

Výjimky...

Podobně jako v Pythonu

- objekt
- uvnitř zajímavé informace
- různé typy
- try-blok
- catch
- finally
- throw

Generické metody a třídy

- jeden a více typových parametrů
- možnost omezit pomocí where
- where:
 - ◆ interface
 - ◆ class
 - ◆ struct
 - ◆ rodičovská třída
 - ◆ ...apod.

Zvěřinec OOP jazyků

- Python (1991)
- C++ (1985)
- Java (1995)
- C# (2000)

Simula67 (1967), Smalltalk (1971),
Object Pascal (1989), Visual Basic (1991)
JavaScript (1995), PHP (1995)...a další

Zvěřinec OOP jazyků 2

Specifika, zvláštnosti, úchylky:

- C# a Java pouze objektově
- C++ pracuje přímo s pamětí
- C#, Java, PY: garbage collector
- PY nemá statické typování
- PY nemá private/protected
- PY, Java ani C++ nemají properties
- ...

Zvěřinec OOP jazyků 3

Specifika, zvláštnosti, úchylky:

- Java nemá var-parametry
- Java class = stejnojmenný soubor
- ...

Všechny jazyky se vyvíjejí a přebírají to, co se jinde osvědčilo.

Automatické testy: NUnit

- potomek Junit (Java)
- ...který je potomek Sunit (Smalltalk)
- podobně jako unittest v Pythonu
-
- testovací projekt
- Test Explorer
- [TestMethod()], Assert(...)

DOTAZY

Hygiena programování (praktické tipy)

1. Oddělovat různé činnosti

1. analýza
 2. design
 3. GUI design
 4. kódování
 5. texty, grafika, zvuky
 6. testování (psaní testů)
 7. ladění
 8. dokumentace
- resources, šablony výstupů...

2. Pracovní deník

1. ujasnění CO budu dělat teď
2. rozhovor (kačenka)
3. záznam pro soustředění na jednu věc
(viz ToDo-list)
4. pozdější dohledání (...a viz verse)
5. sledování, kolik času mi co trvá

3. ToDo-list, bug-list, issue tracker

- * //TODO ve zdrojovém kódu
- * deník, konec nebo vyhledatelná značka
- * samostatný soubor
- * software (GitHub, Trello, GitLab)

4. Správa verzí

- * CSV, SVN (Subversion),
Hg (Mercurial), GiT
- * možnost návratu k minulým verzím
- * sledování rozdílů

I při práci na jednom počítači.

I pro ne-programátorské projekty.

5. Automatické testy, TDD

* unit-testy

1. Napsat testy
2. Spustit testy a ujistit se, že selžou
3. Napsat kód
4. Ladit kód, dokud nesplní testy
5. Refaktorovat kód
6. Opakovat

* testování funkcionalit

* výkonnostní testy

* regresní testy i dnes funguje, co fungovalo včera

* Průběžná integrace (CI)

Dodatky ke grafice (blbinky)

- HDC (Handle)
- externí funkce (unmanaged code)
- Graphics - rotace
- ...apod.