**СТРУКТУРИ**

1. **Описание на структурата**

**Съставен тип данни, включващ краен брой елементи,**

**които могат да бъдат и от различен тип, се нарича СТРУКТУРА**

**Структурата се свързва с точно определено име. Всеки елемент от структурата се нарича поле и достъпът до него се осъществява посредством име на поле**

1. **Обща форма на структура**

**struct <име на структура> /\*struct-запазана дума, име на структура- идентифик. зададен от програмиста\*/**

**{**

**<тип на поле 1><име на поле 1>; /\*идентифик. на променливи от произволен тип\*/**

**<тип на поле 2><име на поле 2>;**

**……………………..**

**<тип на поле N><име на поле N>;**

**};**

**Дефинирането на структура не води до заделяне на място в паметта. Това става с деклариране на променлива от потребителския тип.**

**Пример1: Да се дефинира структура Books, която да съдържа следните елементи: title от тип char, author от тип char, subject от тип char, и book id от тип int. Като се даде достъп до членовете на структурата, чрез използване на оператора (.) да се декларират две променливи от структура Books – Book1 и Book2, на които да се дефинират съответни спецификации, и които спецификации да се отпечатат на монитора:**

**Решение:**

**#include <iostream>**

**#include <cstring>**

**using namespace std;**

**struct Books //създаване на структура Books**

**{**

**char title[50];**

**char author[50];**

**char subject[100];**

**int book\_id;**

**};**

**int main( )**

**{**

**struct Books Book1; // деклариране на Book1 от структура Books**

**struct Books Book2; // деклариране на Book2 от структура Books**

**// Book 1 спецификация**

**strcpy( Book1.title, "Learn C++ Programming");**

**strcpy( Book1.author, "Chand Miyan");**

**strcpy( Book1.subject, "C++ Programming");**

**Book1.book\_id = 6495407;**

**// Book 2 спецификация**

**strcpy( Book2.title, "Telecom Billing");**

**strcpy( Book2.author, "Yakit Singha");**

**strcpy( Book2.subject, "Telecom");**

**Book2.book\_id = 6495700;**

**// отпечатване на Book1 стойностите**

**cout << "Book 1 title : " << Book1.title <<endl;**

**cout << "Book 1 author : " << Book1.author <<endl;**

**cout << "Book 1 subject : " << Book1.subject <<endl;**

**cout << "Book 1 id : " << Book1.book\_id <<endl;**

**// отпечатване на Book2 стойностите**

**cout << "Book 2 title : " << Book2.title <<endl;**

**cout << "Book 2 author : " << Book2.author <<endl;**

**cout << "Book 2 subject : " << Book2.subject <<endl;**

**cout << "Book 2 id : " << Book2.book\_id <<endl;**

**return 0;**

**}**

**Резултат:**

**Book 1 title : Learn C++ Programming**

**Book 1 author : Chand Miyan**

**Book 1 subject : C++ Programming**

**Book 1 id : 6495407**

**Book 2 title : Telecom Billing**

**Book 2 author : Yakit Singha**

**Book 2 subject : Telecom**

**Book 2 id : 6495700**

**За да решим задачата използвахме функцията strcpy.**

**Тази функция копира C низа посочен от източника в масива посочен от местоназначението .**

**Ето и един кратък пример как работи тази функция:**

**#include<iostream>**

**#include <stdio.h>**

**#include <string.h>**

**using namespace std;**

**int main ()**

**{**

**char str1[]="Sample string"; //дефинираме първата променлива**

**char str2[40]; //декларираме втората променлива**

**char str3[40]; //декларираме третата променлива**

**strcpy (str2,str1); //str2 се копира в str1/**

**strcpy (str3,"copy successful"); // в str3 се копира низа "copy successful"**

**cout<<str1<<endl; //отпечатва 3-те низа**

**cout<<str2<<endl;**

**cout<<str3<<endl;**

**return 0;**

**}**

**Резултат:**

**Sample string**

**Sample string**

**copy successful**

**СТРУКТУРА КАТО АРГУМЕНТ НА ФУНКЦИЯ**

**#include <iostream>**

**#include <cstring>**

**using namespace std;**

**/\*създаваме функция printBook където структурата Books се явява неин аргумент book и затова е поставена в скобите\*/**

