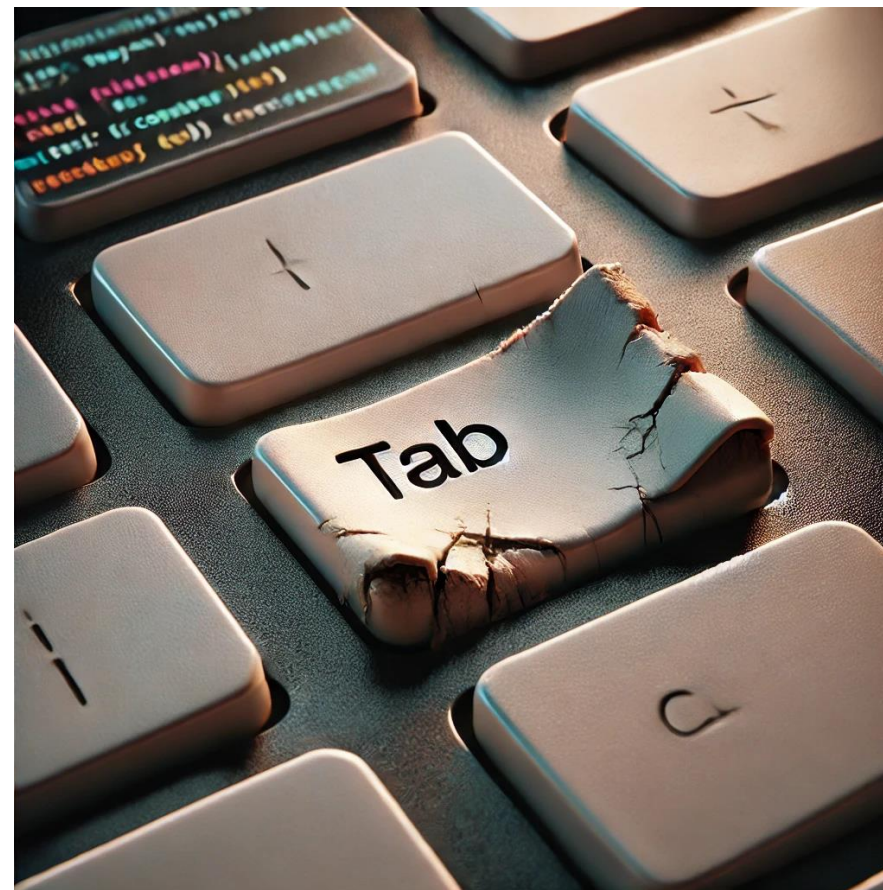
# GitHub Copilot

whisper.py > Zprava

```
1  """
2  Trida pro reprezentaci zprav v aplikaci
3  """
4  ✨
5  class Zprava:
    def __init__(self, text, uzivatel):
        self.text = text
        self.uzivatel = uzivatel

    def __str__(self):
        return f"{self.uzivatel}: {self.text}"

    def __repr__(self):
        return f"Zprava({self.text}, {self.uzivatel})"
```



# GitHub Copilot ve VS Code - Kontext



Vše je o kontextu!



Čím více kódu Copilot vidí, tím relevantnější odpověď obdržíte.



Otevřít, co nejvíce relevantních záložek.

