

Abstraktní datové typy

Z FAV wiki

ADT je matematický model dat a operace nad těmito daty. Využívá základních datových typů pro konstrukci komplexnějších ADT, které využíváme pro snadnější organizaci dat a často pro optimalizaci algoritmů.

Abstraktní datový typ

- je zcela oddělený blok, s klientským programem komunikuje pomocí rozhraní
- je nezávislý na implementaci, na venek se projevuje jen veřejnými metodami
- vše ostatní, kromě ovládacích metod, je uživateli skryto (samotná data a přístup k nim např.) (integrita)
- je (klientsky) nezávislým blokem, může jej využívat kdokoli, kdo zná rozhraní (pokud je ADT tedy třeba v DLL knihovně, můžeme jej používat i z jiných jazyků) (modularita)
- je pevně a jasně definován

ADT definuje a podporuje tyto operace

- **konstruktor** - vytvoří instanci ADT
- **selektor** - vyhledá žádaný prvek (indexer, At(), Pop()...)
- **modifikátor** - upraví data ve struktuře (indexer, Push(), Add()...)

Těmito operacemi jsou data jako taková uživateli skryta. Dynamická množina - množina dat, které předchozí operace podporuje, základ většiny ADT

Příklady ADT:

- Seznam
- Fronta
- Zásobník
- Trie
- Binární vyhledávací strom
- Hashmapa
- ...

ADT je často využíváno návrhovými vzory. V některých jazycích lze ADT vytvořit jako generické, tedy vhodné pro uchování libovolného datového typu (např List<int>). Tyto generické ADT jsou také často v jazycích již implementovány jako knihovny (C# v Generics, C++ v Standard Template Library). U jazyků, které toto nepodporují lze využít Objekt, a ten přetypovávat, což není ideální, ale funguje. U jazyků nemajících objekt je to horší, musíme přijít s vlastní specifickou implementací.