

# 01. Úvod do algoritmizace

---

Pro začátek si řekněme, co je vlastně co:

**Algoritmus** je souhrn přesných a jednoznačných instrukcí, které vedou od vstupních hodnot k získání požadovaného řešení. *Prostě popis co, kdy, kde a jak se má udělat.*

**Algoritmizace** je proces vytváření a sestavování algoritmů.

**Programovací jazyk** je umělý jazyk, který slouží jako komunikační nástroj mezi programátorem a počítačem.

**Počítačový program** je algoritmus napsaný v programovacím jazyce. Algoritmus je obecným řešením problému, program je naprosto konkrétním řešením daného problému.

Zní to složitě? Bude hůř...

## HISTORIE ALGORITMIZACE

Spojovat algoritmickou matematiku jen s moderními technologiemi by byla chyba. Algoritmus je jakýkoliv souhrn instrukcí (slova přesných a jednoznačných budeme vynechávat, ale rozhodně si je zapamatujte, jsou velmi důležitá), která vedou k řešení jakéhokoliv problému. Takže například Eukleides svůj algoritmus pro výpočet největšího společného dělitele dvou přirozených čísel vymyslel někdy ve 4. století př.n.l..

## VLASTNOSTI ALGORITMU

- **správnost:** algoritmus, který vrací špatný výsledek, je k ničemu
- **jednoznačnost:** pokud instrukce není jednoznačná, algoritmus skončí chybou, (instrukce: Nalijte do hrnce vodu. *Teplou vodu? Studenou? Kolik vody? atd.* )
- **rezultativnost:** algoritmus, který nevrátí žádný výsledek, je k ničemu
- **konečnost:** algoritmus, který nikdy neskončí, je k ničemu
- **univerzálnost:** pokud budeme mít algoritmus vaření pytlíkového čaje, musí tomu algoritmu být jedno, jestli vaříme čaj s příchutí jahod nebo černý čaj

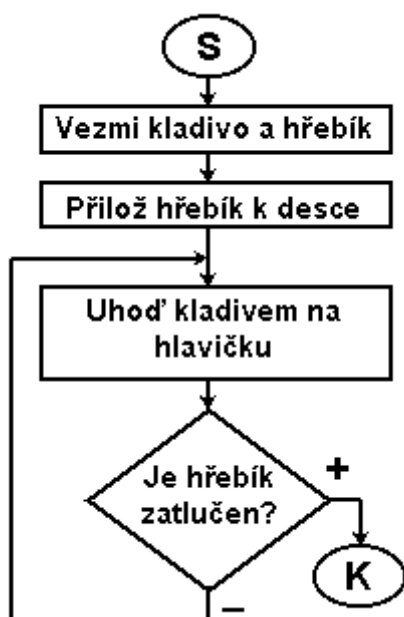
## VYJÁDŘENÍ ALGORITMU

- **slovně:** instrukce píšeme ve větách či heslech (kuchařka)
- **graficky:** instrukce jsou vyjádřeny pomocí *vývojových diagramů*
- **matematicky:** instrukce jsou matematické rovnice a vztahy mezi veličinami
- **programem:** instrukce jsou zapsány v programovacím jazyce

## Slovně vyjádřený algoritmus zatlučení hřebíku:

1. Vezmi kladivo a hřebík
2. Přilož hřebík k desce
3. Uhoď kladivem na hlavičku
4. Je hřebík zatlučen?  
ANO – pokračuj bodem 5  
NE – vrať se na bod 3
5. Ukonči činnost a odlož kladivo

## To samé vyjádřeno graficky:



Vývojové diagramy a jejich použití si rozebereme v další přednášce.

### **EFEKTIVNOST ALGORITMU**

Pro jeden problém může existovat více algoritmů, které vedou ke správnému řešení. Vaší úkolou je vždy vybrat ten nejefektivnější algoritmus, který nejvíce vyhovuje základním prioritám:

**čas:** čím kratší čas je nutný k získání řešení, tím lépe

**paměť:** čím méně paměti PC spotřebujete, tím lépe

**přehlednost:** váš algoritmus může být geniální, ale pokud se v něm nikdo nevyzná, je v praxi nepoužitelný

Příklad (hypotetický!): Sedíte v učebně TeB. Další hodinu máte algoritmizaci ve VT1, na kterou jste měli přinést domácí úkol. Domácí úkol máte dole ve skříňce. Potřebovali byste se ještě stavít v kabinetu ekonomiky, zeptat se na známku.

K vyřešení tohoto problému máte několik možností. Aby byl algoritmus správný, musíte si zjistit známku z ekonomiky a se zvoněním sedět ve VT1 s domácím úkolem před vámi.

Řešení 1.: Po cestě z TeBu se stavíte v kabinetu ekonomiky, pak si ve VT1 necháte věci, seběhnete dolů do skříňky a vrátíte se do VT1.

Řešení 2.: Seběhnete dolů do VT1, kde si odložíte věci, vyběhnete nahoru do kabinetu ekonomiky, pak seběhnete dolů do skříňky a vrátíte se do VT1.

Řešení 3.: Poběžíte až dolů do skříňky, vyběhnete nahoru do kabinetu ekonomiky a seběhnete do VT1.

Můžete najít x dalších řešení, včetně toho, že pojedete výtahem...

**Pokud vás algoritmus dovede ke správnému řešení, je správný. Nemusíte to ale nutně znamenat, že je efektivní a tudíž použitelný.**