

ЛАБОРАТОРНАЯ РАБОТА №1.

Структуры C++.

Цель работы: Изучение языковых конструкций C++, предназначенных для работы с агрегатными (структурными) типами данных.

Теоретические сведения.

Язык C++ предоставляет возможности для создания пользовательских типов данных, которые в дальнейшем могут использоваться наряду со встроенными типами (int, float, char и т.д.). Простейшим пользовательским типом данных в C++ является **структура**.

Структура — агрегатный тип данных, способный содержать разнотипные элементы (целочисленные, вещественные, символьные и другие в любых комбинациях). Основное назначение программных структур — упорядочение данных в оперативной памяти и обеспечение удобного доступа к этим данным.

1.1) Определение структуры

Перед первым использованием структура должна быть определена. Синтаксис определения имеет вид

```
struct name
{
    type1 field1;
    type2 field2;
    type3 field3;
    ...
};
```

Здесь struct — ключевое слово, которое начинает определение структуры, name — имя структуры, type1 — тип данных 1-го элемента структуры, field1 — имя 1-го элемента структуры и т.д. Элементы структуры (field1, field2, field3, ...) часто называют **полями**. В качестве типа данных type может выступить любой стандартный тип (int, float, double, char, bool). Кроме того, элементами структуры могут быть массивы, строки, указатели и другие структуры.

Имена структуры и имена полей выбираются программистом самостоятельно. Хорошей практикой является использование «говорящих» имен, когда имя структуры и имена всех полей отражают их содержание. Такая практика повышает читаемость программного кода, улучшает возможности его понимания и упрощает поиск ошибок.

Рассмотрим далее фрагмент кода

```
struct book
{
    char title[50];
    char authors[50];
    char publisher[25];
    int year;
    unsigned int pages;
};
```

Этот фрагмент содержит определение структуры с именем `book`, содержащей информацию о некоторой книге. Согласно приведенному выше листингу, структура `book` включает в себя пять полей.

1. Поле `title` – текстовая строка (тип данных – `char [50]`), содержащая до 49 текстовых символов. Значение 49 определяется тем, что последний элемент строки в языке C должен содержать 0 (нуль-терминированная строка). Поле `title` предназначено для хранения названия книги, например «Алиса в стране чудес».
2. Поле `authors` – также текстовая строка (тип данных – `char [50]`) для хранения данных об авторе/авторах. Условимся далее, что поле `authors` будет содержать фамилию автора и его инициалы, например «Толстой Л.Н.», «Бродский И.А.», «Цветаева М.И.» и т.д. Если авторов несколько, то они задаются через запятую. Пример: «Зельдович Я.Б., Яглом И.М.».
3. Поле `publisher` – текстовая строка (тип данных `char [25]`) с информацией об издательстве. Примеры: «КоЛибри», «Elsevier» и т.д.
4. Поле `year` хранит целочисленное значение (тип данных – `int`), означающее год издания книги.
5. Поле `pages` содержит объем книги в страницах (тип данных – `unsigned int`).

Фактически приведенный фрагмент кода означает создание пользователем своего собственного типа данных (в рассматриваемом случае – структурного). Обычно определения структур располагаются в начале программного модуля, сразу после директив препроцессора (`#include`, `#define` и др.).

1.2) Объявление структурной переменной

После определения структуры в программе может быть создана одна или несколько **структурных переменных**. Каждая из этих переменных должна иметь уникальное имя. В структурной переменной содержится информация о некоторой конкретной книге.

Объявление структурной переменной в C++ во многом аналогично объявлению переменной встроенного типа. Синтаксис объявления в общем виде

```
struct_name struct_var;
```

Здесь `struct_name` – имя структурного типа (в приведенном выше примере – `book`), `struct_var` – имя структурной переменной. Имя переменной может быть любым со стандартными ограничениями, касающимися всех имен в C++.