**void printBook( struct Books book );**

**struct Books**

**{**

**char title[50];**

**char author[50];**

**char subject[100];**

**int book\_id;**

**};**

**int main( )**

**{**

**struct Books Book1; // деклариране на Book1 от тип Book**

**struct Books Book2; // деклариране на Book2 от тип Book**

**// Book 1 спецификация**

**strcpy( Book1.title, "Learn C++ Programming");**

**strcpy( Book1.author, "Chand Miyan");**

**strcpy( Book1.subject, "C++ Programming");**

**Book1.book\_id = 6495407;**

**// Book 2 спецификация**

**strcpy( Book2.title, "Telecom Billing");**

**strcpy( Book2.author, "Yakit Singha");**

**strcpy( Book2.subject, "Telecom");**

**Book2.book\_id = 6495700;**

**//отпечатване на спецификацията на Book1**

**printBook( Book1 );**

**// отпечатване на спецификацията на Book2**

**printBook( Book2 );**

**return 0;**

**}**

**void printBook( struct Books book )**

**{**

**cout << "Book title : " << book.title <<endl;**

**cout << "Book author : " << book.author <<endl;**

**cout << "Book subject : " << book.subject <<endl;**

**cout << "Book id : " << book.book\_id <<endl;**

**}**

**Резултат:**

**Book title : Learn C++ Programming**

**Book author : Chand Miyan**

**Book subject : C++ Programming**

**Book id : 6495407**

**Book title : Telecom Billing**

**Book author : Yakit Singha**

**Book subject : Telecom**

**Book id : 6495700**

**Указатели към структури**

**Може да дефинирате указатели към структури по подобен начин както дефинирате указател към всяка друга променлива, както е показано по-долу:**

**struct Books \*struct\_pointer;**

**Сега можете да съхраните адреса на променливата на структурата в показаната по-горе указател на променлива. За да намерите адреса на променливата на структурата сложете оператор „&” преди името на структурара:**

**struct\_pointer = &Book1;**

**За да имате достъп до членовете на структурата използващи указател, трябва да използвате оператора “** -> **“ :**

**struct\_pointer->title;**

**#include <iostream>**

**#include <cstring>**

**using namespace std;**

**/\*дефинира функция printBook където структурата struct Books използва указател \*book \*/**

**void printBook( struct Books \*book );**

**struct Books**

**{**

**char title[50];**

**char author[50];**

**char subject[100];**

**int book\_id;**

**};**

**int main( )**

**{**

**struct Books Book1; //декларира Book1 от тип Book**

**struct Books Book2; // декларира Book2 от тип Book**

**// Book 1 спецификация**

**strcpy( Book1.title, "Learn C++ Programming");**

**strcpy( Book1.author, "Chand Miyan");**

**strcpy( Book1.subject, "C++ Programming");**

**Book1.book\_id = 6495407;**

**// Book 2 спецификация**

**strcpy( Book2.title, "Telecom Billing");**

**strcpy( Book2.author, "Yakit Singha");**

**strcpy( Book2.subject, "Telecom");**

**Book2.book\_id = 6495700;**

**//отпечатва Book1 спецификация, подавайки адреса на структурата**

**printBook( &Book1 );**

**// отпечатва Book1 спецификация, подавайки адреса на структурата**

**printBook( &Book2 );**

**return 0;**

**}**

**// тази функция приема указател към структурата като параметър.**

**void printBook( struct Books \*book )**

**{**

**cout << "Book title : " << book->title <<endl;**

**cout << "Book author : " << book->author <<endl;**

**cout << "Book subject : " << book->subject <<endl;**

**cout << "Book id : " << book->book\_id <<endl;**

**}**

**Резултат:**

**Book author : Chand Miyan**

**Book subject : C++ Programming**

**Book id : 6495407**

**Book title : Telecom Billing**

**Book author : Yakit Singha**

**Book subject : Telecom**

**Book id : 6495700**

**ДОПЪЛНИТЕЛНИ ЗАДАЧИ**

**Зад.1 : Да се напише програма, в която се дефинира структура с име Student, задаваща ученик с полета за трите имена, ЕГН и среден успех. Да се декларира променлива а от типа на дефинираната структура и се инициализират полетата и с подходящи стойности. Тези стойности да се изведат на отделни редове**

