

----------------------------- Date.h --------------------------------

#pragma once

struct Date

{

int D;

int M;

int Y;

};

------------------------ Student ----------------

#pragma once

#include "Date.h"

#include <string>

using namespace std;

namespace MySpace //Для прикладу

{

struct Student

{

//Спільні для усіх студентів поля

string Name;

Date BirthDate;

char Gender; //Поле індикатор (які із альтернативних треба викорситовувати)

//Альтернативні поля

union {

struct {

bool Army; //Для хлопців

bool Auto;

};

double Grant; //Для дівчат

};

};

}

------------------ Main-----------------

// ConsoleApplication25.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include "Date.h"

#include "Student.h"

#include <iostream>

using namespace std;

using namespace MySpace; //Для прикладу

void inputDate(Date & date)

{

printf("Day :");

cin >> date.D;

printf("Month :");

cin >> date.M;

printf("Year :");

cin >> date.Y;

}

void printDate(const Date& date)

{

printf("%d.%d.%d :",date.D,date.M,date.Y);

}

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

void inputStudent(Student& student)

{

printf("Name :");

cin >> student.Name;

printf("--- Your birth date -----\n");

inputDate(student.BirthDate);

printf("Gender (M/W) :");

cin >> student.Gender;

switch (student.Gender)

{

case 'M':

printf("Army (Y/N):");

char answer;

cin >> answer;

student.Army = (answer == 'Y') ;

printf("Auto (Y/N):");

cin >> answer;

student.Auto = (answer == 'Y');

break;

default:

printf("Grant :");

cin >> student.Grant;

break;

}

}

void printStudent(const Student& student)

{

printf("Name :%s\n", student.Name.data());

printf("Birth date : "); printDate(student.BirthDate);

printf("Gender %c:\n",student.Gender);

switch (student.Gender)

{

case 'M':

printf("Army: %c \n", student.Army? 'Y': 'N' );

printf("Auto: %c \n", student.Auto? 'Y' : 'N');

break;

default:

printf("Grant : %f\n", student.Grant);

break;

}

}

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

Student\* inputStudentsGroup(int studentsCount)

{

Student\* students = new Student[studentsCount];

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

{

inputStudent(students[i]);

}

return students;

}

void printStudentsGroup(Student\* students, int studentsCount)

{

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

{

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

printStudent(students[i]);

}

}

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

void printMenWhoHaveAuto(Student\* students, int studentsCount)

{

bool isEmpty = true;

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

{

if (students[i].Gender=='M' && students[i].Auto)

{

printStudent(students[i]);

isEmpty = false;

}

}

if (isEmpty)

{

printf("No such students");

}

}

int main()

{

int studentsCount;

printf("Students count =");

cin >> studentsCount;

Student\* students=inputStudentsGroup(studentsCount);

printMenWhoHaveAuto(students, studentsCount);

system("pause");

return 0;

}