

Лабораторная работа №3. Структуры и классы

Цели работы

1. Изучение основ абстракции.
2. Изучение правил определения и использования структур данных, создание переменных типа структуры, обращение к полям.
3. Изучение основ создания пользовательских типов.

Справка

Структура представляет собой абстрактный тип данных, что обозначает производный тип, то есть созданный программистом, с использованием уже существующих типов. Введение абстрактных типов данных способствует смещению акцента в сторону предметной области, то есть существует возможность создания и использования типов данных, с наибольшей полнотой отображающей особенности программируемой задачи. Приведем пример определения структуры

```
struct Book{
// определение полей структуры
    char *author;
    char *title;
    int year;
    int pages;
};
```

Классы представляют абстрактные типы данных с открытым интерфейсом и скрытой внутренней реализацией. Определение простейшего класса без наследования имеет вид:

```
class имя_класса {
// по умолчанию раздел private – частные члены класса public: // открытые функции и переменные
    класса
};
```

Пример:

Создается класс Student. Формируется динамический массив объектов. При тестировании выводится: сформированный список студентов, список студентов заданного факультета, список студентов для заданных факультета и курса.

```
#include <conio.h>
#include <string.h>
#include <iostream.h>
//Структура для даты рождения
struct date{
    char daymon[6];
    int year;
};
//Класс Студент
class Student{
    char name[30]; //private
    date t;
    char adr[30], fac[20]; int kurs;
public: Student();
    char *getfac();
    int getkurs();
    void show();
};
Student::Student(){
    cout<<"Input name:";
    cin>>name;
    cout<<"Input date of born\n";
    cout<<"Day.mon:";
    cin>>t.daymon;
```

```

        cout<<"Year:";
        cin>>t.year;
        cout<<"Input adr:";
        cin>>adr;
        cout<<"Input fac:";
        cin>>fac;
        cout<<"Input kurs:";
        cin>>kurs;
    }
    void Student::show() {
        cout<<"Name : "<<name<<endl;
        cout<<"Was born : "<<t.daymon<< '.'<<t.year<<endl;
        cout<<"Address: "<<adr<<endl;
        cout<<"Fac : "<<fac<<endl;
        cout<<"Kurs : "<<kurs<<endl;
    }
    char *Student::getfac() { return fac; }
    int Student::getkurs() { return kurs; }
    void spisfac(Student spis[],int n){
        char fac[20];
        cout<<"Input faculty:";
        cin>>fac;
        for(int i=0;i<n;i++){
            if(strcmp(spis[i].getfac(),fac)==0)
                spis[i].show();
        }
    }
    //Список студентов заданных факультета и курса
    void spisfackurs(Student spis[],int n){
        int i,k;
        char fac[20];
        cout<<"Input faculty:";
        cin>>fac;
        cout<<"Input the course:";
        cin>>k;
        for(i=0;i<n;i++){
            if((strcmp(spis[i].getfac(),fac)==0)&&spis[i].getkurs()==k)
                spis[i].show();
        }
    }
    //Функция main
    void main(){
        Student *spis;
        int n;
        cout<<"Input a number of students: ";
        cin>>n;
        spis=new Student [n];
        for(int i=0;i<n;i++) {
            cout<<"===== "<<endl;
            spis[i].show();
        }
        spisfac(spis,n);
        spisfackurs(spis,n);
        delete [] spis;
        cout<<"press any key!" while(!kbhit());
    }
}

```

Задание лабораторной работы

Разработать классы для описанных ниже объектов. Включить в класс методы set (...), get (...), show (...).
Определить другие методы.

1. **Student:** Фамилия, Имя, Отчество, Дата рождения, Адрес, Теле- фон, Факультет, Курс. Создать массив объектов. Вывести:
 - а) список студентов заданного факультета;
 - б) списки студентов для каждого факультета и курса; в) список студентов, родившихся после заданного года.
2. **Abiturient:** Фамилия, Имя, Отчество, Адрес, Оценки. Создать массив объектов. Вывести:
 - а) список абитуриентов, имеющих неудовлетворительные оценки; б) список абитуриентов, сумма баллов у которых не меньше за- данной;

