

# POČÍTAČE A PROGRAMOVÁNÍ

Moderní metody vývoje softwaru,  
Demontrační příklad piškvorky

Miroslav Vavroušek

PPI 09

# Opakování z minulé přednášky

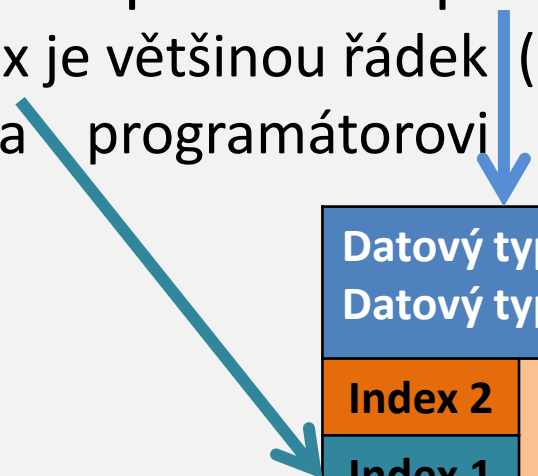
- Vícerozměrná statická a dynamická pole
- Pole polí

Datový typ Indexu 1 a 2: Integer Datový typ buňky: Integer				
Index 2				
Index 1	1	2	..	200
1	8	5	..	29
2	11	14	..	41
:	:	:	∴	:
500	82	21	..	6

Datový typ Indexu: Integer Datový typ buňky: Pole string	
Index	Data
1	Datový typ Indexu: Integer Datový typ buňky: String
	Index   Data
	1   'Ahoj'
	:
	$n_1$   '!'
2	Datový typ Indexu: Integer Datový typ buňky: String
	Index   Data
	1   'Dalsi pole'
	:
	$n_2$   'Stringu!'
:	:
200	Datový typ Indexu: Integer Datový typ buňky: String
	Index   Data
	1   'Posledni'
	:
	$n_{200}$   'v poli!'

# Vícerozměrná statická a dynamická pole

- Využívají dvou a více indexů
  - Indexy v hranatých závorkách jsou odděleny čárkami
- Mají dva a více rozměru
  - pro každý rozměr daný index
- Dvourozměrné pole lze reprezentovat tabulkou
  - První index je většinou řádek (Matematická konvence)
  - Volba je na programátorovi



Datový typ Indexu 1 a 2: Integer Datový typ buňky: Integer				
Index 2				
Index 1	1	2	..	200
1	8	5	..	29
2	11	14	..	41
:	:	:	∴	:
500	82	21	..	6

# Pole polí

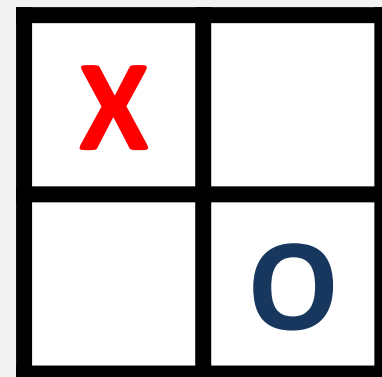
- Lze vytvořit pole, jehož prvky jsou pole
  - Statické i dynamické
  - Jedno i více rozměrné
  - Prvky mohou být pole
    - Statické i dynamické
    - Jedno i více rozměrné

Jednorozměrné pole indexované číslem (Integer), prvky jsou pole textů (Stringu) indexované číslem

Datový typ Indexu: Integer Datový typ buňky: Pole string		
Index	Data	
1	Datový typ Indexu: Integer Datový typ buňky: String	
	Index	Data
	1	'Ahoj'
	:	:
$n_1$	'!'	
2	Datový typ Indexu: Integer Datový typ buňky: String	
	Index	Data
	1	'Dalsi pole'
	:	:
$n_2$	'Stringu!'	
:	:	
200	Datový typ Indexu: Integer Datový typ buňky: String	
	Index	Data
	1	'Posledni'
	:	:
$n_{200}$	'v poli!'	

