

Maturitní témata 2020/21

Otázka č. 24 – Počítačové sítě a programování

Kolekce:

- kolekce nebo kontejner je třída (základní konstrukční prvek objektově orientovaného programování sloužící jako předpis pro vznik objektu) nebo abstraktní datový typ (typy dat které jsou nezávislé na vlastní implementaci) obsahující sadu hodnot jednoho nebo různých typů umožňující přístupování k těmto hodnotám

- kolekce umožňuje zapisovat do sebe hodnoty a získávat je, cíl kolekce je sloužit jako úložiště objektů a zajišťovat k nim přístup, obvykle se kolekce používají pro uchovávání skupin objektů stejného typu podléhajících stereotypnímu zpracování

- pro přístupování ke konkrétnímu elementu kolekce můžou používat různé metody v závislosti na její logické organizaci

- implementace může připouštět provedení jednotlivých operací na celé kolekce, takové operace na celé kolekce můžou v mnoha případech podstatně usnadnit programování

- kolekce můžou být zkoumány ze třech hledisek:

1. z hlediska přístupování k prvkům kolekce
2. z hlediska uchovávání prvků kolekce
3. z hlediska procházení prvků kolekce

- obvykle kolekce implementuje a může provádět následující operace:

- oznamovat jaký má počet objektů které v sobě uchovává
- odstraňovat všechny objekty z kolekce
- přidávat nový objekt do kolekce
- odstraňovat objekt z kolekce
- poskytovat přístup k uchovanému objektu

- rychlost provádění těchto operací se liší v závislosti na konkrétní implementaci kolekce

- mnoho kolekcí se chovají jako kdyby uchovávali data do řady uspořádané nějakým způsobem s přístupem k jednomu nebo obou koncům, takové kolekce se nazývají lineární

- mezi lineární kolekce patří:

- Lineární seznam
- Pole
- Zásobník
- Fronta
- Prioritní fronta (často implementována jako Halda - stromová datová struktura)

- jiné kolekce mohou být interpretovány jako funkce → asociativní kolekce (např. množiny - pořadí datových položek nemá žádnou posloupnost, přidávání a odstraňování prvku ve množině)

Pole:

- pole (vektor), datová struktura která sdružuje daný vždy konečný počet prvků (čísel, textových řetězců, ...) stejného datového typu, k jednotlivým prvkům pole se přistupuje pomocí jejich indexu (celého čísla, označujícího pořadí prvku)

- velikost pole zůstává při běhu programu neměnná (některé programovací jazyky toto omezení nekladou, zvětšení pole je ale časově náročná operace)

- ve většině programovacích jazyků je pole vestavěným datovým typem

- operace s polem:

- přístup k prvku - probíhá v konstantním čase, pomocí indexu lze vypočítat přesnou adresu v paměti
- vyhledávání prvku - (lineární vyhledávání) je řešitelné v lineárním čase, v nejhorším případě je nutné projít celé pole
- vyhledávání prvku v seřazeném poli - hledání metodou půlení intervalu indexů pole (binární vyhledávání) má logaritmickou časovou složitost

- pole jsou jedním z nejzákladnějších datových typů, mají tedy široké použití nejen samy o sobě ale také jako součást komplexnějších struktur jako jsou haldy, hašovací tabulky či řetězce

- v praktických úlohách, zejména v náročnějších výpočetních a grafických aplikacích se uplatňují i vícerozměrná pole

Zásobník:

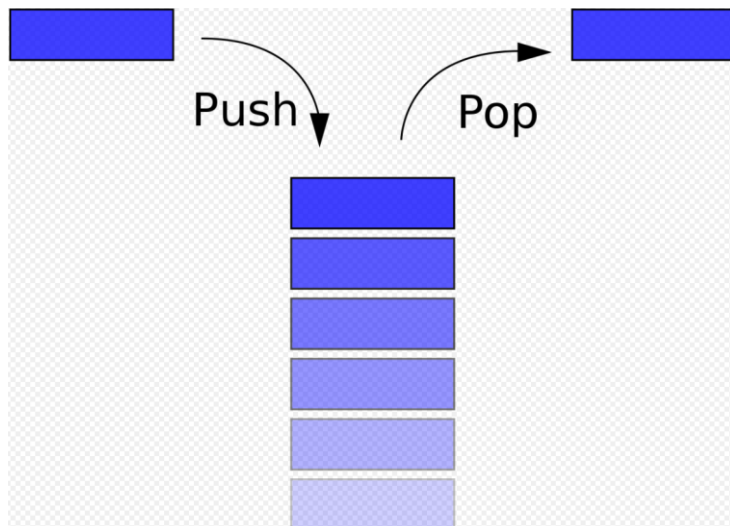
- stack, obecná datová struktura používaná pro dočasné ukládání dat, pro zásobník je typická manipulace s daty - data uložená jako poslední budou čtena jako první, proto se používá výraz LIFO (last in - first out)

