

Приклад лабораторної роботи №3

Автор : Рибалко Д.

Клас Train (а саме інтерфейсна частина Train.h та описова частина Train.cpp) з л.р.№2 залишаються без змін (ми їх просто копіюємо в л.р. №3)
Тепер створимо новий клас TrainAddition

Ось інтерфейсна частина класу TrainAddition, вона зберігається в файлі під назвою TrainAddition.h

```
#ifndef TRAINADDITION_H
#define TRAINADDITION_H
#include "string.h"
#include "Train.h"

class TrainAddition : public Train
{
    char punktDepart[20];
    int mezhoblast;
    float timeDepart;

public:
    TrainAddition();

    void setMezhoblast(int mezhoblast){ this->mezhoblast=mezhoblast;}
    void setPunktDepart(char *punktDepart){ strcpy(this->punktDepart,punktDepart);}
    void setTimeDepart(float timeDepart){ this->timeDepart=timeDepart;}
    int getMezhoblast(){return mezhoblast;}
    char*getPunktDepart(){return punktDepart;}
    float getTimeDepart(){return timeDepart;}

    void enterDannieKlav();
    void showDanie();
    int amountMest();
    virtual ~TrainAddition();
protected:
private:
};

#endif // TRAINADDITION_H
```

Ось описова частина класу TrainAddition, вона зберігається в файлі під назвою TrainAddition.cpp

```
#include <iostream>
#include <stdio.h>
#include <windows.h>
#include "string.h"
#include "TrainAddition.h"
using namespace std;
TrainAddition::TrainAddition()
{
    strcpy(punktDepart,"");
    mezhoblast=0;
    timeDepart=0;
}
TrainAddition::~TrainAddition()
```

```

{
    //dtor
}
void TrainAddition::enterDannieKlav()
{
    TrainAddition temp;
    getchar();
    cout << "Пункт отбытия: ";
    gets(temp.punktDepart);
    setPunktDepart(temp.punktDepart);
    cout << "Время отбытия: ";
    cin >> temp.timeDepart;
    setTimeDepart(temp.timeDepart);
    Train::enterDannieKlav();
    cout << "Межобластные места: ";
    cin >> temp.mezhoblast;
    setMezhoblast(temp.mezhoblast);
    cout << endl;
}
void TrainAddition::showDanie()
{
    cout << "Пункт отбытия:      " << punktDepart << endl;
    cout << "Время отбытия:      " << timeDepart << endl;
    Train::showDanie();
    cout << "Межобластные места:  " << mezhoblast << endl;
    cout << endl;
}

int TrainAddition::amountMest()
{
    int amount=0;
    amount=getPlatskart()+getKupe()+getVip()+getMezhoblast();
    return amount;
}

```

А в класі TrainArray, ми будемо працювати з об'єктами < TrainAddition >.

```

#ifndef TRAINARRAY_H
#define TRAINARRAY_H
#include "Train.h"

```

```

class TrainArray
{
private:
    TrainAddition *sostav;
    int kolvo;
public:
    TrainArray();
    void setKolvo(int n){kolvo=n;}
    int getKolvo(){return kolvo;}
    void creatArrayTrain(int n);
    virtual ~TrainArray();
    void enterArrayTrain();
    void showsArrayTrain();
    void fileWrite(char* txt);
    int fileSize(char *txt);
    void fileRead(char *txt);
    void zadanie1();
    void zadanie2();
    void zadanie3();
protected:
};