# Obsah přenášky

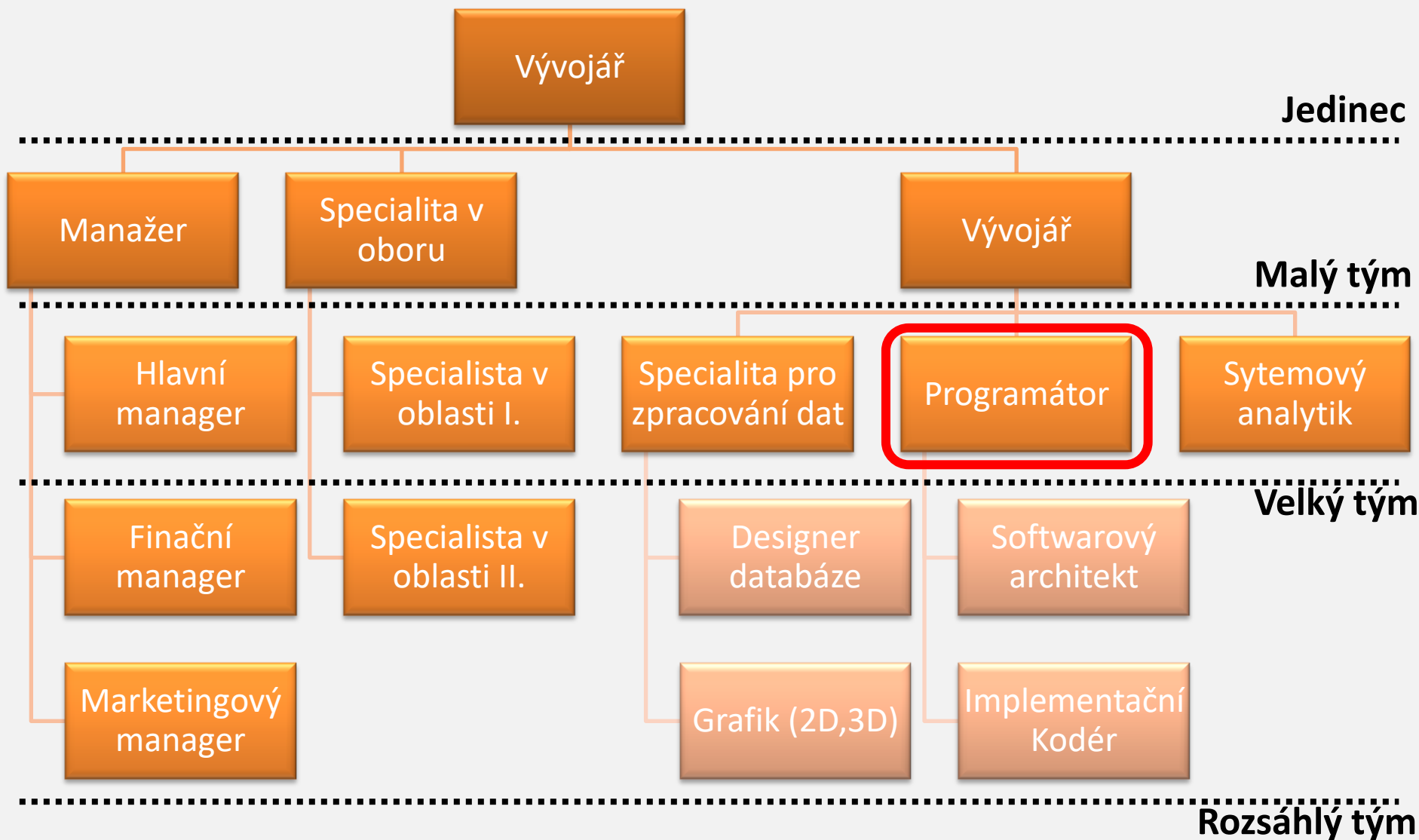
- Moderní metody vývoje softwaru
  - Složení softwarového vývojového týmu
  - Moderní nástroje pro vývoj softwaru
  - Metody řízení vývoje projektu
  - Možnosti popisu a dokumentace projektu
- Demostrační úloha piškvorky
  - Návrh a implementace aplikace realizující hru piškvorky pro dva hráče



# Složení softwarového vývojového týmu

# Moderní metody vývoje softwaru

Složení softwarového vývojového týmu (ilustrace)



