

Programování

Cvičení podmínek a cyklů

There are 10 types of people in the world:

Those who understand binary and those who don't.

Martin Uza

Programování v praxi

• Jak na to?

- Lze o tom přemýšlet i mluvit, je toho plný internet, takže je možné si mnohé přečíst, nebo se podívat, jak to dělají jiní, což může být zábavné i poučné, ale nikdo se to nenaučí, dokud to nezačne dělat.
- Za pár hodin se to naučit nelze.
- Teoretické znalosti rozhodně nejsou k zahození, existují knihy i webové stránky, na kterých se můžete dozvědět o technikách, které jste doposud neznali, ale dokud je nevyzkoušíte, nepoznáte, zda Vám vyhovují.
- Ač je jazyk jen prostředkem k dosažení cíle, je třeba ho dobře ovládat, čehož lze dosáhnout pouze tréninkem.
 - Řeč je v tomto případě pochopitelně o jazyce Pascal.

Programování je extrémně komplexní dovednost



- Je jen málo lidských činností a dovedností, jejichž pokročilejší aspekty jsou propojené a závislé na základech tolik, jak je tomu u programování.
- **Bez naprosto hladkého a samozřejmého zvládnutí základů nemá vůbec cenu dělat cokoliv dalšího.**
- Žádný smysluplný program, který má vykonávat nějakou netriviální činnost, není možné napsat bez použití větvení a cyklů/skoků.
- Větvení a cykly jsou v každém rozumném jazyce, takže vše, co se naučíte v Pascalu, využijete i jinde.
- Cykly i podmínky patří do „dovednosti programovat“, nikoliv do ovládnutí programovacích jazyků.

Cvičení, findMax

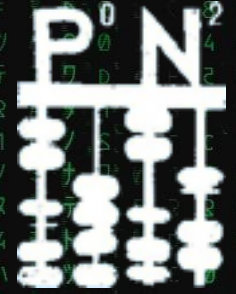
- Napište program findMax, kterému uživatel zadá osm kladných čísel a program vypíše nejvyšší z nich.
- Upravte program tak, aby uživatel mohl při každém běhu zadat jiný počet kladných čísel, který si určí předem. Program se napřed zeptá na počet čísel, která chce uživatel zadat, pak jich právě tolik přečte.
- Modifikujte program, aby mohl uživatel zadat libovolný počet kladných čísel, přičemž zadávání ukončí tím, že zadá záporné číslo nebo nulu.
- Změňte program tak, aby místo nejvyššího čísla vracel druhé nejvyšší.
- *Hint: Vzpomeňte na readln, write a operátory <, >.*

Cvičení, divisibility



- Definice: Číslo a je dělitelem čísla b (tzn. „ a dělí b “), pokud lze číslo b vydělit číslem a beze zbytku.
- Napište program, který přečte ze vstupu celé číslo a vypíše na výstup, jestli je toto číslo sudé, nebo liché.
- Modifikujte program, aby přečetl dvě čísla a a rozhodl, které z nich dělí to druhé, případně jestli se dělí navzájem, nebo mezi nimi žádná dělitelnost není.
- Vytvořte program, který přečte ze vstupu celé číslo a vypíše na výstup všechny jeho dělitele.
- Upravte program tak, aby ze vstupu přečetl dvě čísla a a našel jejich největšího společného dělitele.
- *Hint: Zjistěte si, co dělají operátory div a mod .*

Cvičení, prime Numbers



- Definice: Prvočísla jsou dělitelná právě dvěma různými čísly (sama sebou a číslem 1), jinými ne.
- Napište program, který přečte číslo ze vstupu a vypíše ANO, jedná-li se o prvočísla, jinak vypíše NE.
- Upravte program, aby přijal dvě čísla a vypsál, kolik mezi nimi (v otevřeném intervalu) existuje prvočísel.
- Modifikujte program tak, aby přečetl číslo a rozložil jej na prvočísla, jejichž součinem je původní číslo.
- Změňte program, aby přečetl dvě čísla, rozložil je na prvočísla a z nich vypsál ta, která jsou pro obě čísla společná (36 a 60 mají společná prvočísla 2, 2, 3).
- *Hint: Budou se hodit pomocné proměnné typu boolean.*

Cvičení, digits In Numbers

- Napište program, který přijme jedno číslo a vrátí ANO, pokud je číslo čtyřciferné, jinak vrátí NE.
- Změňte program, aby pro každé číslo vrátil počet jeho cifer (číslo 2 má jednu cifru, číslo 128 má tři).
- Upravte program tak, aby místo počtu cifer vrátil ciferný součet čísla (ciferný součet čísla 512 je osm, ciferný součet čísla 256 je třináct).
- Modifikujte program, aby místo ciferného součtu vrátil číslo, které vzniká přečtením původního čísla pozpátku (z čísla 1024 se stane 4201, z čísla 32 se stane 23).
- *Hint: $2048 = 2 * 1000 + 0 * 100 + 4 * 10 + 8 * 1$*

Rekapitulace

- Bez zvládnutí prvního příkladu z každého tématu (tedy z témat findMax, divisibility, primeNumbers a digitsInNumbers) nemá smysl pokračovat dál.
- Bez zvládnutí (a samozřejmě plného pochopení) druhého příkladu z každého tématu, bude porozumění další látce velmi problematické.
- Bezchybné zpracování třetího příkladu z každého tématu je slušným základem pro další přednášky.
- Nalezení řešení čtvrtých příkladů z každého tématu znamená plné pochopení předchozích přednášek.
- **I v případě, že na řešení některých příkladů sami nepřijdete, měli byste je alespoň chápat!!**