Отметим одно из различий языков C и C++ в части синтаксиса структур: если в C обязательным является использование ключевого слова `struct` при объявлении структурной переменной, то в C++ это слово не используется. Это еще раз подчеркивает то, что структура C++ – это полноценный новый тип данных.

Примеры объявления структурных переменных:

```
book b1, b2, b3;    // объявляем переменные b1, b2 и b3 структурного типа
book shelf[10];     // массив из 10-ти структур типа book с именем shelf
book *pointer;      // указатель на структурную переменную
```

Так же, как и в случае с переменными встроенных типов, структурная переменная может быть объявлена в любом месте программы. Важно при этом, чтобы определение структуры было расположено до объявления любых переменных структурного типа.

С точки зрения компилятора, объявление структурной переменной приводит к выделению оперативной памяти необходимого размера для данной переменной. Если выделение памяти произошло успешно, выделенный участок «закрепляется» за структурной переменной, в том смысле, что доступ к нему возможен только по имени переменной.

Структурная переменная может быть **инициализирована** при объявлении. Это означает, что после выделения оперативной памяти под переменную она заполняется заданными значениями. Синтаксис объявления с инициализацией следующий

```
struct_name struct_var = { field1_value, field2_value, field3_value, ... };
```

При выполнении данного оператора создается переменная с именем `struct_var` структурного типа `struct_name`, причем полю `field1` присваивается значение `field1_value`, полю `field2` – значение `field2_value`, полю `field3` – значение `field3_value` и т.д. Описанная языковая конструкция используется в случаях, когда значения полей создаваемой структурной переменной известны уже на этапе компиляции. Пример

```
book favorite = { «Обломов», «Гончаров И.А.», «Астрель», 2012, 608 };
```

1.3) Использование структуры. Доступ к полям.

Содержимое некоторой структурной переменной может быть прочитано и/или изменено программистом на уровне отдельных полей. Поэлементный доступ осуществляется с помощью оператора точки «.». При этом указывается имя структурной переменной, далее знак точки и имя интересующего нас поля (`struct_var.field`). Примеры

```
b1.title = «Приключения Тома Сойера и Гекльберри Финна»;  
b2.title = «Мартин Иден»;  
b3.year = favorite.year;  
printf("%s", b1.pages);  
shelf[3].authors = «Ильф И.А., Петров Е.П.»;
```

Предполагается, что структурные переменные `b1`, `b2`, `b3`, `favorite` и `shelf` были объявлены ранее так, как указано выше. Из приведенных примеров видно, что конструкция `struct_var.field` может использоваться как справа от знака равенства (для чтения значения поля), так и слева от него (для изменения значения). Последний пример демонстрирует использование доступа к массиву структур `shelf`.

1.4) Пространства имен.

C++ во многих смыслах является наследником языка C и поддерживает большинство конструкций своего предшественника без изменений. Вместе с тем, в C++ введен ряд дополнений, призванных облегчить работу программиста. Одной из таких особенностей является поддержка пространств имен (namespaces). Конструкция namespace была введена в C++ для того, чтобы избежать конфликтов имен при использовании сторонних библиотек. Такие конфликты возникают при совпадении имени пользовательской переменной, константы или функции с именем переменной, константы или функции в подключаемом библиотечном модуле.

Язык C++ позволяет определить для каждой библиотеки свое пространство имен и таким образом избежать возможных конфликтов. К примеру, функции стандартной

библиотеки языка C (C Run-Time library, CRT) погружены в пространстве имен std. Эти функции могут использоваться в программе C++, если в начале программного модуля имеется директива using namespace std.

В C++ введен новый формат подключения заголовочных файлов стандартной библиотеки – расширение .h не указывается, а перед названием библиотеки добавляется латинская буква c. Например

```
#include <cstdio>
#include <cstdlib>
```

1.5) Примеры консольных приложений Visual C++.

Рассмотрим пример приложения, использующего описанную выше структуру book. В приложении создается одна переменная-структура с именем my_book. Поля структуры заполняются некоторыми данными. Далее содержимое всех полей выводится на экран.

