C# & .NET

http://d3s.mff.cuni.cz

CHARLES UNIVERSITY

Faculty of Mathematics and Physics

Cvičení

RNDr. Filip Krijt

krijt@d3s.mff.cuni.cz

http://d3s.mff.cuni.cz/~krijt/

Plán

- Rekapitulace předchozích úloh
  - Vaše zkušenosti?
  - „Optimální“ návrh
  - Běžné problémy
- Zadání další úlohy k samostatné práci

Kontaktní informace

- Filip Krijt
- krijt@d3s.mff.cuni.cz
- http://d3s.mff.cuni.cz/~krijt/
  - Místnost, email, Skype…
  - Informace ke cvičení

Rekapitulace předchozích úloh

- Vaše dojmy? :)

Rekapitulace předchozích úloh

- Dost řešení :)
- Úloha bude prodloužena
- Různé přístupy k návrhu
  - Vše do mainu
  - Statické metody
  - Objekty
- Různé způsoby čtení
- Různé způsoby počítání

Boj s ReCodExem

- Pozor na formátování
  - CodEx v principu jen porovnává váš výstup s oficiálním výstupem po bytech, libovolný rozdíl -> WA
  - Hodí se porovnávat s ukázkovým výstupem tam, kde to jde – hexadecimální editor
  - CodEx používá Unixové konce řádků – pro ladění se dá nastavit v Console.Out.NewLine
- Výsledek neříká vždycky co se na první pohled zdá
- Poslední test momentálně nepřesný

Objektový návrh

- Naučte se přemýšlet spíš v objektech a jejich interakcích než sekvenčním provádění kódu
- Návrh shora dolů
  - Dekompozice, rozděluji úlohu na menší a menší kusy, které postupně implementuji
  - Opačný postup jde principielně také, ale má problémy s integrací vznikajících součástí
- Kompromisy – občas jdou vlastnosti proti sobě
- Má cenu znát a řídit se nějakými best practices, nicméně jsou to doporučení, ne dogmata

Objektový návrh – heuristiky

- Hledejte kusy funkcionality, které jde seskupit dohromady pod nějaké rozhraní
- Dependency inversion principle - spoléhejte na abstrakce (rozhraní), ne konkrétní třídy a implementace
- Single responsibility principle – třída by neměla dělat / zastřešovat víc jak jednu věc
- Cílem je vždycky zajistit nějaké kvalitativní vlastnosti => Analyzovat, zda skutečně návrhem něco zajišťuji, nebo jen zbytečně komplikuji

DEMO

Nevhodné použití

- Pozor na různé typy kolekcí – ty doporučené jsou v System.Collections.**Generic**
  - Pokud používáte kolekci z jiného namespace, měl by pro to být **velmi** dobrý důvod a měli byste kolekci mít dobře prozkoumanou
- Obecně pozor na používání tříd které dobře neznáte – může tam být nějaký předpoklad o kterém nevíte
  - SortedDictionary vs. Dictionary – MSDN Library Help

Nevhodné použití

- Pozor na metody typu StreamReader.ReadToEnd
  - Pro některá použití může být validní – víme že data jsou omezená, nebo jen chceme něco vyzkoušet
  - V případě kdy o datech nemůžeme nic předpokládat nám ale hrozí OutOfMemoryException
- Stringy jsou neměnné (immutable)
  - Zápis word += c znamená vytvoření nového stringu a kopírování dosavadních prvků
  - Třída StringBuilder

Práce s výjimkami

- Pozor na obecné chytání výjimek (catch, catch(Exception))
  - Může sežrat i nějakou výjimku na kterou byste byli rádi upozorněni – např. OutOfMemoryException :)
- Výjimky mají hierarchickou stromovou strukturu
  - Tj. můžu odchytávat společného předka
  - Adoptujte i pro vlastní výjimky
- MSDN Library záznam dané třídy a metody

Prostor pro dotazy