

Maturitní témata 2020/21

Otázka č. 30 – Počítačové sítě a programování

Výjimky:

- exception je v programování výjimečná situace která může nastat za běhu programu, jedná se o zobecnění vnitřního přerušení vyvolaného chybou při provádění programu, ve vyšších programovacích jazycích je obvykle výjimka spojena s objektem nesoucím informace o chybovém stavu
- ve většině vyšších programovacích jazyků které podporují zpracování výjimek, je implicitně (blok - zdrojový kód do bloků) nebo explicitně (např. klíčovým slovem try) vymezeno v jaké části kódu se má hlídat vznik výjimek, výskyt výjimky se ošetřuje zvláštním blokem kódu obvykle uvozeným klíčovým slovem catch nebo except
- využití výjimek a jejich zpracování není pro všechny programovací jazyky vždy stejné,
- mezi programovací jazyky s podporou zpracování výjimek patří např. C++, C#, Java, PHP a Python

Ladění:

- metodický postup pro nalézání a snižování množství chyb v počítačových programech nebo elektronického hardware tak, aby fungoval jak se předpokládá
- ladění chyb bývá obtížnější v systémech které jsou silně provázané protože jedna chyba vyvolá další, ladění probíhá pomocí testování systému u software za pomoci ladících výpisů nebo ladících nástrojů (např. debugger)
- ladění je obecně zdoluhavý a únavný úkol, schopnost odstranit problém nejvíce závisí na zkušenosti programátora ale ladění programu se značně liší i dle použitého programovacího jazyka, dostupných nástrojů jako jsou debugery (softwarové nástroje které umožňují programátorovi sledovat průběh programu, pozastavit jej, znovu pustit, nastavit body zastavení a změnit hodnoty v paměti) apod.
- obecně platí že vyšší programovací jazyky jako je Java provádí ladění snáze protože mají takové vlastnosti jako je zpracování výjimek, atd.
- výpis ladění je pozorováním hlášení které indikují průběh provedení procesu, prvním krokem při ladění je pokus o reprodukci (zopakování) problému, v okamžiku kdy se podaří problém zopakovat je třeba zjednodušit vstup do tohoto programu aby se snáze provádělo ladění
- programátor odstraňuje některé části z původního programu a kontroluje zda se problém stále ještě vyskytuje
- vzdálené ladění (remote debugging) je způsob odstraňování chyb kdy laděný program běží na jiném systému než kde je umístěn debugger

- Anti-debugging je zavedením jedné nebo více technik do počítačového kódu které brání pokusům o ladění finálního programu, existuje několik typů technik např.:

- API: ověří přítomnost debuggeru pomocí standardních systémových funkcí
- Výjimky: kontrola modifikace obsluhy výjimek
- Pozměněný kód
- Časování a latence - zkontroluje čas potřebný pro provedení instrukcí

- ladění může být omezeno pomocí jedné nebo více daných technik, k dispozici je mnoho anti-debugging technik které dokáží dostatečně chránit software proti většině hrozeb

Druhy a ošetření chyb try, except, chybová událost:

- chyby rozdělujeme na

- Syntaktické: chyba v zápise, snáze se odhalují
- Sémantické: program dělá něco jiného než jsme zamýšleli, špatně se hledají

- některé druhy chyb:

- algoritmická chyba
- chyba syntaxe
- chybná data
- chyba kapacity nebo meze
- časová nebo součinnostní chyba
- chyba zotavení
- chyba nedodržování standardů a procedur

- ošetření chyb se může dělit na:

- Aktivní ošetření chyb - v programu zmapujeme všechna zranitelná místa a ošetříme je podmínkami
- Pasivní ošetření chyb - zejména když je operace složitější a bylo by příliš náročné ošetřovat všechny možné chybové stavy, nastupují výjimky, tzv. pasivní ošetření chyb, pokusíme se nebezpečnou část kódu spustit v „chráněném režimu“, tento režim je nepatrně pomalejší a liší se tím že pokud dojde k chybě máme možnost ji odchytit a zabránit pádu programu, o chybě zde hovoříme jako o výjimce, využíváme k tomu tzv. try-catch bloky
 - do bloku try umístíme nebezpečnou část kódu, pokud nastane v bloku try chyba jeho vykonávání se přeruší a program přejde do bloku catch, pokud vše proběhne v pořádku try se vykoná celý a catch se přeskočí

```

try
{
    Console.WriteLine(Matematika.Podil(a, b));
}
catch
{
    Console.WriteLine("Při dělení nastala chyba.");
}

```

- v pythonu se používá místo catch příkaz except, except se může používat vícekrát

```

try:
    # Zde se umístí hlavní část kódu programu.
except TypVyjimky:
    # Zde se bude zpracovávat jmenovaná výjimka.
except JinyTypVyjimky:
    # Zde se ošetřuje jiná výjimka.
else:
    # Zde umístíme úklidový kód, který se provede
    # v případě, že nenastane žádná výjimka.

```