

# Řízení toku kódu

## 3. cvičení

Jiří Zacpal

KMI/ZP3CS – Základy programování 3 (C#)

# Blok (složený příkaz)

- umožňuje zapsat více příkazů na místě jednoho
- syntaxe:

```
{  
    deklarace_1;  
    ...  
    deklarace_M;  
  
    příkaz_1;  
    ...  
    příkaz_N;  
}
```

# Větvení programu

- konstrukce **if**
  - umožní rozvětvení programu; příkazy se provedou, pokud je splněna daná podmínka
  - pomocí klíčového slova `else` lze definovat větev, která se provede, pokud podmínka splněna nebude
  - příkaz `if` lze vnořit do jiného `if`
- konstrukce **switch**
  - podle hodnoty celočíselného výrazu umožní rozvětvení programu do libovolného množství větví
  - nelze vytvořit větev pro interval hodnot
  - pomocí klíčového slova `default` lze vytvořit větev, jejíž příkazy se budou provádět, pokud nebyla vybrána jiná větev

# Konstrukce IF

- syntaxe jednoduchého if:

```
if ( podmínka ) příkaz
```

- příklad:

```
if (x==0) printf("Proměnná x je  
nula.\n");
```

- syntaxe s využití větve else:

```
if ( podmínka ) příkaz_1 else  
příkaz_2
```

- příklad:

```
if (x>0) printf("x je kladné\n");  
else printf("x není kladné\n");
```

# Líné vyhodnocování

- V logických výrazech se vyhodnocuje pouze část výrazu (podle asociativity) nutná pro získání výsledku.

- příklady:

```
if ( (c>0) && (c<(i=m%n)) )  
{ ... }
```

```
if ( (c<0) || (c>(i=m%n)) )  
{ ... }
```

# Příklad 1

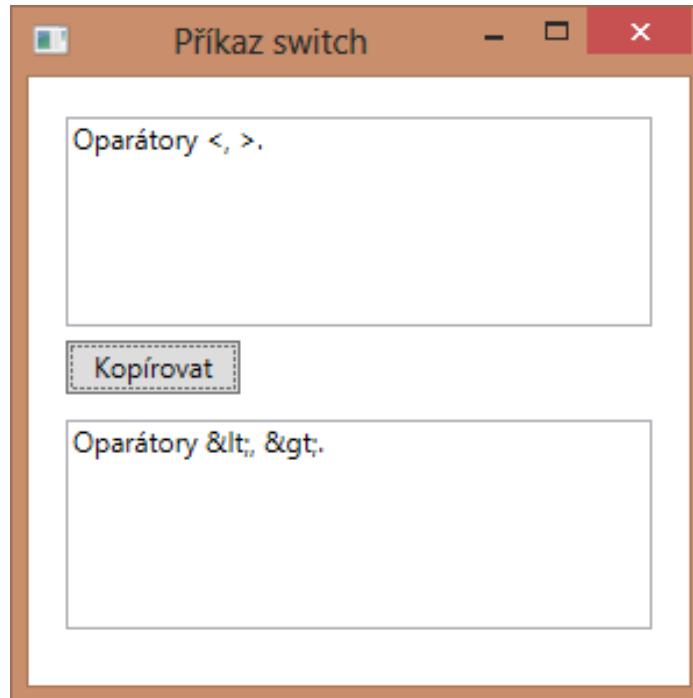
The screenshot shows a Windows application window titled "Data". Inside the window, there are two date input fields. The first field, labeled "První", contains the date "18. 9. 2013" and has a calendar icon with the number "15" next to it. The second field, labeled "Druhý", also contains the date "18. 9. 2013" and has a similar calendar icon with the number "15". Below these fields is a button labeled "Porovnat". Underneath the button is a text area displaying the results of the comparison: "První == Druhý : True", "První != Druhý : False", "První < Druhý : False", "První <= Druhý : True", "První > Druhý : False", and "První >= Druhý : True". At the bottom right of the window is a button labeled "Konec".

# Konstrukce SWITCH

- syntaxe:

```
switch (celočíselný_výraz) {  
    case konstanta_1: příkazy_1  
  
    break;  
    ...  
    case konstanta_N: příkazy_N  
  
    break;  
    default: příkazy_default_větve  
}
```

# Příklad 3





# Konstrukce cyklů

- konstrukce **while**
  - cyklus s podmínkou testovanou na začátku cyklu
  - nemusí proběhnout vůbec (nesplněná podmínka)
- konstrukce **do - while**
  - cyklus s podmínkou testovanou na konci
  - proběhne alespoň jednou
- konstrukce **for**
  - nejčastěji používán jako cyklus s pevným počtem opakování

# Cyklus while

- syntaxe:

```
while (podmínka) příkaz
```

# Cyklus do - while

- syntaxe:  
do příkaz while (podmínka);
- provede se alespoň 1x

# Cyklus for

- syntaxe:  
for (start; podmínka; iterace)  
příkaz

# Příkazy přerušení cyklu

- příkaz `continue`
  - skok na konec nejvnitřnějšího cyklu, výpočet pokračuje další iterací (včetně testu případné podmínky)
- příkaz `break`
  - okamžité opuštění nejvnitřnějšího cyklu

# Ošetřování chyb a výjimek

- příkazy try a catch

```
try
```

```
{
```

```
    //zde se může vyskytnout chyba
```

```
}
```

```
catch(typchyby identifikator)
```

```
{
```

```
    //zde bude zpracována výjimka
```

```
}
```

# Kontrola přetečení

- použití klíčových slov checked a unchecked
- příklady:

```
int c=int.MaxValue;  
checked  
{  
    int v=c++;  
}  
int nv=unchecked(c++);
```

# Vyvolávání výjimek

- pomocí příkazu throw

```
throw new TridaChyby(„Hlášení“);
```



# Zachycení neošetřené výjimky

- pomocí catch

```
catch(Exception ex)
{
    //zde bude zpracována výjimka
}
```

# Blok finally

- uvádí se za bloky try a catch
- tento blok bude proveden vždy
  - když není vyvolána výjimka
  - když je vyvolána libovolná výjimka

**finally**

```
{  
    ...  
}
```

# Příklad 4

The screenshot shows a Windows application window titled "Exceptions". Inside the window, there are two input fields for operands: "levý operand" (left operand) containing the text "Ahoj" and "pravý operand" (right operand) containing the number "8". Between these fields is a vertical list of five radio buttons for mathematical operations: "+ Sčítání", "- Odčítání", "\* Násobení" (which is selected), "/ Dělení", and "% Zbytek". Below the radio buttons is a button labeled "Vypočti" (Calculate). Underneath this button is a label "Výraz" (Expression) followed by an empty text box. Below that is a label "Výsledek" (Result) followed by a text box containing the error message "Vstupní řetězec nemá správný formát." (Input string has an incorrect format). At the bottom left of the window is a button labeled "Konec" (End).

# Úkol

- Vytvořte grafickou aplikaci, která pro zadané číslo vypíše všechna prvočísla menší než toto číslo. V aplikaci ošetřete špatný vstupní formát a přetečení vstupu.

