

# Algoritmus, ADT

**Metodický koncept k efektivní podpoře klíčových odborných kompetencí s využitím cizího jazyka ATCZ62 - CLIL jako výuková strategie na vysoké škole**

**Interreg**   
EVROPSKÁ UNIE  
**Rakousko-Česká republika**  
Evropský fond pro regionální rozvoj



**Europäische Union**  
**Evropská unie**  
Europäischer Fonds für  
regionale Entwicklung  
Evropský fond pro  
regionální rozvoj



**UNIVERSITY**  
**OF APPLIED SCIENCES**  
**UPPER AUSTRIA**

# Algoritmus

- přesný návod či postup, kterým lze vyřešit daný typ úlohy
- teoretický princip řešení problému (oproti přesnému zápisu v konkrétním programovacím jazyce).
- Vlastnosti
  - Konečnost (finitnost)
  - Obecnost
  - Determinovanost
  - Výstup (resultativnost)
  - Elementárnost

# Algoritmus

- Metody návrhu
  - Shora dolů – postup řešení rozkládáme na jednodušší operace, až dospějeme k elementárním krokům
  - Zdola nahoru – z elementárních kroků vytváříme prostředky, které nakonec umožní zvládnout požadovaný problém
  - Kombinace obou – postup shora dolů doplníme "částečným krokem" zdola nahoru (použití knihovny funkcí, vyšší programovací jazyk nebo systém pro vytváření programů...)

# Algoritmus

- Metody návrhu
  - Rozděl a panuj – dělí problém na dílčí úlohy (musí být nezávislé), které pak řeší, často implementován rekurzivně nebo iteračně
  - Hladový algoritmus – řešení optimalizačních úloh, vybírá vždy lokální minimum, ve snaze najít globální minimum
  - Dynamické programování - dělí problém na dílčí úlohy (mohou být závislé), které pak řeší
  - Hledání s návratem (backtracking) – způsob řešení algoritmických problémů založený na prohledávání stavového stromu problému, vylepšení hledání řešení hrubou silou, založen na prohledávání do hloubky možných řešení

# Algoritmus

- Druhy algoritmů
  - Rekurzivní algoritmy – využívají (volají) samy sebe.
  - Pravděpodobnostní (probabilistické) algoritmy – provádějí některá rozhodnutí náhodně či pseudonáhodně.
  - Paralelní algoritmy – rozdělení úlohy mezi více počítačů
  - Genetické algoritmy – pracují na základě napodobování biologických evolučních procesů
  - Heuristický algoritmus – snaží se nalézt pouze nějaké vhodné přiblížení; používá se v situacích, kdy dostupné zdroje (např. čas) nepostačují na využití exaktních algoritmů (nebo pokud nejsou žádné vhodné exaktní algoritmy vůbec známy).

# ADT – Abstraktní datový typ

- Typy dat, které jsou nezávislé na vlastní implementaci
- Cíl – zjednodušit a zpřehlednit program, který provádí operace s daným datovým typem
- Všechny ADT lze realizovat pomocí základních algoritmických operací (přiřazení, sčítání, násobení, podmíněný skok,...)

# ADT

- Vlastnosti
  - Všeobecnost implementace – jednou navržený ADT může být zabudován a bez problémů používán v jakémkoliv programu.
  - Přesný popis – propojení mezi implementací a rozhraním musí být jednoznačné a úplné.
  - Jednoduchost – uživatel se nemusí starat o vnitřní realizaci a správu ADT v paměti.
  - Zapouzdření – rozhraní jako uzavřená část, uživatel ví, co ADT dělá, ale ne jak to dělá
  - Integrita – uživatel nemůže zasahovat do vnitřní struktury dat
  - Modularita – „stavebnicový“ princip programování
- Pokud je ADT programován objektově, jsou většinou tyto vlastnosti splněny.

# ADT

- Typy operací
  - Konstruktor – vytváří novou hodnotu ADT, sestavení platné vnitřní reprezentace hodnoty na základě dodaných parametrů
  - Selektor – slouží k získání hodnot, které tvoří složky nebo vlastnosti konkrétní hodnoty abstraktního datového typu
  - Modifikátor – provádí změnu hodnoty datového typu