

VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA

Katedra technických studií

Semestrální projekt xDSA

Autor práce: Musil Adam

Jihlava 2018

Zadání:

2. Zápis studentů na předměty

Napište program pro vedení a sledování zápisu studentů na předměty. Program pak umožňuje

vytvářet a sledovat tabulku zápisů studentů a zápisů na předměty. Předpokládá se, že předmět má

neomezenou kapacitu, studenti zapsaní na předmět mohou svoji registraci zrušit.

Předpokládá se tak

intenzivní zápis předmětů a rušení předmětu studentem. Aplikace musí vždy poskytnout seznam

studentů seřazený dle studentského čísla.

Řešení:

Samotní studenti budou kvůli zapouzdření reprezentování třídou Student.

Tabulka studentů může být reprezentována mnoha datovými strukturami:

Pole – staticky alokované pole je nevhodné protože nelze předem určit množství potřebného místa, u dynamicky alokovaného pole vzniká problém při naplnění a následně realokaci, která je může být velmi časově náročná. Dalším problémem je mazání prvku, nejdříve je potřeba prvek nalézt – binárním půlením pokud je pole setříděné nebo sekvenčně pokud není, smazat prvek a potom přesunout všechny prvky “za ním” o pozici dopředu.

Spojový seznam – rychlý zápis a odstranění prvků, značným zpomalením vkládání prvku je ověření zda se prvek už v seznamu nevyskytuje protože je potřeba nejdříve projít celý seznam. Také je potřeba před vypsáním prvků seznam setřídít nebo udržovat nebo vkládat nové prvky ve správném pořadí. Další výhodou je jednoduchá implementace.

Hashovací tabulka – rychlý zápis a přístup k prvkům ale nelze vypsát data setříděná, takže by bylo potřeba před vypsáním nejdříve zapsat data do jiné datové struktury, kterou potom lze třídit.

Binární vyhledávací strom – Rychlý zápis a odstranění prvků. Struktura binárního stromu zabraňuje výskytu duplicitních prvků a při inorder průchodu stromem lze prvky vypsát seřazené bez potřeby použití třídících algoritmů. Nevýhodou je složitá implementace.

Vzhledem k požadavkům jsem se rozhodl řešit zadání pomocí červeno černého stromu.

Implementace:

Třída Student:

Privátní proměnné:

- String jmeno
- String prijmeni
- Int id

Veřejné metody:

- Student(int) – konstruktor primárně určený pro generování nových záznamů pro testování, jméno a příjmení nastaví na hodnotu “John Doe”.
- Student(string, string, int) – konstruktor pro vytvoření studenta. Nastaví jméno, příjmení i ID podle zadaných hodnot.
- getName – metoda pro získání jména a příjmení studenta, navrací řetězec znaků ve formátu jméno + “ “ + příjmení.
- getID – metoda pro získání ID studenta, návratový typ integer.
- Přetížený operator << pro tisk do streamu.
- Přetížené operátory porovnání <, >, ==.

Třída RnBtree:

Implementace červeno černého stromu. Jednotlivé prvky stromu jsou tvořeny strukturou Node.

Node

proměnné

- leftChild – pointer na levého potomka, pokud potomek neexistuje má hodnotu nullptr.
- rightChild – pointer na pravého potomka.
- parent – pointer na rodiče, pokud je Node kořenem stromu má hodnotu nullptr.
- isRed – integer určující barvu, 0 – černá, 1 – červená, 3- dvojité černá
- value – data uložená v node (Student)

RnBtree:

Privátní proměnné a metody:

- root – pointer na kořen stromu, pokud má hodnotu nullptr strom je prázdný.
- rotateLeft – metoda levé rotace, pro obnovení vlastností červenočerného stromu.
- rotateRight – metoda pravé rotace.
- isRed – metoda pro zjištění barvy node. Metoda nejdříve zkontroluje zda node existuje.
- swapColours – metoda pro prohození barev 2 node.
- setColour – metoda pro nastavení barvy node
- fixViolations – metoda pro opravu struktury stromu po vložení prvku, dle potřeby rotuje a obarvuje položky stromu.
- Insert – rekurzivní pomocná metoda pro vložení nového prvku.
- printInOrder – rekurzivní pomocná metoda pro inorder tisk do streamu.
- deleteHelper – rekurzivní pomocná metoda pro odstranění prvku.
- minValueNode – rekurzivní metoda, která navrátí nejmenší potomek zadaného prvku.
- fixDeletion – metoda pro opravu struktury stromu po smazání prvku
- cleanup – rekurzivní postorder metoda na odstranění všech prvků stromu.

