Приклад. Розробити інтерфейси

IComparable

compareTo

та

IWritable

wtiteToFile

readFromFile

На основі цих інтерфейсів створити класи Student (прізвище, кількість оцінок, масив оцінок) та Group (кількість студентів, масив покажчиків на студентів). Реалізувати можливість збереження інформації про студентів у файлі.

#pragma once

\_\_interface IComparable

{

int compareTo(void\* obj);

};

//-----------------------------------------------------

#pragma once

#include<string>

using namespace std;

\_\_interface IWriteable

{

void writeToFile(string fileName);

void readFromFile(string fileName);

};

//-----------------------------------------------------

#pragma once

#include<string>

#include "IComparable.h"

#include <fstream>

using namespace std;

class Student:public IComparable

{

public:

string Name;

int Course;

int scoresCount;

int\* scores;

Student(string name, int course, int scoresCount, int firstScore, ...);

Student();

//Student(string data); //Ivan 3 4 12 11 10 8 split

string toString();

void input();

void input(ifstream &file);

double getAverageScore();

int compareTo(void\* obj);

~Student();

};

//-----------------------------------------------------

#include "stdafx.h"

#include "Student.h"

#include<iostream>

using namespace std;

Student::Student():Name("no name"),Course(0),scoresCount(0)

{

}

Student::Student(string name, int course, int scoresCount, int firstScore, ...)

{

Name = name;

Course = course;

this->scoresCount = scoresCount;

int\* temp=&firstScore;

scores = new int[scoresCount];

for (int i = 0; i < scoresCount; i++)

{

scores[i] = temp[i];

}

}

string Student::toString()

{

string s= Name+" "+to\_string(Course)+" "+to\_string(scoresCount);

for (int i = 0; i < scoresCount; i++)

{

s += " " + to\_string(scores[i]);

}

return s;

}

void Student::input()

{

printf("Name:");

cin >> Name;

printf("Course:");

cin >> Course;

printf("Scores count:");

cin >> scoresCount;

scores = new int[scoresCount];

for (int i = 0; i < scoresCount ; i++)

{

printf("Scores %d", i);

cin >> scores[i];

}

}

void Student::input(ifstream &file)

{

file >> Name;

file >> Course;

file >> scoresCount;

scores = new int[scoresCount];

for (int i = 0; i < scoresCount; i++)

{

file>> scores[i];

}

}

double Student::getAverageScore()

{

double sum=0;

for (int i = 0; i < scoresCount; i++)

{

sum += scores[i];

}

return sum/scoresCount;

}

int Student::compareTo(void \* obj)

{

Student\* otherStudent = (Student\*)obj;

const double eps = 0.0001;

double avS1 = getAverageScore();

double avS2 =otherStudent->getAverageScore();

if (abs(avS1-avS2)<eps)

{

return 0;

}

if (avS1>avS2)

{

return 1;

}

else

{

return -1;

}

}

Student::~Student()

{

delete[] scores;

}

//-----------------------------------------------------

#pragma once

#include "Student.h"

#include "IWriteable.h"

#include <string>

using namespace std;

class Group:public IWriteable

{

public:

int studentsCount;

Student\*\* students;

Group(int studentsCount);

Group(string fileName);

Group();

void input();

void print();

void writeToFile(string fileName);

void readFromFile(string fileName);

Student\* getTheBest();

~Group();

};

//-----------------------------------------------------

#include "stdafx.h"

#include "Group.h"

#include <iostream>

#include <fstream>

using namespace std;

Group::Group(int studentsCount)

{

this->studentsCount = studentsCount;

students = new Student\*[studentsCount];

}

Group::Group(string fileName)

{

readFromFile(fileName);

}

Group::Group()

{

studentsCount = 0;

students = 0;

}

void Group::input()

{

printf("Students count: ");

cin >> studentsCount;

students = new Student\*[studentsCount];

for (int i = 0; i < studentsCount; i++)

{

students[i] = new Student();

students[i]->input();

}

}

void Group::print()

{

printf("------- Students group ------\n");

for (int i = 0; i < studentsCount; i++)

{

printf("%s\n", students[i]->toString().data());

}

}

void Group::writeToFile(string fileName)

{

ofstream file(fileName);

file << studentsCount<<endl;

for (int i = 0; i < studentsCount; i++)

{

file << students[i]->toString()<<endl;

}

file.close();

}

void Group::readFromFile(string fileName)

{

ifstream file(fileName);

file >> studentsCount;

students = new Student\*[studentsCount];

for (int i = 0; i < studentsCount; i++)

{

students[i] = new Student();

students[i]->input(file);

}

file.close();

}

Student \* Group::getTheBest()

{

Student\* theBest = students[0];

for (int i = 1; i < studentsCount; i++)

{

if (theBest->compareTo(students[i])<0)

{

theBest = students[i];

}

}

return theBest;

}

Group::~Group()

{

for (int i = 0; i < studentsCount; i++)

{

delete students[i];

}

delete[] students;

}

//-----------------------------------------------------

// ConsoleApplication50.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

#include <fstream> //0. Підключаємо <fstream>

#include <string>

#include "Group.h"

using namespace std;

int main()

{

Group\* group = 0;

int answer;

do

{

system("cls");

printf("1. Read from keyboard\n");

printf("2. Read from file\n");

printf("3. Print to screen \n");

printf("4. Save to file\n");

printf("5. Get the best student\n");

printf("6. Exit\n");

printf("Your choise:");

cin >> answer;

string fileName;

switch (answer)

{

case 1:group = new Group();

group->input();

break;

case 2:

printf("File name :");

cin >> fileName;

group = new Group(fileName);

break;

case 3:

if (group != 0)

{

group->print();

}

else {

printf("Group is empty");

}

break;

case 4:

if (group != 0)

{

printf("File name :");

cin >> fileName;

group->writeToFile(fileName);

}

else {

printf("Group is empty");

}

break;

case 5:

if (group != 0)

{

Student\* theBestStudent= group->getTheBest();

printf("The best: %s", theBestStudent->toString().data());

}

else {

printf("Group is empty");

}

break;

case 6:

if (group != 0)

{

delete group;

}

break;

}

system("pause");

} while (answer != 6);

return 0;

}