# Moderní nástroje pro vývoj softwaru



# Moderní nástroje pro vývoj softwaru

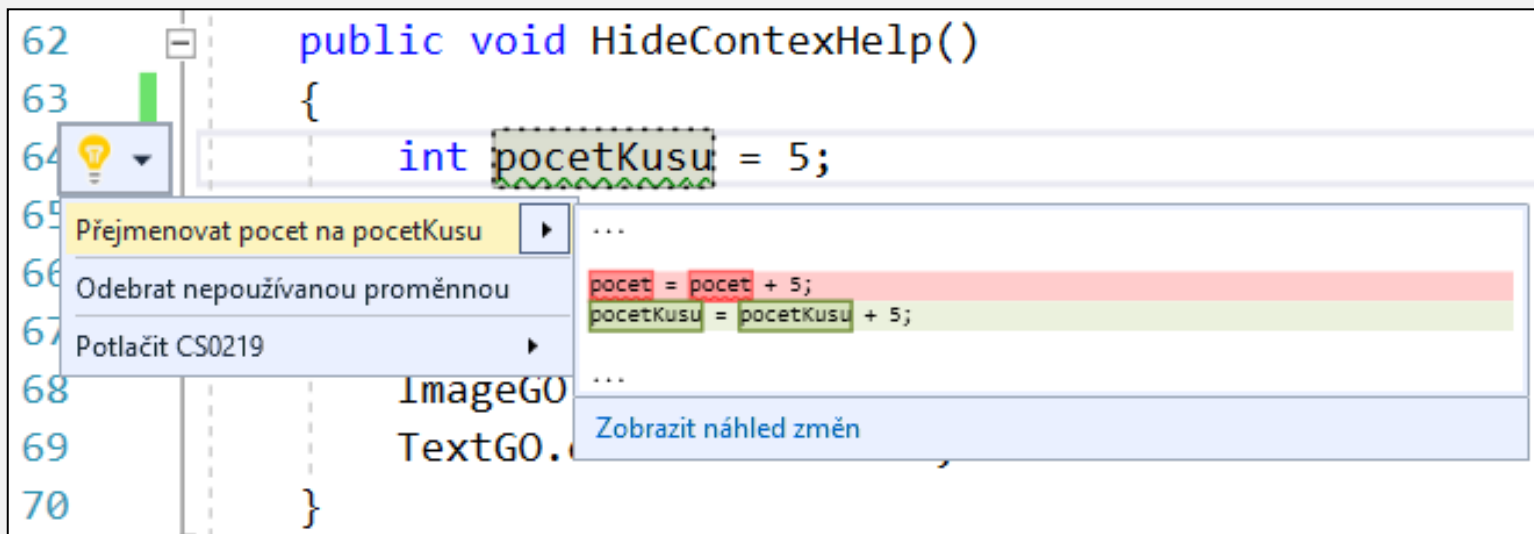
## Kooperace vývojového týmu

- Udržování projektu v aktuální verzi
  - Verzování
    - Projekt je označen verzí
    - Nutnost implementace změn od vývojáře
      - Konflikty řešení a změn
  - Specializované nástroje pro spravu projektu
    - Team Foundation Server, atd.
- Distribuce aktuální verze projektu
  - Sdílené disky
  - Vývojářské servery
  - Team Foundation Server


# Moderní nástroje pro vývoj softwaru

## Nástroje pro usnadnění vývoje

- Refaktorování
  - Změny v aplikaci, které nemají vliv na její chování
  - Proveděny automaticky vývojovým prostředím
  - Změna názvů atd. (všude kde se vyskytují)
  - Urdžují projekt srozumitelný a dobře čitelný



```
62 public void HideContextHelp()
63 {
64     int pocetKusu = 5;
65     ...
66     pocet = pocet + 5;
67     pocetKusu = pocetKusu + 5;
68     ...
69     ImageGO
70     TextGO.
71 }
```