**#include <iostream>**

**using namespace std;**

**struct Student //дефинира структура Student**

**{**

**char first\_name[20]; //членове на структурата ↓**

**char second\_name[20];**

**char last\_name[20];**

**char egn[11];**

**double sreden\_uspeh;**

**};**

**int main()**

**{**

**Student a={"Petar","Ivanov","Dimitrov","9810147040",4.50};**

**/\* декларира променлива а от типа на структурата Student и се инициализират полетата и с подходящи ст-ти**

**Виж урока за Инициализация на символни низове\*/**

**cout<<a.first\_name<<endl;**

**cout<<a.second\_name<<endl;**

**cout<<a.last\_name<<endl;**

**cout<<a.egn<<endl;**

**cout<<a.sreden\_uspeh<<endl;**

**return 0;**

**}**

**Резултат:**

**Petar**

**Ivanov**

**Dimitrov**

**9810147040**

**4.5**

**Зад. 2 Да се напише програма, в която се дефинира структура с име Student, задаваща ученик с полета за трите имена, ЕГН и среден успех.**

**Да се декларира променлива от тип Student и за всяко поле да се въведат стойности от клавиатурата. Въведените стойности да се изведат на един ред, разделени с интервал.**

**#include <iostream>**

**using namespace std;**

**struct Student**

**{**

**char first\_name[20];**

**char second\_name[20];**

**char last\_name[20];**

**char egn[11];**

**double sreden\_uspeh;**

**};**

**int main()**

**{**

**Student a; // а- име на променливата от структура с име Student**

**cout<<"Ime:";**

**cin>>a.first\_name;**

**cout<<"Prezime:";**

**cin>>a.second\_name;**

**cout<<"Familia:";**

**cin>>a.last\_name;**

**cout<<"EGN:";**

**cin>>a.egn;**

**cout<<"sreden uspeh:";**

**cin>>a.sreden\_uspeh;**

**cout<<a.first\_name<<" ";**

**cout<<a.second\_name<<" ";**

**cout<<a.last\_name<<" ";**

**cout<<a.egn<<" ";**

**cout<<a.sreden\_uspeh<<" ";**

**return 0;**

**}**

**Резултат:**

**Ime:petar**

**Prezime:ivanov**

**Familia:vasiliv**

**EGN:9003156040**

**sreden uspeh:5.60**

**petar ivanov vasiliv 9003156040 5.6**

**Зад.3 Да се напише програма, която въвежда от клавиатурата имената, ЕГН и средния успех на всички ученици от един клас. Да се изведат на екрана данните само за учениците с успех, по-висок или равен на 5.50. За всеки ученик данните да се извеждат на нов ред, разделени с интервал.**