Veřejné metody:

- `RnBtree()` – konstruktor vytvoří prázdný strom.
- `push(T)` – vložení prvku, volá pomocnou metodu `insert`.
- `Print(ostream&)` – metoda pro tisk do streamu, volá `printInOrder`.
- `deleteValue(int)` – metoda pro odstranění prvku.
- `isEmpty()` – metoda pro zjištění jestli je strom prázdný.
- `Purge()` – odstraní ze stromu všechny prvky.
- `Search(int)` – metoda pro vyhledávání prvku ve stromu, pokud se prvek ve stromu nenalézá vrací `nullptr` jinak vrátí ukazatel na vyhledávaný prvek.

Program se ovládá pomocí třídy `Menu`.

Menu

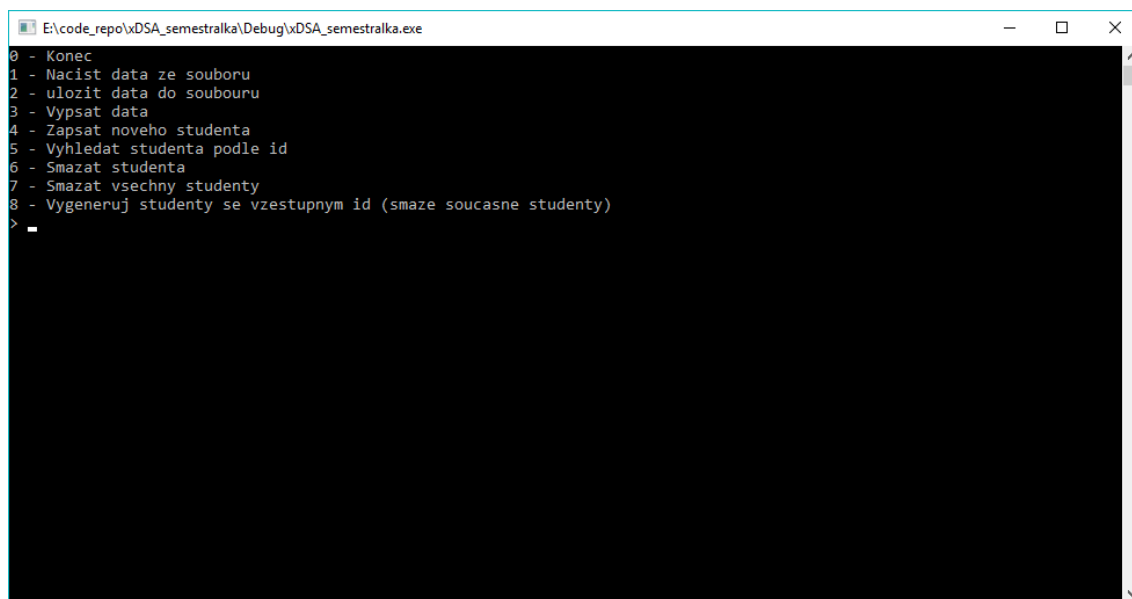
Chráněné proměnné:

- `menuItems` – vektor položek menu
- `userInput` – pro uložení volby.

Veřejné metody:

- `Menu()` – defaultní konstruktor.
- `Menu(std::initializer_list<string>)` – přetížený konstruktor, naplní vektor `menuItems`.
- `addMenuItem(string)` – přidání další položky do menu
- `printMenu()` – vypíše menu do `cout`.
- `readUserChoice()` – načte volbu uživatele z `cin` do `userInput`
- `getUserChoice()` – navrátí obsah `userInput`

Ovládání:



```
E:\code_repo\xDSA_sememstralka\Debug\xDSA_sememstralka.exe
0 - Konec
1 - Nacist data ze souboru
2 - ulozit data do souboru
3 - Vypsati data
4 - Zapsat noveho studenta
5 - Vyhledat studenta podle id
6 - Smazat studenta
7 - Smazat vsechny studenty
8 - Vygeneruj studenty se vzestupnym id (smaze soucasne studenty)
>
```

Po spuštění se zobrazí menu s nabídkou. Program umožňuje načíst či uložit data do textového souboru, každý student je vypsán na jeden řádek ve formátu “ID jméno příjmení”. Data lze vypsát do konzole ve stejném formátu. Při zvolení možnosti “Zapsat noveho studenta” je uživatel dotázán na jméno, příjmení a id studenta, který je poté vytvořen a přidán do stromu. Vyhledání studenta podle id vypíše id jméno a příjmení studenta pokud se jeho záznam vyskytuje ve stromě, pokud ne vypíše “zaznam nenalezen”. Položka Smazat studenta vymaže záznam studenta podle id ze stromu. Smazat Všechny studenty vymaže všechny záznamy. Položka Vygeneruj studenty se vzestupnym id nejdříve vyčistí strom a poté vytvoří zadaný počet studentů a zařadí je do stromu, například při zadání hodnoty 100 vytvoří 100 studentů s id 0 až 99 a zařadí je.