===== ЛИСТИНГ 1.1 =====

```
#include "stdafx.h"
#include <clocale>
#include <cstdlib>
#include <cstring>

using namespace std;

struct book
{
    char title[50];
    char authors[50];
    char publisher[25];
    int year;
    unsigned int pages;
};

book my_book;

int _tmain(int argc, _TCHAR* argv[])
{
    setlocale(LC_ALL, "rus");

    strcpy_s(my_book.title, "Обратные тригонометрические функции");
    strcpy_s(my_book.authors, "Новоселов С.И.");
    strcpy_s(my_book.publisher, "Учпедгиз");
    my_book.year = 1956;
    my_book.pages = 125;

    printf("    ИНФОРМАЦИЯ О КНИГЕ\n");
    printf("Название: %s\n", my_book.title);
    printf("Автор(ы): %s\n", my_book.authors);
    printf("Издательство: %s\n", my_book.publisher);
    printf("Год      : %i\n", my_book.year);
    printf("Страниц : %i\n", my_book.pages);

    system("pause");
    return 0;
}
```

=====

Обратим внимание на следующие особенности приведенного примера:

- 1) Заголовочные файлы библиотек CRT (locale, stdlib, string) включены в формате C++; для доступа к библиотечным функциям из пространства имен std использована директива using.
- 2) Определение структуры book расположено в самом начале программного модуля, до момента объявления первой структурной переменной (в данном случае – переменной my_book).
- 3) Для изменения значений текстовых полей (title, authors, publisher) используется функция strcpy_s из библиотеки string. Это «безопасный» (safe) аналог функции strcpy. Отметим, что использование оператора присваивания, например my_book.publisher = “Учпедгиз”, в данном случае приведет к утечке оперативной памяти (объясните почему).

Следующий пример демонстрирует работу с массивами структур. Обратите внимание на то, как объявляется массив из N структур с именем collection и как в дальнейшем происходит доступ к отдельным его элементам.

===== ЛИСТИНГ 1.2 =====

```
#include "stdafx.h"
#include <locale>
#include <stdlib>
#include <cstring>
#include <Windows.h>

using namespace std;

struct book
{
    char title[50];
    char authors[50];
    char publisher[25];
    int year;
    unsigned int pages;
};

const int N = 2;
book collection[N];           // создаем массив из N структур book

int _tmain(int argc, _TCHAR* argv[])
{
    setlocale(LC_ALL, "rus");

    for (int i = 0; i < N; i++)
    {
        printf("\n\nВведите данные для книги №%d\n", i + 1);
        printf("  Название - ");
        SetConsoleCP(1251);
        gets_s(collection[i].title, 49);
        SetConsoleCP(866);

        printf("  Автор(ы) - ");
        SetConsoleCP(1251);
        gets_s(collection[i].authors, 49);
        SetConsoleCP(866);

        printf("  Издательство - ");
        SetConsoleCP(1251);
```

```

        gets_s(collection[i].publisher, 49);
        SetConsoleCP(866);

        printf("  Год выпуска - ");
        scanf_s("%d", &collection[i].year);
        getchar();

        printf("  Число страниц - ");
        scanf_s("%u", &collection[i].pages);
        getchar();
    }

    char find_title[50];
    printf("\n\n Введите название искомой книги - ");
    SetConsoleCP(1251);
    gets_s(find_title, 49);
    SetConsoleCP(866);

    bool found = false;
    for (int i = 0; i < N; i++)
    {
        if (strcmp(collection[i].title, find_title) == 0)
        {
            found = true;
            printf("\n===== НАЙДЕНА КНИГА =====\n");
            printf("  Название:      ");
            puts(collection[i].title);
            printf("  Автор(ы):      ");
            puts(collection[i].authors);
            printf("  Издательство:  ");
            puts(collection[i].publisher);
            printf("  Год выпуска:   %d\n", collection[i].year);
            printf("  Страниц:       %u\n", collection[i].pages);
        }
    }
    if (!found)
        printf(" Книга с таким названием не найдена в коллекции!\n");

    system("pause");
    return 0;
}
=====