# Nejdřív komentuj, pak kóduj

```
class AutoNoiseMap:
    def __init__(self, ref_cmap, auto_cmap):
        self._r_cmap = CMAP(ref_cmap)
        self._a_cmap = CMAP(auto_cmap)

        self._mapping = self.__map()
        self._idx_ra = self._makeIndex()

    def maps(self):
        return self._mapping

    def autoSiteID(self, cname, rid):
        return self._idx_ra[cname][rid]

    def _makeIndex(self):
        idx = {}

        for a, rs in self._mapping:
            if not a[0] in idx:
                idx[a[0]] = {}

                for r in rs:
                    idx[a[0]][r[1]] = a[1]

        return idx
```

```
"""
Třída reprezentující XMAPu.
Bude uchovávat záznamy typu XMAPRecord obsahující následující parametry:
- queryID
- refID - id chromozomu v referenci
- Cigar řetězec
- vzájemně mapování značek mezi query a referenci,
  které jsou zadány jako sekvence (ref_id, query_id)

Chování:
- třída bude umět vrátit query_id a dvojici značek,
  které budou odpovídat dvěma po sobě jdoucím značkám v referenci
"""

import re
import sys
from typing import Dict, List, Tuple

class XMAPRecord: ...

class XMAP: ...

"""
Otestování:
- na vstupu máme xmapu
- vytvoříme XMAP objekt
- zavoláme queriesForSites
"""
```

# Nejdřív komentuj, pak kóduj

Komentářem  
poskytujeme umělé  
inteligenci kontext, který  
umožní vygenerování  
relevantnějšího kódu

Automatické  
doplňování generuje  
kód s komentáři

Výsledek: přehledný,  
přenositelný a  
znovupoužitelný kód

# Další tipy

Ručně vkládejte importy

Pojmenovávejte proměnné, funkce podle toho co reprezentují, či co vykonávají, proměnná/funkce/třída by neměla být pojmenována **a**, **aux**, **temp**

Příklad: `def a(r)` vs. `def circleArea(radius)`

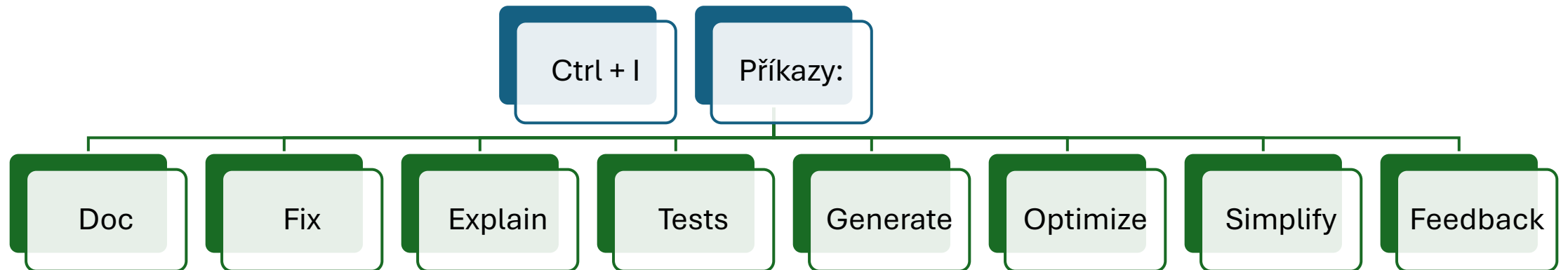
Doplňte `from math import pi`

Složitější problémy: rozděľ a panuj, komentujte postup výpočtu krok po kroku



# Copilot

---

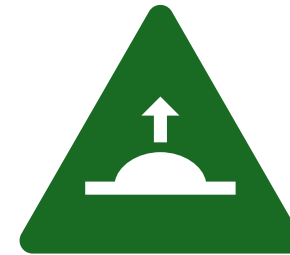


# Je vše jen super?

---



**Problém s bezpečným kódem**



**Ukázkový problém**

Napište funkci, která má na vstupu cestu k souboru a vrátí jeho obsah, ale pouze pokud se soubor vyskytuje v adresáři **safedir**.



# Psát nebo verifikovat?

- Asistenti snižují úsilí vynakládané na psaní kódu a nahrazují jej úsilím vloženým do jeho kontroly. Yan a kol. (2024)
  - Vzniká nová potřeba verifikovat vygenerovaný kód
  - Dle studie z MIT a Microsoftu tráví programátoři 22% času verifikací návrhů od AI
  - Ze studie také vyplynulo, že programátoři tráví aktivitami spojenými s asistentem 51% času.
-

# Děkuji za pozornost!