**#include <iostream>**

**using namespace std;**

**struct Student**

**{**

**char first\_name[20];**

**char second\_name[20];**

**char last\_name[20];**

**char egn[11];**

**double sreden\_uspeh;**

**};**

**int main()**

**{**

**Student students[35];**

**int n,i;**

**cout<<"Broi uchenici:";**

**cin>>n;**

**for(i=0;i<n;i++)**

**{**

**cout<<"Ime:";**

**cin>>students[i].first\_name;**

**cout<<"Prezime:";**

**cin>>students[i].second\_name;**

**cout<<"Familia:";**

**cin>>students[i].last\_name;**

**cout<<"EGN:";**

**cin>>students[i].egn;**

**cout<<"sreden uspeh:";**

**cin>>students[i].sreden\_uspeh;**

**}**

**for(i=0;i<n;i++)**

**if(students[i].sreden\_uspeh>=5.50)**

**{**

**cout<<students[i].first\_name<<" ";**

**cout<<students[i].second\_name<<" ";**

**cout<<students[i].last\_name<<" ";**

**cout<<students[i].egn<<" ";**

**cout<<students[i].sreden\_uspeh<<endl;**

**}**

**return 0;**

**}**

**Резултат:**

**aaa aaa aaa 111 5.6**

**eee eee eee 333 6**

**Зад. 4 Да се напише програма, която създава структура Firma с полета name, egn, duty и payment, указващи името на работник във фирмата, неговото ЕГН, длъжност и заплата. Да се въведе цяло число n и след него n на брой данни от тип Firma. Да се изведат имената на тези работници, чиято заплата е по-малка от 700 лева.**

**#include <iostream>**

**using namespace std;**

**struct Firma**

**{**

**char name[31];**

**char egn[11];**

**char duty[21];**

**double payment;**

**};**

**int main()**

**{**

**int n;**

**Firma workers[21];**

**cin>>n;**

**for(int i=0;i<n;i++)**

**{**

**cin.get();**

**cin.getline(workers[i].name,31);**

**cin>>workers[i].egn;**

**cin>>workers[i].duty;**

**cin>>workers[i].payment;**

**}**

**for(int j=0;j<n;j++)**

**{**

**if(workers[j].payment<700)**

**{**

**cout<<workers[j].name<<endl;**

**}**

**}**

**return 0;**

**}**

**Обяснения:**

**Функцията cin.getline() се използва, когато се налага да се въвеждат низове, съдържащи интервали. По подразбиране (ако не е указан специален символ) въвеждането спира при достигане на символ за нов ред, като той не се записва в низа. А когато се въвеждат данни с оператора cin>>, в буфера остава знак за нов ред.**

**При следващото въвеждане със същия оператор този знак се прескача. Но ако след въвеждане посредством cin>> се налага въвеждане на низ чрез cin.getline(), в който не е указан символ за край на въвеждането, то не се въвежда нищо, защото по подразбираневъвеждането спира при срещане на символ за нов ред, който е останал в буфера на клавиатурата от предното въвеждане чрез cin>>.**

**Това налага, преди да се въведе низът, да се вземе символът за нов ред.**

**Това може да стане, като се ползва специална функция, която възприема само един символ от буфера на клавиатурата – cin.get().**

**Зад.5 Да се напише програма, която създава структура Hospital с полета name, diagnose и days, указващи името на пациент и болница, диагнозата му и брой дни, прекарани в болницата. Да се въведе цяло число n и след него n на брой данни от тип Hospital. Да се изведат**