```

Этот пример также демонстрирует использование:

1. Операторов scanf, printf, gets_s, puts для ввода/вывода данных (подробное описание – см. доп. литературу и справочные ресурсы Интернет).
2. Функции SetConsoleCP из библиотеки Windows.h> для изменения кодовой страницы. Установки кодовых страниц позволяют корректно вводить текстовые данные на русском языке.
3. Функции strcmp из библиотеки <cstring> для сравнения текстовых строк.
4. Логической переменной (флага) found для индикации успешного поиска.

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ №1.

- 1) Спроектируйте программную структуру для своего варианта. Обоснуйте выбор типа данных для каждого поля структуры.
- 2) Реализуйте консольное приложение, использующее спроектированную структуру. Внутри приложения: а) определите новый структурный тип; б) создайте массив из нескольких (5-10) структур; в) заполните структуры поэлементно, используя ввод с

клавиатуры или текстового файла; г) выведите на экран содержимое всего массива; д) реализуйте в программе дополнительные функции, указанные в задании для лабораторной работы (см. свой вариант).

Процедура сдачи лабораторных работ

- 1) Получите свой вариант задания на лабораторную работу у преподавателя.
- 2) Ознакомьтесь с описанием лабораторной работы. Изучите необходимые языковые конструкции, разберите примеры. При необходимости обратитесь к лекционным материалам, дополнительной литературе и справочным ресурсам в локальной сети кафедры ИСКМ и в сети Интернет.
- 3) Спроектируйте программу в соответствии с полученным заданием. В качестве основы разрешается использовать примеры, приведенные в данном методическом пособии. Вместе с тем, творческая инициатива студента (например, дополнительная функциональность программы, улучшенная структура, более удобный интерфейс и т.д.) также всячески приветствуется.
- 4) Наберите код приложения в выбранной среде программирования, например Visual Studio. Выполните трансляцию приложения, при необходимости исправьте синтаксические ошибки в коде (ошибки времени компиляции).
- 5) Проверьте работу приложения для различных входных данных. В случае некорректной работы используйте пошаговый режим выполнения программы и встроенный отладчик (меню `debug` в среде Visual C++) для поиска ошибок времени исполнения.
- 6) При возникновении существенных затруднений на любом из описанных выше этапов обратитесь за помощью к преподавателю.
- 7) Сдайте проверенную программу преподавателю, предварительно оформив в тетради шапку отчета. Во время сдачи программы студенту необходимо
 - показать на примерах корректность работы программы, т.е. их соответствие заданию;
 - при необходимости кратко сформулировать алгоритм работы программы и показать, как этот алгоритм реализован в конкретной программе;
 - знать и при необходимости объяснить назначение и принцип работы всех используемых им языковых конструкций;
 - понимать назначение и смысл всех использованных в программе переменных, констант, функций, классов и т.д.При успешной сдаче этой части лабораторной работы преподаватель выставляет свою подпись и дату в поле «Программа».
- 8) Полностью оформите отчет в тетради (см. пример). Подготовьтесь к ответу на контрольные вопросы.
- 9) Сдайте отчет по лабораторной работе преподавателю. Преподаватель проверяет: полноту и правильность оформления отчета, корректность записи кода и результатов работы программы. Теоретические знания студента проверяются с помощью контрольных вопросов (выборочно).
- 10) Получите подпись преподавателя в поле «Отчет». Лабораторная работа считается сданной.

