Větvení programu

Jiří Zacpal

KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

KMI/ZPP1 Základy programování v Pythonu 1

Řešení z minulé hodiny

```
a=3
b=2
c=4

vysledek="platí" if a**2+b**2==c**2 else "neplatí"
print (vysledek)

vysledek="je v intervalu" if b<a<c else "není v intervalu"
print (vysledek)

vysledek="je možno sestrojit" if a+b>c and a+c>b and b+c>a else "není možno
sestrojit"
print (vysledek)
```

Podmíněný příkaz

Motivační příklad

- Máme rozhodnout, zda číslo a je menší než číslo b:

```
a = 1
b = 2
result = True if a < b else False
print(result)
```

- Pokud je a<b, pak k proměnné m přičteme a+b, jinak a-b:

```
a = 1
b = 2
m=0
m += a+b if a < b else a-b
print(m)
```

Motivační příklad

- Pokud je a<b, pak k proměnné m přičteme a+b a a nastavíme na 0, jinak a-b a b nastavíme na 0:

```
a = 1
b = 2
m=0
m += a+b if a < b else a-b
a,b= 0,b if a < b else a,0
print(m)
```

- Nepřehledné.
- Musíme 2x vyhodnocovat operátor.
- Co když budeme chtít vykonat víc příkazů?

Fyzické a logické řádky

- Fyzický řádek
  - je posloupnost znaků ukončená znakem konce řádku,
  - jeden řádek v editoru je jeden fyzický řádek.
- Logický řádek
  - je řádek, jak jej chápe překladač a interpret,
  - většinou je shodný s fyzickým řádkem, ale ve speciálních případech sestávat i z několika fyzických řádků:
    - Končí-li fyzický řádek znakem \, sloučí se s následujícím fyzickým řádkem do jednoho logického řádku. Takto sloučit i několik po sobě následujících řádků.
    - Otevřete-li na řádku závorku, kterou na tomto řádku nezavřete. Sloučí se všechny fyzické řádky až po řádek s odpovídající uzavírací závorkou do jednoho logického řádku.

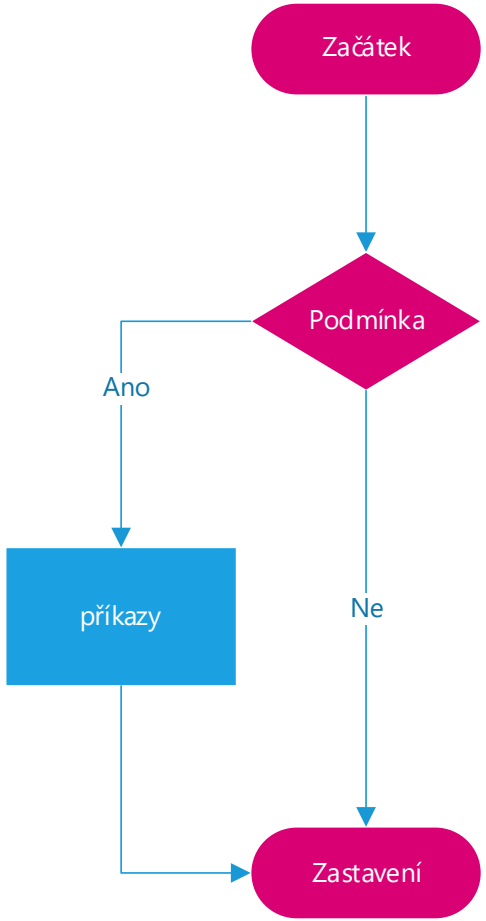
Jednoduchý a složený příkaz

- Python děli příkazy na jednoduché a složené.
- Jednoduché příkazy (anglicky simple stalements)
  - se dají vždy zapsat na jeden logický řádek,
  - Python umožňuje umístění několika příkazů na řádek. V takovém případě se musí jednotlivé příkazy oddělovat středníky.
- Složené příkazy (anglicky compound statements)
  - sestávají z hlavičky ukončené dvojtečkou a z těla, které je tvořeno posloupnosti příkazů označovanou často jako blok příkazů nebo zkráceně pouze blok,
  - hlavička musí začínat vždy na novém řádku a tělo musí vždy obsahovat nejméně jeden příkaz,
  - obsahuje-li tělo pouze jednoduché příkazy, mohou následovat za hlavičkou (to se však používá spíše výjimečné),
  - standardně se příkazy těla píší na další řádky a všechny se shodně odsazuji oproti hlavičce. Doporučovaná velikost odsazeni je 4 znaky, ale teoreticky muže být libovolná — definuje ji první příkaz těla.