64  Přejmenovat pocet na pocetKusu

66 Odebrat nepoužívanou proměnnou

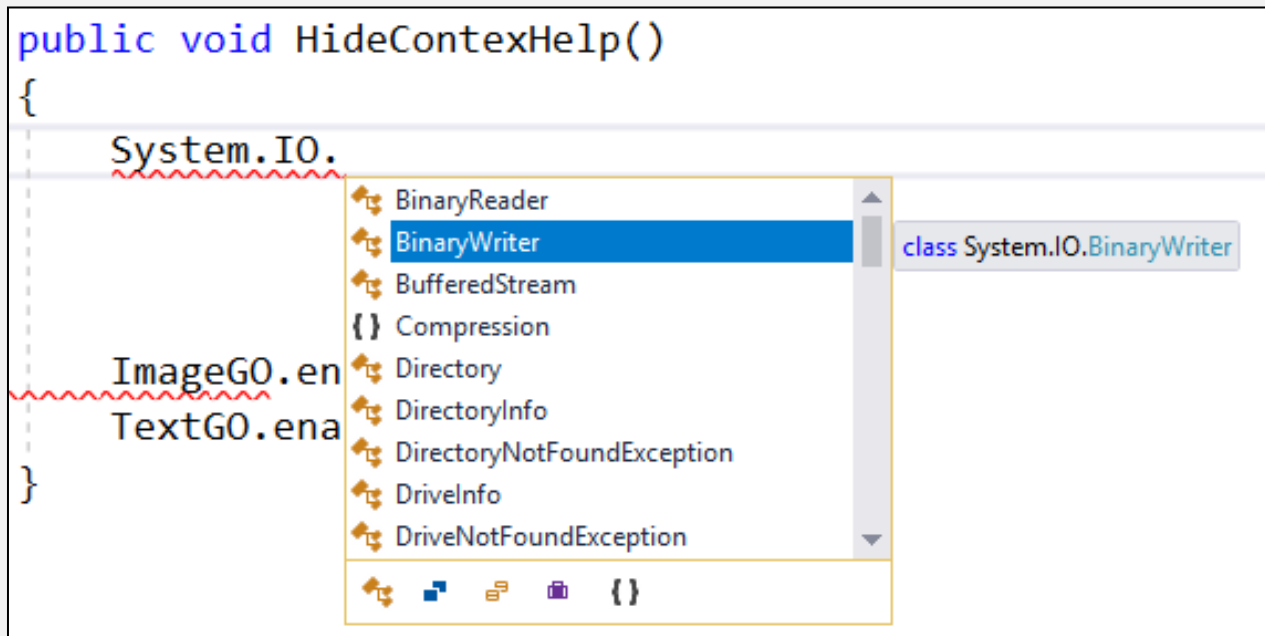
67 Potlačit CS0219

Zobrazit náhled změn

# Moderní nástroje pro vývoj softwaru

## Nástroje pro usnadnění vývoje

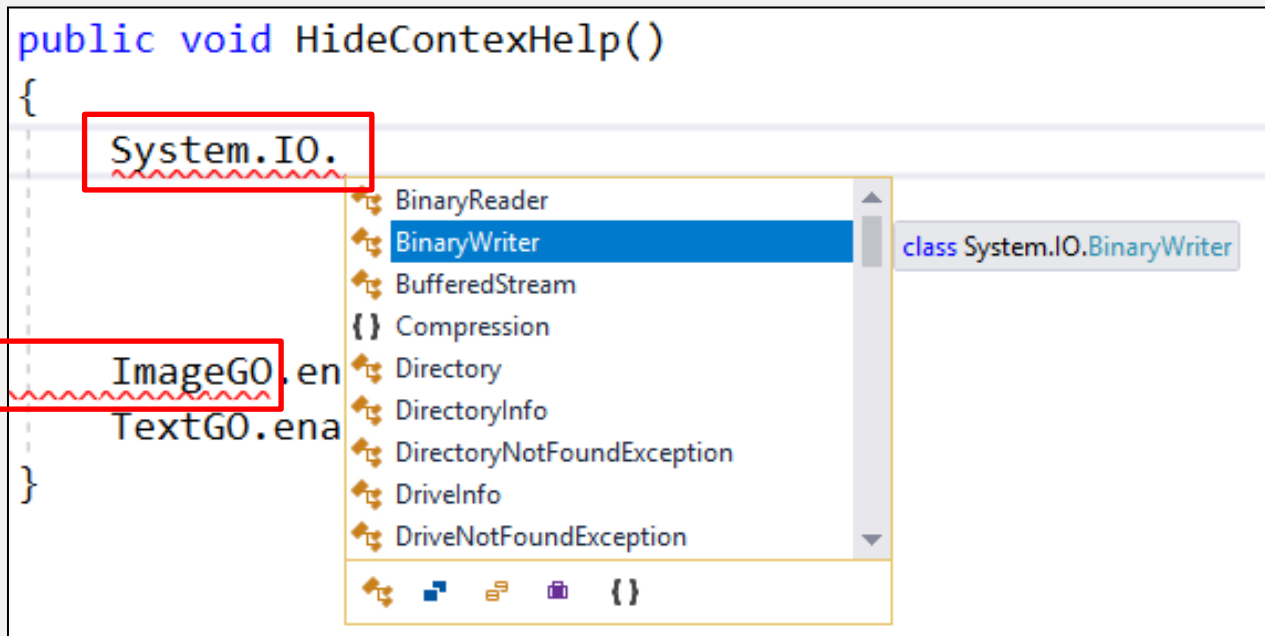
- Našeptávače
  - Usnadňují a urychlují zápis kodu
  - Při zapsání prvních písmen nabízí možné volby
  - Volby se kontextově mění podle aktuálního kódu