ВАРИАНТЫ ЗАДАНИЙ К ЛАБОРАТОРНОЙ РАБОТЕ

№	Описание
1	<p><i>Предметная область:</i> Кадры предприятия</p> <p><i>Структурный тип данных:</i> Сотрудник предприятия (employee)</p> <p><i>Элементы структуры:</i> Фамилия (surname) Имя (name) Отчество (patronymic) Дата рождения (bdate) Отдел (division) Должность (position)</p> <p><i>Задание:</i> 1) Вывести на экран ФИО и должности всех сотрудников, работающих в заданном отделе (номер отдела вводится с клавиатуры). 2) Вывести на экран данные обо всех сотрудниках старше 50 лет.</p>
2	<p><i>Предметная область:</i> Рынок недвижимости</p> <p><i>Структурный тип данных:</i> Квартира в многоквартирном доме (flat)</p> <p><i>Элементы структуры:</i> Число комнат (rooms) Общая площадь (total) Жилая площадь (living) Число балконов/лоджий (balconies) Этаж (floor) Этажность дома (storeys) Район (district)</p> <p><i>Задание:</i> 1) Рассчитать общую площадь квартир, расположенных в заданном районе (название района вводится с клавиатуры) 2) Вывести на экран данные обо всех квартирах, расположенных на последнем этаже многоквартирного дома (5 и более этажей).</p>
3	<p><i>Предметная область:</i> Магазин компьютерной техники</p> <p><i>Структурный тип данных:</i> Персональный компьютер (computer)</p> <p><i>Элементы структуры:</i> Марка процессора (cpu_brand) Тактовая частота процессора (cpu_clock) Марка материнской платы (motherboard) Марка жесткого диска (harddrive) Марка видеокарты (videocard) Наличие модуля wifi (wifi) Цена в рублях (price) Количество на складе (store)</p> <p><i>Задание:</i> 1) Рассчитать общую стоимость всех компьютеров, имеющихся на складе. 2) Вывести на экран информацию о компьютере с самой высокой тактовой частотой в ценовом диапазоне от 20000 руб. до 30000 руб.</p>

4	<p><i>Предметная область:</i> Пассажирские железнодорожные перевозки</p> <p><i>Структурный тип данных:</i> Железнодорожный билет (ticket)</p> <p><i>Элементы структуры:</i> Номер рейса (train) Станция отправления (station1) Станция прибытия (station2) Дата отправления (dep_day) Время отправления (dep_time) Вагон (coach) Место (seat) Стоимость (price)</p> <p><i>Задание:</i> 1) Вывести на экран информацию о самом дорогом из всех проданных билетов. 2) Вывести на экран количество свободных мест в заданном вагоне заданного поезда (номер рейса, дата отправления и номер вагона задаются с клавиатуры).</p>
5	<p><i>Предметная область:</i> Техническое обслуживание (ТО) автомобилей</p> <p><i>Структурный тип данных:</i> Транспортное средство (vehicle)</p> <p><i>Элементы структуры:</i> ФИО владельца (owner) Регистрационный номер (reg_number) Тип автомобиля (type) Марка автомобиля (brand) Объем двигателя (volume) Мощность двигателя (power) Дата последнего ТО (last_to) Пробег (mileage)</p> <p><i>Задание:</i> 1) Вывести на экран регистрационные номера всех автомобилей с пробегом свыше 100000 км. 2) Вывести на экран фамилии владельцев всех автомобилей, прошедших ТО в течение последнего месяца.</p>

6	<p><i>Предметная область:</i> Растровая компьютерная графика</p> <p><i>Структурный тип данных:</i> Файл изображения (file)</p> <p><i>Элементы структуры:</i> Имя файла (name) Размер файла в байтах (size) Формат данных (format) Ширина в пикселях (width) Высота в пикселях (height) Разрешение (dpi) Глубина цвета (depth)</p> <p><i>Задание:</i> 1) Вывести на экран название и формат всех графических файлов, имя которых начинается на заданную букву (другой вариант – заканчивается заданной буквой). Буква вводится с клавиатуры. 2) Найти самое большое по занимаемой площади изображение (учитываются поля width, height, dpi).</p>
7	<p><i>Предметная область:</i> Домашняя DVD-фильмотека</p> <p><i>Структурный тип данных:</i> Диск формата DVD/Blu-ray (disk)</p> <p><i>Элементы структуры:</i> Название фильма (title) Режиссер (director) Жанр фильма (genre) Длительность в мин. (duration) Год выпуска (year) Киноккомпания (company) Рейтинг IMDB (imdb)</p> <p><i>Задание:</i> 1) Найти самый поздний фильм заданного режиссера. Фамилия режиссера вводится с клавиатуры. 2) Вывести на экран все фильмы в жанре комедии с рейтингом IMBD выше 8.0.</p>