Jednoduchý podmíněný příkaz

- umožní rozvětvení programu; příkazy se provedou, pokud je splněna daná podmínka,
- syntaxe konstrukce `if`:

```
if podmínka:
    příkazy
```



Příklad

- Napište program:

```
a = 5
if a < 10:
    print(a)
```

- Program pro výpočet absolutní hodnoty:

```
x = -5
if x < 0:
    x = -x
print(x)
```

Příklad

- Program pro uložení menší hodnoty do proměnné a:

```
a = 4
b = 2
if b < a:
    c = a
    a = b
    b = c
print(a)
print(b)
```

- Větvení je příkaz, proto můžeme větvení do sebe vnořovat:

```
a = 4
if a % 2 == 0:
    if a == 4:
        print(1)
print(2)
```

Klauzule větvení

Příklad

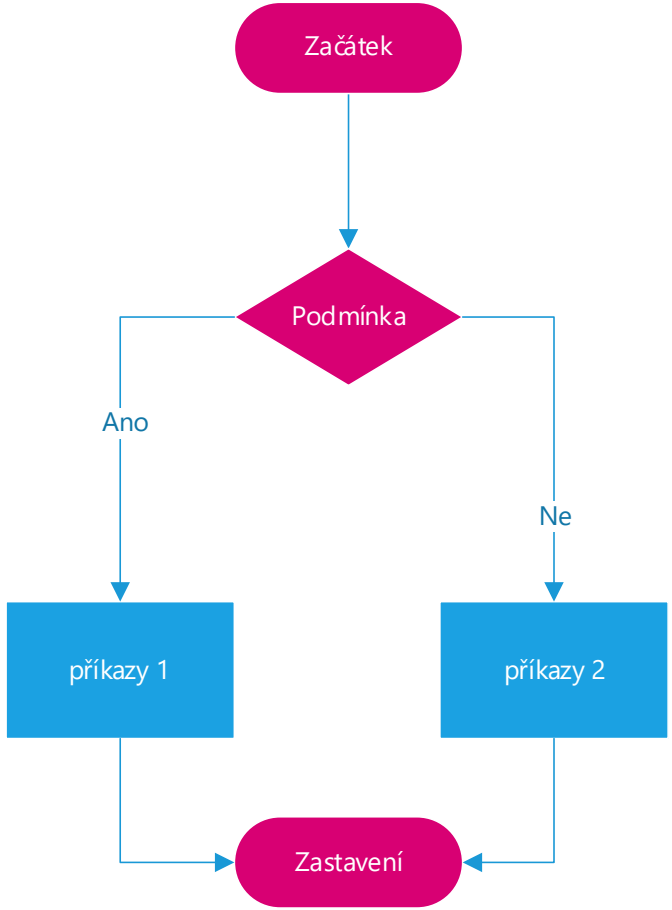
- Napište si následující program:

```
x = 10
if x > 20:
    y = 1
if x <= 20:
    y = 2
print(y)
```

Úplný podmíněný příkaz

- každá klauzule má hlavičku a tělo,
  - hlavička začíná klíčovým slovem a konči dvojtečkou
  - tělo je odsazený blok příkazu.
- syntaxe s využití větve `else`:

```
if podmínka:
    příkazy 1
else:
    příkazy 2
```



Příklad

- Přepišme minulý příklad pomocí klauzulí větvení:

```
x = 10
if x > 20:
    y = 1
else:
    y = 2
print(y)
```

Příklad

- Program pro funkci signum:

```
n = 10
if n > 0:
    signum = 1
if n < 0:
    signum = -1
if n == 0:
    signum = 0
print(signum)
```

Příklad

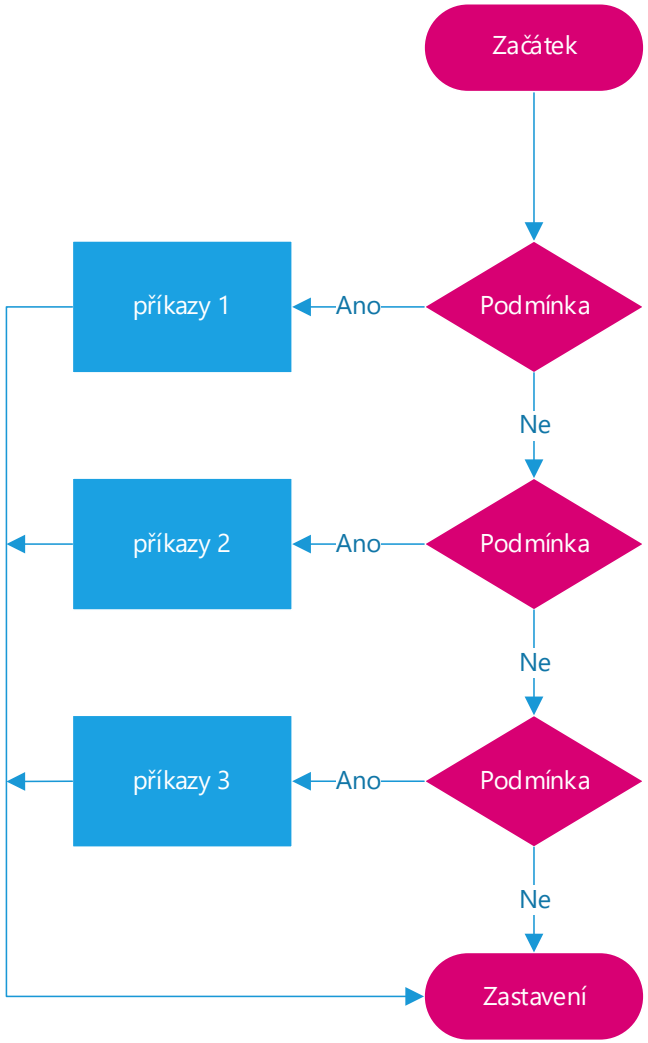
- Program pro funkci signum pomocí else:

```
n = 10
if n > 0:
    signum = 1
else:
    if n < 0:
        signum = -1
    else:
        signum = 0
print(signum)
```

