

Проект "Информационна система"

Проектът представлява програма, която реализира информационна система, извършващ описаните по-долу действия над база от данни (БД). БД се състои от записи, съдържащи: име, среден успех и телефонен номер.

Библиотеките, които съм ползвала са :

#include <iostream> - за стандартния вход/изход

#include <string> - за представянето на имената на студентите в БД

#include <fstream> - за четене/писане във файл

#include <queue> - за съхранение на данни от файл и извеждането им на конзолата (контейнерът <priority_queue>)

В проекта съм използвала 4 основни класа :

1.class Student - за реализацията на данните, съдържащи информация за студент

2.class node - за представяне на връх в В-дърво

3.class btree - за реализацията на самото В-дърво

4.class Mydatabase - за "сглобяване" на горните класове и управляване на дейностите, върху БД

Следва описание на основните функции за всеки клас

1.class Student

Student(string _name = "anonymous", double = 2, int = 0) - конструктор

string getName() const - селектор, който връща името

double getMark() const - селектор, който връща отценката

int getPhone() const - селектор, който връща телефонният номер

void setMark(double newmark) - мутатор, който променя отценката

void setPhone(int newphone) - мутатор, който променя телефонният номер

friend istream& operator>>(istream& is, Student& s) - оператор за вход

friend ostream& operator<<(ostream& os, Student const& s) - оператор за изход

2. class node

int noofkeys - показва, колко ключа има в даден връх

mypair data[MAX] - масив , които пази съответните двойки (ключ и указател към следващ връх, като за ключове съм ползвала самите обекти (Student))

node *father - указател към родителя

node *first - указател към наследника

node() - конструктор

int leafnode() - връща 1 ако дадения връх е лист и 0 в противен случай

void insertinnode(mypair x) - вмъква дадена двойка

mypair splitnode(mypair x) - разделя 1 връх на 2 ако се надхвърли максимално позволения брой за двойки в даден връх

void display() - отпечатва върха

3. class btree

int mkeys - максимален брой ключове във връх

node *root - указател към връх

void insert(Student x) - вмъкване на студент

void displaytree() - отпечатване на дървото

node *search(string x) - търсене на връх, които съдържа студент със съответното име

void Delete(Student x) - изтриване на студент

void to_txt_tree() - отпечатване на дървото в txt файл

4.class Mydatabase

Mydatabase() - конструктор, който създава база данни, като ги записва във txt файл

void from_file_to_tree(btree&) - четена от съответния файл и прехвърляне в B-дървото

void getGradebyName(btree, string) - отпечатване на оценка по подадено име

void getPhonebyName(btree, string) - отпечатване на телефонния номер по подадено име

void updateGrade(btree&, string,double) - промяна на оценка по подадено име

void updatePhone(btree&, string,int) - промяна на телефонния номер по подадено име

void sorted_database() - сортиране на базата данни и отпечатването (за тази функция съм използвала допълнителен файл, в който да прехвърля данните от B-дървото, след това прочитам съответните данни и ги прехвърлям в приоритетна опашка, и накрая извеждам на конзолата

void deleting(btree&,string) - изтриване на студент по подадено име