Информационна система

Проекта може да се раздели на две части:

1. Подържащи структури
    a. BTree
    b. BNode
    c. Database
    d. *Index
2. Специфични за проекта структури
    a. Student
    b. InformationSystem

Класът Database има за цел да съхранява голям брой данни, които бързо да бъдат достъпвани, чрез подходящ ключ.

Обекти

BTree: идеално балансирано дърво, предназначено за голям обем от данни.

BTree<T>(int degree): създава празно дърво с минимална *degree* разклоненост, съхраняващо обекти от тип **T**.

int degree: минималния брой обекти, пазени в един връх на дървото.

bool #empty: true ако в дървото има поне един елемент, **false** в противен случай.

const BNode* #get_root: връща указател към корена на дървото.

T* #search(const T& key): ако *key* се намира в дъвото връща указател към него, в противен случак NULL.

#insert(const T& key): вкарва *key* в дървото.

#remove(const T& key): изтрива *key* от дървото.

BNode<T>(int degree, bool leaf): създава празен връх, съхраняващ обекти от тип **T,** с минимална *degree* разклоненост. При *leaf* = true върхът е листо.

T* #get_values: масив от всички стойности в даден връх.

BNode<T> #get_children:** масит от всички наследници на даден връх.

int #size: текущия брой елементи във върха.

bool #full: true ако върхът съдържа **2*degree - 1** елемента, в противен случай **false**.

bool #leaf: true ако върхът е листо, в противен случай **false**.

T* #search(const T& key): ако *key* се намира във върхът или в някой от неговите наследници връща указател към него, в противен случак NULL.

#non_full_insert(const T& key): вкарва *key* в непълен връх.
#split_child(int child_index): разделя наследник с child_index на две.
#find_key_index(const T& key): предполагаемата позиция на *key* във върха.
#max(): най-голямата стойност в йерархията на върха.
#min(): най-малката стойност в йерархията на върха.
#remove(const T& key): изтрива *key* от йерархията на върха.

Database: Базаданни използваща BTree за индексация на записите.

Database<Row, Index>: Съхранява записи от тип *Row* и ги индексира чрез типа *Index*.
const Row* #search(query): указател към запис отговарящ на *query*.
void #insert(Row): добавя *Row* към базата, ако вече не е съществува запис с еднакъв индекс.
const Row* #update(query, Row): презаписва *Row* върху запис отговарящ на *query*
const Row* #update(query, function(Row)): прилага *function* върху запис отговарящ на *query*
void #remove(query): изтрива запис отговарящ на *query* заявката.
void #dump(ostream& os): принтира в нарастващ ред всички записи от базата.

*Index: структура, която дефинира начина по-който да се вземе ключ спрямо който да се идентифицира запис в Database.

typedef TYPE index_type
index_type #operator() (Row): уникален ключ на Row

Student: структура студент

Student (const char* name, double grade, tel_type tel): създава студент с име *name*, среден успех *grade* и телефонен номер *tel*.
const char* #get_name: името на студента.
double #get_grade: спредния успех на студента.
unsigned long #get_tel: телефонния номера на студента.
void print(ostream& os): извежда студента на os стреам-а.
ostream& operator<<(ostream& os, const Student& s) === print(os);
ostream& operator<<(std::ostream& os, const Student* s) === print(os);
istream& operator>>(std::istream& is, Student& s) === Student (n, g, t);

StudentIndex: дефинира начина по-който Student да се индексира от Database.

typedef string index_type
index_type #operator() (const Student& s): името на студент *s*.

InformationSystem: иформационна система, работеща върху Database<Student, StudentIndex>.

#load(const char* filename): прочита от файл *filename* студенти.
#save(const char* filename): запазва във файл *filename* текущите студенти.
#add_student(cons Student& s): добавя студент *s* към системата.
#add_student(const char* name, double grade, Student::tel_type tel): създава и добавя студент с име *name*, среден успех *grade* и телефон *tel*.
#remove_student(const char* name): изтрива студент с име *name*.
double #average_grade(const char* name): средния успех на студент с име *name*.
#average_grade(const char* student_name, double grade): задава средния успех на студент с име *name* на *grade*.
Student::tel_type #telephone(const char* student_name): телефонния номер на студент с име *name*.
#telephone(const char* student_name, Student::tel_type tel): задава телефонния номер на студент с име *name* на *grade*.

Подобрения

Database: директна рабора с файлове, индексиране по ред в даден файл.