```

```
#endif // TRAINARRAY_H
```

```
#include <fstream>
#include <stdlib.h>
#include <iostream>
#include "Train.h"
#include "TrainArray.h"
#include "TrainAddition.h"
using namespace std;
TrainArray::TrainArray()
{
    //ctor
}
void TrainArray::creatArrayTrain(int n)
{
    sostav=new TrainAddition [n];
}
TrainArray::~TrainArray()
{
    //dtor
}
void TrainArray::enterArrayTrain()
{
    for(int i=0;i<getKolvo();i++)
    {
        sostav[i].enterDannieKlav();
    }
}

void TrainArray::showsArrayTrain(){
    system("CLS");
    for(int i=0;i<getKolvo();i++)
    {
        sostav[i].showDanie();
    }
}

void TrainArray::fileWrite(char* txt)
{
    ofstream t(txt,ios::binary | ios::app);
    for(int i=0;i<getKolvo();i++)
    {
        t.write((char*)&sostav[i], sizeof sostav[i]);
    }
    t.close();
}

int TrainArray::fileOfSize (char *txt) //определение количества поездов в файле
{
    ifstream t(txt,ios::binary);
    t.seekg(0,ios_base::end);
    int k = t.tellg()/sizeof (TrainAddition);
    t.close();
    return k;
}

void TrainArray::fileRead(char *txt) //чтение из файла
{
    ifstream t(txt,ios::binary );
    for(int i=0;i<kolvo;i++)
    {
        t.read((char*)&sostav[i], sizeof sostav[i]);
    }
    t.close();
}
```

```
}
```

```
void TrainArray::zadanie1()
{
    getchar();
    cout << "Задание 1" << endl;
    cout << "Введите город: ";
    char town[20];
    gets(town);
    bool flag=0;
    for(int i=0;i<kolvo;i++)
    {
        if (!strcmpi(sostav[i].getPunkt(),town))
        {
            sostav[i].showDanie();
            flag=1;
        }
    }
    if(!flag) cout << "Не найдено" << endl;
}
```

```
void TrainArray::zadanie2()
{
    cout << "Задание 2" << endl;
    cout << "Введите город: ";
    char town[20];
    double time2;
    gets(town);
    bool flag=0;
    cout << "Введите время: ";
    cin >> time2;
    for(int i=0;i<kolvo;i++)
    {
        if (!strcmpi(sostav[i].getPunkt(),town) && sostav[i].getTime()==time2){
            sostav[i].showDanie();
            flag=1;}
    }
    if(!flag) cout << "Не найдено" << endl;
}
```

```
void TrainArray::zadanie3()
{
    getchar();
    cout << "Задание 3" << endl;
    cout << "Введите город: ";
    char town[20];
    int obshmesta;
    bool flag=0;
    gets(town);
    cout << "Введите кол.- во мест: ";
    cin >> obshmesta;
    for (int i=0;i<kolvo;i++)
    {
        if (!strcmpi(sostav[i].getPunkt(),town) && sostav->amountMest()== obshmesta)
        {
            sostav[i].showDanie();
            flag=1;}
    }
    if(!flag) cout << "Не найдено" << endl;
}
```

І все це починає працювати, коли ми створимо в головній функції main об'єкт класу TrainArray

```
////////main.cpp////////
#include <iostream>
#include <stdio.h>
#include <windows.h>
#include <fstream>
#include "Train.h"
#include "TrainArray.h"
#include "TrainAddition.h"
using namespace std;

int main()
{
    int n,kolvo;
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    char txt[20];
    TrainArray vagon;
    system("CLS");
    cout << "\t< Меню >"<< endl;
    cout << "Запис в файл - 1, чтени с файла - 2"<< endl;
    cout << "Введите номер: ";
    cin >> n;
    system("CLS");
    switch (n)
    {
    case 1:
    {
        getchar();
        cout << " --Запись в файл--" << endl;
        cout << "Название файла: ";
        gets(txt);
        cout<<"Кол.-во поездов: ";
        cin >> kolvo;

        vagon.setKolvo(kolvo);
        vagon.creatArrayTrain(kolvo);
        vagon.enterArrayTrain();
        vagon.showsArrayTrain();
        vagon.fileWrite(txt);

        break;
    }
    default :
    {
        getchar();
        cout << " **Чтение с файла**" << endl;
        cout << "Название файла: ";
        gets(txt);
        kolvo=vagon.fileOfSize(txt);
        cout << "Количество поездов: " << kolvo << endl;

        vagon.setKolvo(kolvo);
        vagon.creatArrayTrain(kolvo);
        vagon.fileRead(txt);
        vagon.showsArrayTrain();

    }
    }
    vagon.zadanie1();
    vagon.zadanie2();
    vagon.zadanie3();
    return 0;
}
```

