Programování

Ladění, debugging

Řešení chyb v kódu v průběhu implementace

- **Syntaktická chyba**
  - Viditelná přímo v IDE (červené podtržení)
  - Zabraňuje kompilaci a sestavení projektu
  - IDE se nám snaží poradit, jak daný problém vyřešit v přehledu chyb
- **Běhová chyba** (nepohlídání podmínek, vstupů, …)
  - Odhalení pomocí **try-catch**
  - Nebrání v překladu kódu
  - Často způsobuje pád programu, vytvoření výjimky v běhu programu
- **Logická chyba**
  - Velmi špatně odhalitelná
  - Debugging

Syntaktická chyba

- Chyb, které odhalí samotné IDE a znemožňuje spuštění
- Oprava takové chyby závisí od jejího typu
- Zpravidla se jedná o chybějící nebo přebývající znaky
- Syntaktická chyba vzniká i voláním proměnné nebo funkce která neexistuje
- Chyby, které jsou vypsány v příslušné sekci řešte odshora
  - Je dost možné, že drobná chyba je navázána na další chybová hlášení
- V případě, že nerozumíte, co za chyby vám IDE hlásí – zkopírujte chybovou hlášku a využijte Google (určitě nejste první s daným problémem)

Běhová chyba

- Středně závažná chyba, která se projeví až po spuštění a používání programu
- Typickým příkladem běhové chyby například nemožnost přistoupit na volaný soubor, nesprávný formát vstupů nebo špatná práce s pamětí
- Pro tyto typy chyb využíváme pojem **Exceptions** (výjimky v běhu programu)
- Vznik výjimky můžeme vynutit pomocí klíčového slova **throw**
  - *throw new Exception("zpráva nové výjimky");*
- Výjimky odchytáváme pomocí konstrukce **try-catch**

Ukázka použití konstrukce try-catch

- Do bloku, kterému předchází klíčové **try** vkládáme kód, který může vytvořit výjimku v běhu programu
- Pomocí **catch** můžeme odchytávat buď specifické typy chyb nebo obecně všechny výjimky
- Blok **finally** je vykonán bez ohledu na výskyt výjimky

```
int errors = 0;
try
{
    // krizová část kódu

}
catch (InvalidTimeZoneException e1)
{
    Console.WriteLine("Zjištěna chyba: {0}", e1.Message);
    errors++;
}
catch (FormatException e2)
{
    Console.WriteLine("Zjištěna chyba: {0}", e2.Message);
    errors++;
}
finally
{
    Console.WriteLine("Vyskytlo se {0} vyjímek v krizové části", errors);
}
```

Logická chyba

- Vytvořený program lze spustit, nevzniká žádná výjimka při běhu
- Program ovšem nedělá to, co od něj očekáváme
- Neexistuje obecný postup jak takovou chybu opravit
- Pro identifikaci chyby využíváme **debuggingu** (krokování)
- Umístění tzv. breakpointu na řádek kódu, od kterého sledujeme chování programu
  - Sledujeme hodnoty proměnných a kontrolujeme vůči navrženému algoritmu
- Chyba ovšem může vzniknout již při návrhu algoritmu
  - Proto nepodceňujme návrh algoritmu například pomocí vývojového diagramu

Ukázka debuggingu

```
53          }
            – odkazy
54          public void VypisVozidel()
55          {
56              string parkovist = "";
57              for (int i = 0; i < kapacita; i++)
58              {
59                  if (zaparkovano[i] != null)
60                  {
61                      parkovist += zaparkovano[i].ToString() + "\n";
62                  }
63              }
64              Console.WriteLine(parkovist);
65          }
            – odkazy
```