- pro manipulaci s uloženými datovými položkami se používá tzv. ukazatel zásobníku který udává relativní adresu poslední přidané položky tzv. vrchol zásobníku

- obsahem zásobníku mohou být jakékoli datové struktury, může být realizován jak programovými prostředky, tak i elektronickými obvody

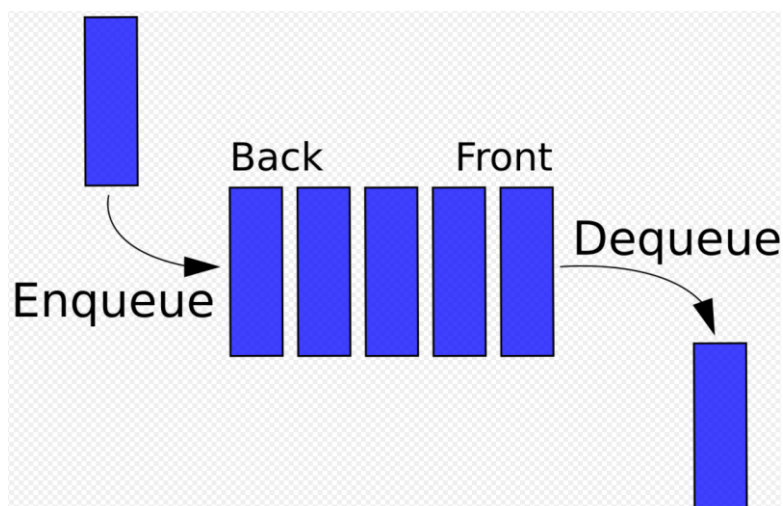
- nejznámější aplikací zásobníku je vnitřní zásobník realizovaný procesorem

- ve většině vyšších programovacích jazyků může být zásobník poměrně jednoduše implementován pomocí pole, nebo lineárního seznamu, to co identifikuje datovou strukturu jako zásobník není implementace ale rozhraní - uživatel smí pouze přidávat (push) nebo ubírat (pop) hodnoty z pole či lineárního seznamu spolu s několika málo pomocnými funkcemi



Fronta:

- abstraktní datový typ, typu FIFO (first in - first out), opakem zásobníku (LIFO)
- frontu lze implementovat pomocí pole - prvky fronty se postupně řadí do prvků pole počínaje nejnižším indexem, data se z pole čtou od nejnižšího indexu, je třeba dávat pozor na zaplnění pole (statické pole má omezenou délku), přesouvání prvků pole směrem k začátku ale je hlavně při velkém množství dat náročné → proto se pole používá jako tzv. kruhová fronta a po vyčerpání volného místa na konci pole se zapisuje na uvolněná místa na začátku pole
- fronta implementovaná lineárním seznamem se podobá zásobníku liší se ovšem v přidávání na konec, u fronty se hodí mít dva ukazatele (na začátek a na konec kvůli snadnějšímu přidávání)
- využití:
 - synchronizační primitivum
 - operátor roura (pipe) - komunikace mezi procesy v operačních systémech
 - kruhový buffer - vyrovnávací paměť pro datové toky
 - řazení prioritní frontou (haldou)



Seznam:

- dynamická datová struktura vzdáleně podobná poli (umožňuje uchovat velké množství hodnot ale jiným způsobem), obsahující jednu a více datových položek (struktur) stejného typu které jsou navzájem lineárně provázány vzájemnými odkazy pomocí ukazatelů nebo referencí, aby byl seznam lineární nesmí existovat cykly ve vzájemných odkazech
- seznamy mohou existovat jednosměrné a obousměrné, v jednosměrném seznamu odkazuje každá položka na položku následující a v obousměrném seznamu odkazuje položka na následující i předcházející položky
- zavádí se také ukazatel nebo reference na aktuální (vybraný) prvek seznamu, na konci (a začátku) seznamu musí být definována zarážka označující konec seznamu, pokud vytvoříme cyklus tak že konec seznamu navážeme na jeho počátek (jedná se o kruhový seznam)