

Seminář PRG

1. hodina - 6.9.2024

Gymnázium Voděradská 2024/2025
Jan Borecký



Kdo jsem?

- Absolvent Gymvodu (OC.6 2018/19)
- Bc. - FEL ČVUT, Otevřená Informatika - Počítačové hry a grafika
- Mgr. - MFF UK, Informatika - Vizuální výpočty a vývoj počítačových her
- Práce
 - Digiteq Automotive - VX Lab Developer
 - SCS Software - Houdini Developer
- Jsem student stejně jako vy, proto mi tykejte :)
- Náruživý hráč
- Skaut
- Sporty (jiu-jitsu, badminton, cokoliv)



O čem bude seminář PRG?

1. Základy algoritmizace a programování
2. **Git** a práce s verzovacím systémem
3. C#
 - a. Syntax a struktura
 - b. Datové typy
 - c. Cykly, rekurze
 - d. Pole, slovníky apod.
 - e. Soubory a práce s textem
 - f. Objektivě orientované programování**
 - g. Numerické, řadící a vyhledávací algoritmy
 - h. Formulářové aplikace .NET
4. Práce v týmu na **větším projektu**
 - Maturitní otázky 2, 4, 7, 9, 12, 14, 17, 19, 22, 24
(Podle seznamu maturitních témat 2021/2022)



Co si ze semináře (doufám) odnesete?

- Příprava na maturitu z IVT
- Budete umět napsat jednoduchý kód v C#
- Vyzkoušíte si práci v týmu
- Přičichnete si k programování a zjistíte, jestli se tomu vůbec chcete věnovat
 - Je v pořádku zjistit, že ne



Organizace semináře

Pátky 1.-2. hodina (8:00 - 9:40, pauza 8:45 - 8:55)

- Přestávky & dřívější konec
- Úkoly a známkování
 - Odevzdáváte nahráním (většinou) projektu na svůj repozitář, ke kterému mi dáte přístup
 - Úkoly jsou bodované, při zadání každého úkolu budete vědět i přepočet bodů na známku
- Veškeré materiály a zdroje najde na <https://github.com/Yeenya/Gymvod/tree/main/2024-2025/Septimy>
- Nebojte se ozvat, pokud něčemu nerozumíte nebo nevidíte na projektor / tabuli

Základy algoritmizace a programování



Proč chceme algoritmizovat?

- Potřebujeme vyřešit problém, který se většinou opakuje / něco zjistit
- Chceme univerzální řešení na daný typ problému
- Potřebujeme popsat jednotlivé kroky vedoucí k řešení



Algoritmus

- **Přesný návod**/postup, kterým lze vyřešit **daný typ úlohy**
- Vlastnosti:
 - **Konečnost** - skončí v konečném počtu kroků
 - **Determinovanost** - v každé situaci je jednoznačně určen a je dané, co se má dál dělat
 - **Hromadnost** - řeší obecný typ problému (např. násobení, tedy jak $3 * 7$, tak $5 * 9$)
- My jako programátoři po algoritmu chceme co největší efektivitu (časově i paměťově)



Základní struktury algoritmu

- **Sekvence** - posloupnost za sebou jdoucích kroků
- **Selekce** - podmínka rozvětví algoritmus a vykoná se pouze vybraná podposloupnost kroků
- **Iterace** - cyklus - opakování kroků v závislosti na splnění podmínek



Postup při tvorbě algoritmu

- 1) Specifikace - formulace požadavků, výchozích hodnot, výsledků, formu a přesnost řešení
- 2) Analýza - ověření řešitelnosti úlohy, výběr nejvhodnějšího řešení, definice restrikcí pro vstupy
- 3) Algoritmizace - sestavení jednoznačného sledu operací
- 4) Sestavení programu - sepsání algoritmu v programovacím jazyce, jeho kompilace a vytvoření spustitelného programu
 - Stejně to vypadá v praxi při komunikaci s klientem / uživatelem



C#

- Vysokoúrovňový objektově orientovaný programovací jazyk
- Autor - Microsoft
- Založený na C++ a Javě
- Využití - databázové programy, webové aplikace, formulářové/konzolové aplikace ve Windows, mobilní aplikace, Unity & v omezené míře Godot
- Úzce spojený s **.NET**




.NET (Framework)

- .NET = soubor technologií v softwaru pro podporu programování
- .NET Framework = verze .NET pro Windows
- Obsahuje velkou knihovnu tříd, mezijazykovou interoperabilitu atd.
- Programy napsané v .NET Frameworku se spouští ve **virtuálním stroji** (= virtualizovaný počítač / softwarová emulace počítače)
 - řeší bezpečnost, správu paměti, řešení výjimek a chyb



Microsoft Visual Studio

- Vývojové prostředí, tzv. **IDE** (Integrated Development Environment)
- Vícejazykové - C, C++, C#, Visual Basic atd.
- Obsahuje editor zdrojového kódu, kompilátor, interpreter a debugger
- Další příklady IDE:
 - CLion - C a C++
 - PyCharm - Python
 - Idea - Java
 - Rider - vícejazykový, postavený na .NET
 - Visual Studio Code - odlehčené Visual Studio (editor, debugger)
- Používáme MS Visual Studio Community  (**NE Visual Studio Code  !**)
(<https://visualstudio.microsoft.com/cs/free-developer-offers/>)



Funkce

- Způsob pro strukturalizaci a znovuvyužití částí algoritmu
- Mohou mít vstupní parametr(y)
- Mohou navracet hodnotu
- V reálném světě si funkce lze představit jako stroj, který za nás udělá něco, co bychom museli vždy dělat ručně



Příklad funkce - Pythagorova věta

```
float Pythagoras(a, b)
{
    float aSquared = a * a;
    float bSquared = b * b;
    float cSquared = aSquared + bSquared;
    float c = Sqrt(cSquared); // Sqrt nám dává .NET ve Visual Studiu
    return c;
}

void Main()
{
    float a = 2f; // Místo "hard-codingu" bychom mohli číst vstup od uživatele
    float b = 5f; // Stejně jako u a
    float c = Pythagoras(a, b);
    Console.WriteLine(c);
}
```



MS Visual Studio projekt

- Založíme si nový konzolový projekt a ukážeme si, co všechno v něm je a proč to tam je

Děkuji za pozornost

Zpětná vazba:

<https://forms.gle/gAjUmYwmDm5oUkEi8>

Kontakt:

Mail - honza.borecky@seznam.cz

Discord - yeenya (Yeenya#6930)