# Moderní nástroje pro vývoj softwaru

## Nástroje pro usnadnění vývoje

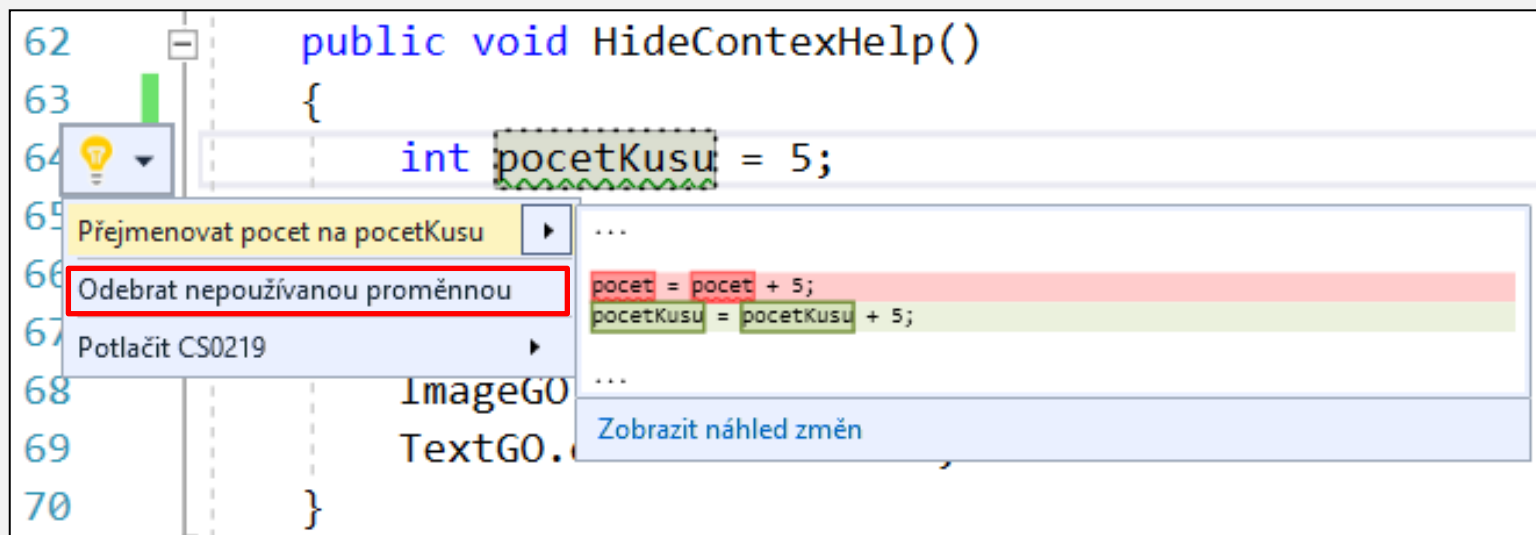
- Dynamická kontrola syntaxe
  - Neustále kontroluje kód a označuje chyby
    - Chyby nejsou oznámeny až při kompilaci
  - U některých chyb nabízí automatickou opravu



# Moderní nástroje pro vývoj softwaru

## Nástroje pro usnadnění vývoje

- Optimalizace kódu
  - Vyvojové prostředí nabízí optimalizaci kódu
    - Nevyužité proměnné
    - Nedosažitelné části kódu, atf.
    - Nepoužívané knihovny



The screenshot shows a code editor with a C# method `HideContextHelp()`. Line 64 contains the declaration `int pocetKusu = 5;`. A lightbulb icon is visible to the left of this line. A context menu is open over the variable `pocetKusu`, with the option "Odebrat nepoužívanou proměnnou" (Remove unused variable) highlighted in red. The menu also includes "Přejmenovat pocet na pocetKusu", "Potlačit CS0219", and "Zobrazit náhled změn". In the background, other code is visible, including `pocet = pocet + 5;` and `pocetKusu = pocetKusu + 5;`.

```
62 public void HideContextHelp()
63 {
64     int pocetKusu = 5;
65     ...
66     pocet = pocet + 5;
67     pocetKusu = pocetKusu + 5;
68     ImageGO
69     TextGO.
70 }
```