- в) выбрать N абитуриентов, имеющих самую высокую сумму баллов, и список абитуриентов, имеющих полупроходной балл.
3. **Aeroflot:** Пункт назначения, Номер рейса, Тип самолета, Время вылета, Дни недели. Создать массив объектов. Вывести:
- а) список рейсов для заданного пункта назначения;
 - б) список рейсов для заданного дня недели;
 - в) список рейсов для заданного дня недели, время вылета для которых больше заданного.
4. **Book:** Автор, Название, Издательство, Год, Количество страниц. Создать массив объектов. Вывести:
- а) список книг заданного автора;
 - б) список книг, выпущенных заданным издательством; в) список книг, выпущенных после заданного года.
5. **Worker:** Фамилия и инициалы, Должность, Год поступления на работу, Зарплата. Создать массив объектов. Вывести:
- а) список работников, стаж работы которых на данном предприятии превышает заданное число лет;
 - б) список работников, зарплата которых больше заданной;
 - в) список работников, занимающих заданную должность.
6. **Train:** Пункт назначения, Номер поезда, Время отправления, Число общих мест, Купейных, Плацкартных. Создать массив объектов. Вывести:
- а) список поездов, следующих до заданного пункта назначения;
 - б) список поездов, следующих до заданного пункта назначения и отправляющихся после заданного часа;
 - в) список поездов, отправляющихся до заданного пункта назначения и имеющих общие места.
7. **Product:** Наименование, Производитель, Цена, Срок хранения, Количество. Создать массив объектов. Вывести:
- а) список товаров для заданного наименования;
 - б) список товаров для заданного наименования, цена которых не превышает указанной;
 - в) список товаров, срок хранения которых больше заданного.
8. **Patient:** Фамилия, Имя, Отчество, Адрес, Номер медицинской карты, Диагноз. Создать массив объектов. Вывести:
- а) список пациентов, имеющих данный диагноз;
 - б) список пациентов, номер медицинской карты которых находится в заданном интервале.
9. **Bus:** Фамилия и инициалы водителя, Номер автобуса, Номер маршрута, Марка, Год начала эксплуатации, Пробег. Создать массив объектов. Вывести:
- а) список автобусов для заданного номера маршрута;
 - б) список автобусов, которые эксплуатируются больше 10 лет; в) список автобусов, пробег у которых больше 10 000 км.
10. **Customer:** Фамилия, Имя, Отчество, Адрес, Телефон, Номер кредитной карточки, Номер банковского счета. Создать массив объектов. Вывести:
- а) список покупателей в алфавитном порядке;
 - б) список покупателей, номер кредитной карточки которых находится в заданном интервале.
11. **File:** Имя файла, Размер, Дата создания, Количество обращений. Создать массив объектов. Вывести:
- а) список файлов, упорядоченный в алфавитном порядке;
 - б) список файлов, размер которых превышает заданный;
 - в) список файлов, число обращений к которым превышает заданное.
12. **Word:** Слово, Номера страниц, на которых слово встречается (от 1 до 10), Число страниц. Создать массив объектов. Вывести:
- а) слова, которые встречаются более чем на N страницах;
 - б) слова в алфавитном порядке;
 - в) для заданного слова номера страниц, на которых оно встречается.
13. **House:** Адрес, Этаж, Количество комнат, Площадь. Создать массив объектов. Вывести:
- а) список квартир, имеющих заданное число комнат;
 - б) список квартир, имеющих заданное число комнат и расположенных на этаже, который находится в определенном промежутке;
 - в) список квартир, имеющих площадь, превосходящую заданную.
14. **Phone:** Фамилия, Имя, Отчество, Адрес, Номер, Время внутригородских разговоров, Время междугородних разговоров. Создать массив объектов. Вывести:
- а) сведения об абонентах, время внутригородских разговоров которых превышает заданное;
 - б) сведения об абонентах, воспользовавшихся междугородней связью;
 - в) сведения об абонентах, выведенные в алфавитном порядке.
14. **Person:** Фамилия, Имя, Отчество, Адрес, Пол, Образование, Год рождения. Создать массив объектов. Вывести:
- а) список граждан, возраст которых превышает заданный; б) список граждан с высшим образованием;
 - в) список граждан мужского пола.

Требования к отчёту по лабораторной работе

Отчёт должен содержать:

1. Титульный лист
2. Вариант задания
3. UML-диаграмму и листинг программы
4. Вывод