**I /**

**#include <iostream>**

**using namespace std;**

**/\***

**\***

**\*/**

**struct Hospital**

**{**

**string name;**

**string diagnose;**

**int days;**

**};**

**int main(int argc, char\*\* argv) {**

**string bolest;**

**int broiPacienti;**

**cout << "Broq na pacientite: "<< endl;**

**cin >> broiPacienti;**

**Hospital pacienti [broiPacienti];**

**for(int i = 0 ; i < broiPacienti; i++){**

**cout << "Ime na pacienta"<<endl;**

**cin >> pacienti[i].name;**

**cout << "Bolesta na pacienta"<<endl;**

**cin >> pacienti[i].diagnose;**

**cout << "Prestoi na pacienta"<<endl;**

**cin >> pacienti[i].days;**

**cout << endl;**

**}**

**Hospital prestoi = pacienti[0];**

**cout << "Vkaraite bolest"<<endl;**

**cin >> bolest;**

**for(int i = 0 ; i < broiPacienti;i++){**

**if(pacienti[i].days > prestoi.days){**

**prestoi = pacienti[i];**

**}**

**if(pacienti[i].diagnose == bolest)**

**{**

**cout << "Horata koito imat bolesta sa: ";**

**cout << pacienti[i].name << endl;**

**}**

**}**

**cout << "Dannite za choveka s nai mnogo dni prestoi e: "<<endl;**

**cout << prestoi.name <<" "<< prestoi.diagnose <<" "<< prestoi.days;**

**return 0;**

**}**

**#include <iostream>**

**#include<string>**

**#include<cstring>**

**using namespace std;**

**struct Hospital**

**{**

**char name[31];**

**char diagnose[31];**

**int days;**

**};**

**int main()**

**{**

**int n;**

**cout<<"broi pacienti";**

**cin>>n;**

**Hospital patients[50];**

**for(int i=0;i<n;i++)**

**{**

**cin.get(); //възприема само 1 символ от буфера на клавиатурата**

**cout<<"Patient name";**

**cin.getline(patients[i].name,31);**

**cout<<"patient diagnose";**

**cin>>patients[i].diagnose;**

**cout<<"patient days at the hospital";**

**cin>>patients[i].days;**

**}**

**int ind=0,max=patients[0].days;**

**char FindDiagnose[31];**

**cout<<"napishi diagnoza";**

**cin>>FindDiagnose;**

**for(int j=0;j<n;j++)**

**{**

**if(strcmp(patients[j].diagnose,FindDiagnose)==0) //compare string**

**cout<<patients[j].name<<endl;**

**if(max<patients[j].days)**

**{**

**max=patients[j].days;**

**ind=j;**

**}**

**}**

**cout<<patients[ind].name<<" ";**

**cout<<patients[ind].diagnose<<" ";**

**cout<<patients[ind].days<<" ";**

**return 0;**

**}**

**Командата strcmp сравнява два низа. Ето пример за това как се използва:**

#include <iostream>

#include <cstring>

using namespace std;

int main ()

{

char key[] = "apple";

char buffer[80];

do {

cout<<"Guess my favorite fruit? ";

cin>>buffer;

} while (**strcmp** (key,buffer) != 0);

cout<<"Correct answer!";

return 0;

}

**Зад. 6 Да се напише програма, която създава структура People с полета first\_name, last\_name и egn, указващи името, фамилията и ЕГН на човек. Да се въведе цяло число n и след него n на брой данни от тип People. Да се изведат данните, сортирани по първото име на човека.**

**#include <iostream>**

**#include <cstring>**

**#include<string>**

**using namespace std;**

**struct People**

**{**

**char first\_name[16];**

**char last\_name[16];**

**char egn[11];**

**};**

**int main ()**

**{**

**int n;**

**People peoples[50];**

**int arr[51],i,j,swap;**

**cin>>n;**

**for(i=0;i<n;i++)**

**{**

**cin>>peoples[i].first\_name;**

**cin>>peoples[i].last\_name;**

**cin>>peoples[i].egn;**

**arr[i]=i; //запазване индексите на масива peoples в масива arr**

**}**

**/\*сортиране на елементите на масива по метода на мехурчето, като се сортират индексите на масива arr\*/**

**for(i=0;i<n-1;i++)**

**for(j=0;j<n-i-1;j++)**

**if(strcmp(peoples[arr[j]].first\_name,peoples[arr[j+1]].first\_name)>0)**

**{**

**swap=arr[j];**

**arr[j]=arr[j+1];**

**arr[j+1]=swap;**

**}**

**for(i=0;i<n;i++)**

**{**

**cout<<peoples[i].first\_name<<" ";**

**cout<<peoples[i].last\_name<<" ";**

**cout<<peoples[i].egn<<endl;**

**}**

**return 0;**

**}**

**Зад.7 Да се дефинира масив с елементи от тип структура. Структурата да съдържа елементите име, факултетен номер, специалност и успех. Да се въведат данните в масива. Да се осъществи достъп до елементите на структурата чрез указател и указателна операция.**