# Metody řízení vývoje projektu

# Moderní nástroje pro vývoj softwaru

Metody řízení vývoje projektu

Příprava projektu



Plánování projektu

Realizace projektu

Ukončení projektu

Poprojektová fáze

# Moderní nástroje pro vývoj softwaru

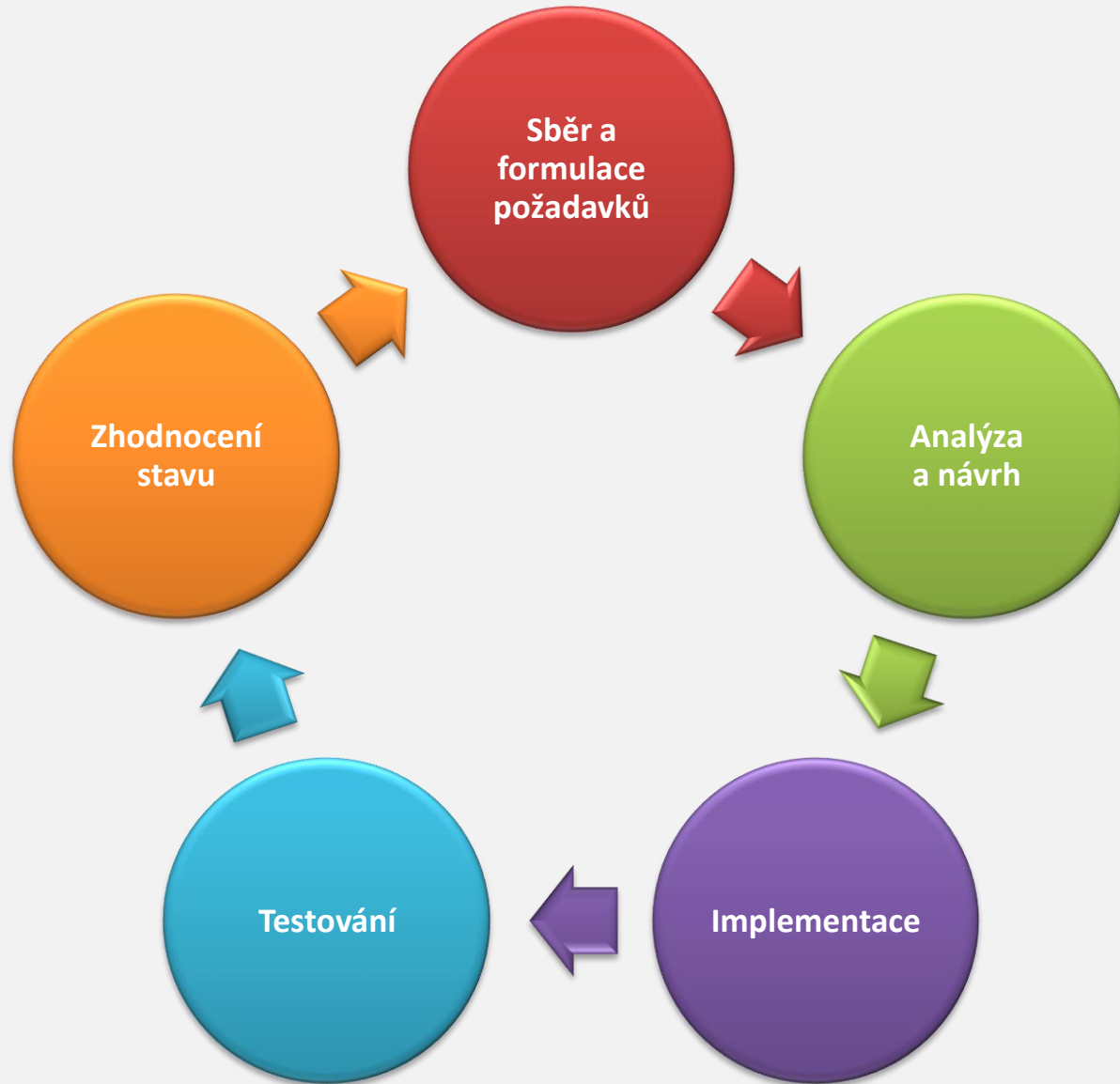
## Metody řízení vývoje projektu

- Programování shora-dolů (prototypování)
  - Řešení úlohy rozkladem na podúlohy (podproblemy)
  - Využívá (zatím neexistujících) podprogramů
- Programování zdola-nahoru (designování)
  - vytváření podprogramů (řešení podúloh),
    - Které budeme potřebovat
  - Následně z nich skládáme řešení větších podúloh, až k hlavní úloze