Klauzule elif

- každá klauzule má hlavičku a tělo,
  - hlavička začíná klíčovým slovem a konči dvojtečkou
  - tělo je odsazený blok příkazu.
- syntaxe s využití větve `else`:

```
if podmínka 1:
   příkazy 1
elif podmínka 2:
   příkazy 2
elif podmínka 3:
   příkazy 3
```



Příklad

- Program pro funkci signum pomocí else:

```
n = 10
if n > 0:
    signum = 1
elif n < 0:
    signum = -1
else:
    signum = 0
print(signum)
```

- Pozor na splněnou podmínku v elif:

```
n = -2
if n < 0:
    print(1)
elif n < 10:
    print(2)
```

Příklad

```
cislo = 6

if(cislo % 2 == 0):
    if (cislo % 3 == 0):
        print("Cislo je delitelne 2 i 3.")
    else:
        print("Cislo je delitelne 2.")
else:
    if (cislo % 3 == 0):
        print("Cislo je delitelne 3.")
    else:
        print("Cislo neni delitelne 2 ani 3.")
```

Příklad

```
cislo = 6

if(cislo % 2 == 0) and (cislo % 3 == 0):
    print("Cislo je delitelne 2 i 3.")
elif (cislo % 2 == 0) and (cislo % 3 != 0) :
    print("Cislo je delitelne 2.")
elif (cislo % 2 != 0) and (cislo % 3 == 0):
    print("Cislo je delitelne 3.")
else:
    print("Cislo neni delitelne 2 ani 3.")
```

Přepínač

Přepínač match ... case

- přepínač je složený příkaz, jehož hlavička je uvedena klíčovým slovem `match` následovaná `výrazem`, podle něhož se bude program rozhodovat,
- jednotlivá možná pokračování jsou definovaná jako složené příkazy s hlavičkou uvozenou klíčovým slovem `case` za nímž následuje `vzor`, který může být hodnotou nebo speciálním výrazem

```
match výraz:
    case vzor 1: příkazy 1
    case vzor 2: příkazy 2
    case vzor 3: příkazy 3
```

Příklad

- Napíšeme přepínač pro hod kostkou:

```
hod=2
match hod:
    case 1: print("Padla jednička.")
    case 2: print("Padla dvojka.")
    case 3: print("Padla trojka.")
    case 4: print("Padla čtyřka.")
    case 5: print("Padla pětka.")
    case 6: print("Padla šestka.")
```

Příklad

- Chceme detekovat, zda padlo sudé či liché číslo:

```
hod=2
match hod:
    case 1: print("Padlo liché číslo.")
    case 2: print("Padlo sudé číslo.")
    case 3: print("Padlo liché číslo.")
    case 4: print("Padlo sudé číslo.")
    case 5: print("Padlo liché číslo.")
    case 6: print("Padlo sudé číslo.")
```

Příklad

- V tomto případě je ale lepší použít jen dvě části case:

```
hod=2
match hod:
    case 1|3|5: print("Padlo liché číslo.")
    case 2|4|6: print("Padlo sudé číslo.")
```

- Můžeme také doplnit case s _, která se vykoná, pokud výraz neodpovídá žádnému vzoru:

```
hod=8
match hod:
    case 1|3|5: print("Padlo liché číslo.")
    case 2|4|6: print("Padlo sudé číslo.")
    case _: print("Toto není na šestistěnné kostce.")
```

Úkol

- Napište program, který pro zadané číslo dne vypíše, o jaký den v týdnu se jedná a o jaký měsíc v roce 2023.

- Příklad:

```
Zadej číslo dne:123
Den v týdnu:
středa

Měsíc:
květen
```