8	<p><i>Предметная область:</i> Компьютерные игры</p> <p><i>Структурный тип данных:</i> Компьютерная игра (game)</p> <p><i>Элементы структуры:</i> Название (title) Жанр (genre) Платформа (platform) Год выпуска (year) Компания-разработчик (developer) Число участников (players) Мин. частота процессора в МГц (min_cpu) Мин. объем опер. памяти в МБ (min_ram) Мин. объем места на диске в МБ (min_hdd)</p> <p><i>Задание:</i> 1) Найти компьютерную игру заданного жанра с самыми низкими требованиями по объему используемой оперативной памяти. Жанр игры задается с клавиатуры. 2) Вывести на экран названия всех компьютерных игр, которые могут быть установлены на заданном компьютере. Характеристики компьютера – частота процессора, объемы памяти RAM и HDD задаются с клавиатуры.</p>
9	<p><i>Предметная область:</i> Рынок автотранспортных средств</p> <p><i>Структурный тип данных:</i> Автомобиль (car)</p> <p><i>Элементы структуры:</i> Марка автомобиля (brand) Название модели (model) Название комплектации (complect) Рыночный сегмент (segment) Тип автомобильного кузова (body) Объем двигателя в литрах (volume) Мощность двигателя в л.с. (power) Год выпуска (year) Стоимость в рублях (price)</p> <p><i>Задание:</i> 1) Рассчитать и вывести на экран отношение мощности двигателя к стоимости автомобиля для всех представленных моделей и комплектаций. Найти модель с самым высоким отношением. 2) Вывести на экран названия всех комплектаций заданной модели автомобиля в порядке возрастания их стоимости. Название модели вводится с клавиатуры.</p>

10	<p><i>Предметная область:</i> Политическая география</p> <p><i>Структурный тип данных:</i> Суверенное государство (state)</p> <p><i>Элементы структуры:</i> Название государства (country) Столица (capital) Форма правления (government) Язык(и) (language) Религия (religion) Площадь территории (area) Численность населения (population) Континент (continent)</p> <p><i>Задание:</i> 1) Рассчитать суммарную площадь и население государств Северной Америки. 2) Вывести на экран название и столицу самого крупного по численности населения испано-язычного государства.</p>
11	<p><i>Предметная область:</i> Спортивные достижения человечества</p> <p><i>Структурный тип данных:</i> Спортивный рекорд (record)</p> <p><i>Элементы структуры:</i> Вид спорта (sport) Дисциплина (discipline) Вид рекорда (мировой/европейский/...) (type) Мужской/женский (gender) Дата установления (date) Фамилия рекордсмена (name) Страна (country) Значение рекорда (achievement)</p> <p><i>Задание:</i> 1) Рассчитать общее количество рекордов разного уровня для заданного вида спорта. Вид спорта вводится с клавиатуры. 2) Вывести на экран все мировые рекорды, установленные женщинами в течение последнего года.</p>