# Moderní nástroje pro vývoj softwaru

## Metody řízení vývoje projektu



# Moderní nástroje pro vývoj softwaru

## Metody řízení vývoje projektu

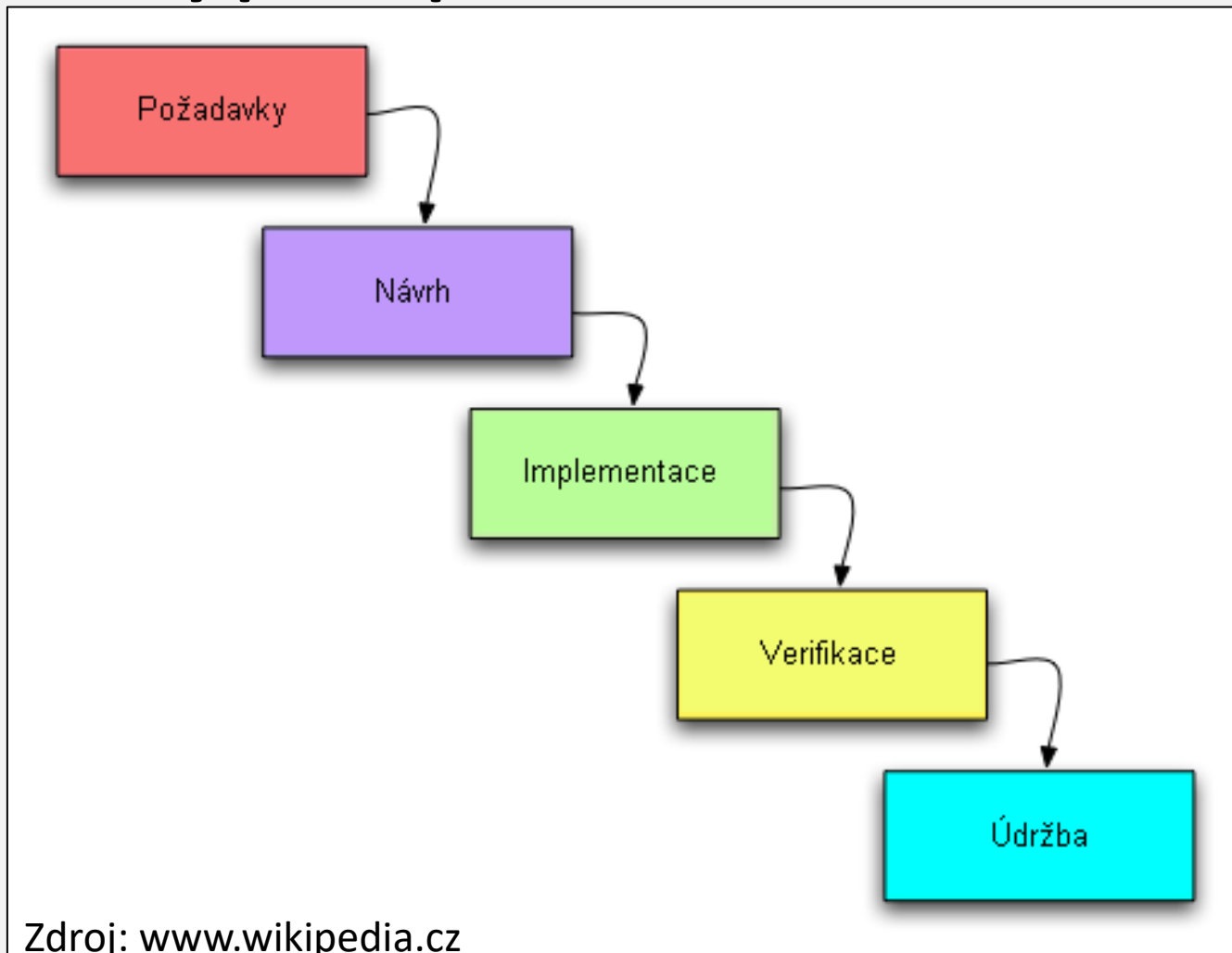
- **Vodopádový přístup**

- Projekt je rozdělen na fáze jdoucí postupně za sebou
  - Některé se mohou překrývat
- Důraz je kladen na plánování, časové rozvrhy, termíny, rozpočty
  - Realizace celého systému najednou
- Přísná kontrola je udržována po celou dobu životnosti projektu
  - Prostřednictvím využití rozsáhlých písemných dokumentů
  - Prostřednictvím revizí
  - Schvalování uživatelem a managementem na konci fází

# Moderní nástroje pro vývoj softwaru

## Metody řízení vývoje projektu

- **Vodopádový přístup**



# Moderní nástroje pro vývoj softwaru

## Metody řízení vývoje projektu

- **Inkrementální (přírůstkový) přístup**
  - Kombinace sekvenčních a iteračních metodik vývoje
    - Omezení projektových rizik rozdělením projektu na menší segmenty
    - Zjednodušuje možnost zavedení změn během procesu vývoje
  - Jsou prováděny série malých vodopádů
    - Každý vodopád je prováděn pro malou část systému a je dokončen před pokračováním na další přírůstek, nebo
  - Prvotní koncept, analýza požadavků, design architektury a systémové jádro
    - Jsou definovány vodopádovým přístupem
    - Následuje iterativní prototypový přístup

# Moderní nástroje pro vývoj softwaru

## Metody řízení vývoje projektu

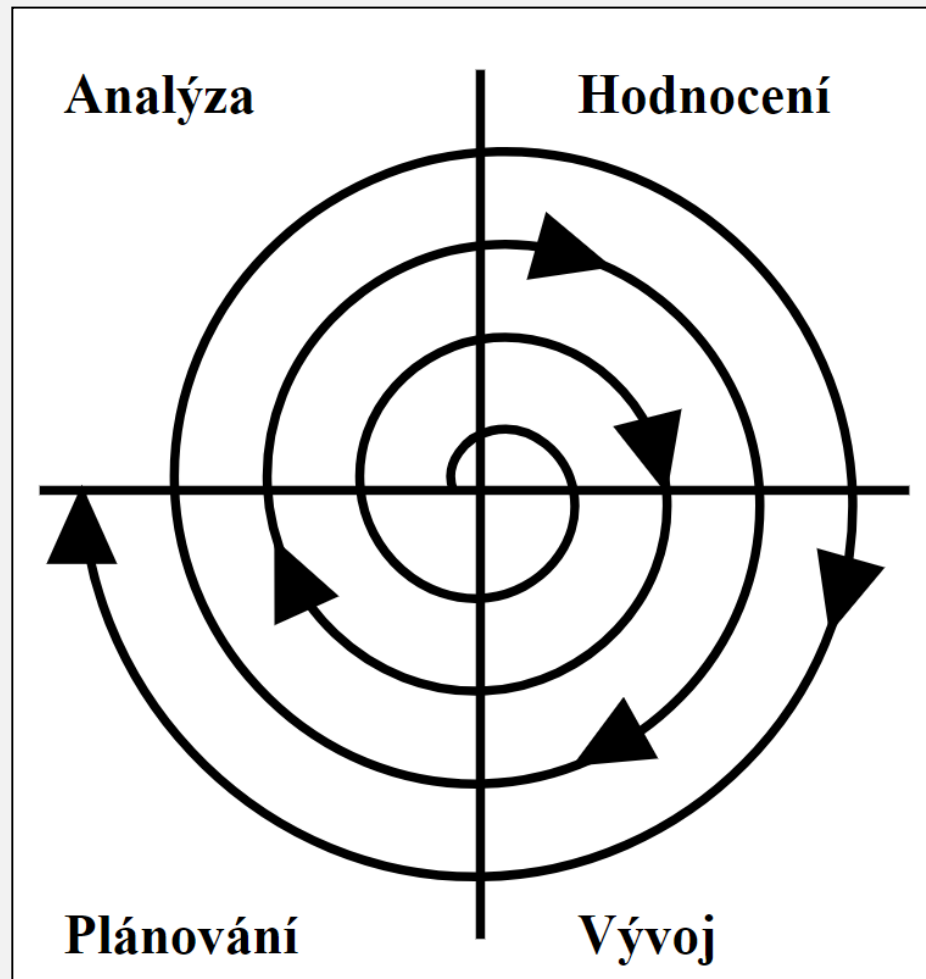
- **Spirálový přístup**

- Kombinuje prvky designového přístupu a prototypového přístupu
- Zaměřuje se na analýzu minimalizaci projektových rizik rozdělením projektu
  - Umožnění změn během procesu vývoje
  - Vyhodnocení rizik v průběhu vývoje
- Každý cyklus spirály spouští stejný sled kroků pro každou část produktu
  - Pro každou úroveň elaborace (od konceptuálních dokumentů až po programování jednotlivých programů)
- Během každého cyklu spirály jsou spouštěny čtyři základní fáze (kvadranty):
  - Analýza – stanovení cílů, alternativ a rozsahu iterace
  - Vyhodnocení – vyhodnocení alternativ, identifikace a řešení rizik
  - Vývoj – vývoj produktu a kontrola očekávaných výsledků
  - Plánování – plán pro příští iteraci
- V počátku každého cyklu se identifikují zainteresované subjekty a jejich podmínky kladené na úspěch iterace, na konci každého cyklu se provádí revize a předání.

# Moderní nástroje pro vývoj softwaru

Metody řízení vývoje projektu

- **Spirálový přístup**



# Moderní nástroje pro vývoj softwaru

## Metody řízení vývoje projektu

- Vývoj pomocí prototypů
  - Není samostatným a kompletním přístupem k vývoji
  - Vývoj neúplných verzí software, tzv. prototypů
  - Snaha snížit nebezpečí projektových rizik
    - rozdělením projektu na menší části a zjednodužit tak možnost změn
  - Uživatel je zapojen v celém procesu vývoje
    - To zvyšuje pravděpodobnost přijetí konečné implementace
  - Malé ukázky systému jsou vyvíjeny iterativním procesem
    - Dokud se prototyp nesplňuje požadavky uživatele

# Možnosti popisu a dokumentace projektu



# Moderní nástroje pro vývoj softwaru

## Možnosti popisu a dokumentace projektu

- Dokumentace
  - Popisuje realizaci a chování softwaru
  - Textové dokumenty, prezentace atd.
- Referenční příručky
  - Popisuje možnosti a chování vnitřních částí
  - <https://www.freepascal.org/docs-html/ref/ref.html>
- Digramy UML
  - Datové digramy, procesní (vývojové) diagramy

# Demontrační příklad piškvorky

```
Piskvorky
Hrac 1 - O, HRAC 2 - X
  A  B  C  D  E  F  G  H  I  J
1:
2:
3:           O
4:       X  O  O  O  O  X
5:           O  X
6:           X  X
7:
8:
9:
10:

HRAJE HRAC 2:
Zadej radek[1..10,0 KONEC]: 
```

# Piškovorky

## Prvotní sběr a formulace požadavků

- Jak se to hraje?

<https://cs.wikipedia.org/wiki/Piškvorky>

- Naše pravidla:

- Herní pole má předem danou velikost
- Hrají dva hráči
- Hraje se na pět stejných symbolů v řadě

- Naše požadavky

- Jednoduchá (krátce potřebný čas pro dokončení)
- Intuitivní (uživatel dokáže aplikaci využívat samostatně)
- Prenositelná v rámci platformy MS Windows

Prvotní sběr a  
formulace požadavků

Prvotní analýza  
a návrh

Milníky

# Piškovorky

## Prvotní analýza a návrh

- Rozhodnutí:

- Vyvojové prostředí FreePascal
- Hráč zadává tah udáním řádku a sloupce
- Hra musí provádět pouze korektní tahy
- Hra nebude kontrolovat vyhru (ponechano na hráčích)
- Procesní graf hry:



Prvotní sběr a  
formulace požadavků

Prvotní analýza  
a návrh

Milníky