12	<p><i>Предметная область:</i> Валютный рынок</p> <p><i>Структурный тип данных:</i> Текущий курс иностранной валюты (course)</p> <p><i>Элементы структуры:</i> Денежная единица (currency) Государство (state) Код ISO4217 (code) Дробная единица (subunit) Величина дробления (fraction) Текущий курс в рублях (rate)</p> <p><i>Задание:</i> 1) Вывести на экран текущие значения курсов всех иностранных валют в формате «денежная единица – курс в рублях». 2) Вывести на экран информацию обо всех денежных единицах, курс которых на текущий момент превышает курс американского доллара.</p>
13	<p><i>Предметная область:</i> Глобальная сеть Интернет</p> <p><i>Структурный тип данных:</i> Интернет-ресурс, сайт (site)</p> <p><i>Элементы структуры:</i> Название ресурса (name) Тип ресурса - поисковый, информационный, ... (type) Доменное имя (domain) Среднее число пользователей в день (users) Число веб-сайтов, ссылающихся на данный ресурс (linked) Среднее время загрузки веб-страницы (time) Используемые веб-технологии (technologies)</p> <p><i>Задание:</i> 1) Найдите и выведите на экран доменное имя самого быстрого поискового интернет-сервиса (т.е. с самой высокой скоростью загрузки страницы). 2) Выведите на экран информацию обо всех интернет-ресурсах, у которых среднее число пользователей в день превышает заданную величину (вводится с клавиатуры).</p>

14	<p><i>Предметная область:</i> Высшее профессиональное образование</p> <p><i>Структурный тип данных:</i> Итоговая оценка студента по дисциплине (rating)</p> <p><i>Элементы структуры:</i> ФИО студента (name) Номер зачетной книжки (number) Специальность (speciality) Группа (group) Название дисциплины (course) Номер семестра (semester) Итоговая оценка по 100-балльной шкале (mark)</p> <p><i>Задание:</i> 1) Выведите на экран ФИО всех студентов, получивших неудовлетворительные оценки (менее 60 баллов) в заданном семестре. Номер семестра вводится с клавиатуры. 2) Выберите студента с самой высокой итоговой оценкой по дисциплине.</p>
15	<p><i>Предметная область:</i> Авиаперевозки</p> <p><i>Структурный тип данных:</i> Авиарейс (flight)</p> <p><i>Элементы структуры:</i> Номер рейса (number) Авиакомпания (airline) Откуда (from) Куда (to) Протяженность в км (distance) Длительность полета (duration) Тип самолета (aircraft) Число посадочных мест (passengers)</p> <p><i>Задание:</i> 1) Найдите авиарейс с самой высокой средней скоростью полета. Выведите на экран тип используемого самолета. 2) Рассчитайте общую протяженность всех авиарейсов, выполняемых из аэропорта Москвы.</p>

16	<p><i>Предметная область:</i> Учет больных в поликлинике по месту жительства</p> <p><i>Структурный тип данных:</i> Обращение в поликлинику (visit)</p> <p><i>Элементы структуры:</i> ФИО пациента (patient) Номер мед. полиса (insurance) Номер мед. карты (card) ФИО врача (doctor) Дата обращения (date) Диагноз (diagnosis) Назначенное лечение (therapy)</p> <p><i>Задание:</i> 1) Найдите пациента в самой поздней датой обращения в поликлинику. 2) Рассчитайте общее количество пациентов, которые были приняты данным врачом за все время его работы. ФИО врача вводится с клавиатуры.</p>
17	<p><i>Предметная область:</i> Чемпионат страны по командному виду спорта</p> <p><i>Структурный тип данных:</i> Команда-участник (participant)</p> <p><i>Элементы структуры:</i> Название клуба (club) Город (city) ФИО тренера (coach) Дата основания клуба (date) Бюджет команды в руб. (budget) Текущее количество очков (points) Текущее место в чемпионате (place)</p> <p><i>Задание:</i> 1) Найдите команду с самым высоким отношением количества набранных очков к объему потраченных денег (бюджету). 2) Выведите на экран названия всех команд, выступающих за выбранный город. Название города вводится с клавиатуры.</p>
18	<p><i>Предметная область:</i> Периодическая система химических элементов</p> <p><i>Структурный тип данных:</i> Химический элемент (element)</p> <p><i>Элементы структуры:</i> Название элемента (name) Символ (symbol) Номер в таблице (number) Тип (type) Атомная масса (mass) Электронная конфигурация (electrons)</p> <p><i>Задание:</i> 1) Найдите химический элемент по его атомной массе. 2) Выведите на экран названия всех имеющихся в массиве химических элементов - металлов.</p>