**#include <iostream>**

**using namespace std;**

**struct stud**

**{**

**char name[30];**

**char facNumber[12];**

**char specialnost[20];**

**float uspeh;**

**};**

**int main()**

**{**

**int n,i;**

**stud student[50];**

**stud \*q;**

**q=student;**

**cout<<" Въведете брой на студентите, n <=50 ";**

**cin>>n;**

**for(int i=0;i < n;i++)**

**{**

**cout<<"Въведете ime :";**

**cin>> q->name;**

**cout<<"Въведете facNumber :";**

**cin>> q->facNumber;**

**cout<<"Въведете specialnost :";**

**cin>> q->specialnost;**

**cout<<"Въведете uspeh :";**

**cin>> q->uspeh;**

**q++;**

**}**

**q=student;**

**cout<<"Въведени са следните данни:"<< endl;**

**for( i=0;i<n;i++)**

**{**

**cout<< q->name<< endl;**

**cout<< q->facNumber<< endl;**

**cout<< q->specialnost<< endl;**

**cout<< q->uspeh<< endl<< endl;**

**q++;**

**}**

**return 0;**

**}**

**Резултат:**

**Въведени са следните данни:**

**Ivan**

**Ivanov**

**12345**

**5.43**

**Marin**

**Petrov**

**3456**

**4.6**

**Задача 8: Да се създаде структура telephone, със полета name и number.**

**Да се дефинират полетата на структурата със съответни стойности и да се използва променлива от тип telephone наречена index и да се отпечатат съответното име и телефонен номер.**

**#include <iostream>**

**using namespace std;**

**struct telephone**

**{**

**char \*name;**

**int number;**

**};**

**int main()**

**{**

**struct telephone index;**

**index.name = "Jane Monroe";**

**index.number = 12345;**

**cout << "Name: " << index.name << '\n';**

**cout << "Telephone number: " << index.number;**

**return 0;**

**}**

**Резултат:**

**Name: Jane Monroe**

**Telephone number: 12345**

**Задача 9: Да се състави структура с елементи име и година на създаване на филми. Да се въведат 3 филма и да се отпечатат на екрана.**

**// масив от структури**

**#include <iostream>**

**#include <string>**

**#include <sstream>**

**using namespace std;**

**struct movies\_t {**

**string title;**

**int year;**

**} films [3];**

**void printmovie (movies\_t movie);**

**int main ()**

**{**

**string mystr;**

**int n;**

**for (n=0; n<3; n++)**

**{**

**cout << "Enter title: ";**

**getline (cin,films[n].title);**

**cout << "Enter year: ";**

**getline (cin,mystr);**

**stringstream(mystr) >> films[n].year;**

**}**

**cout << "\nYou have entered these movies:\n";**

**for (n=0; n<3; n++)**

**printmovie (films[n]);**

**return 0;**

**}**

**void printmovie (movies\_t movie)**

**{**

**cout << movie.title;**

**cout << " (" << movie.year << ")\n";**

**}**

**Резултат:**

**Enter title: aaaaa**

**Enter year: 11111**

**Enter title: bbbbb**

**Enter year: 2222**

**Enter title: rrrr**

**Enter year: 3333**

**You have entered these movies:**

**aaaaa (11111)**

**bbbbb (2222)**

**rrrr (3333)**

**Задача 10: Да се използва указател към структура, като се даде възможност на потребителя да въведе от клавиатурата име и година на създаване на филм и програмата да го отпечата на екрана.**

**// указател към структура**

**#include <iostream>**

**#include <string>**

**#include <sstream>**

**using namespace std;**

**struct movies\_t {**

**string title;**

**int year;**

**};**

**int main ()**

**{**

**string mystr;**

**movies\_t amovie;**

**movies\_t \* pmovie;**

**pmovie = &amovie;**

**cout << "Enter title: ";**

**getline (cin, pmovie->title);**

**cout << "Enter year: ";**

**getline (cin, mystr);**

**(stringstream) mystr >> pmovie->year;**

**cout << "\nYou have entered:\n";**

**cout << pmovie->title;**

**cout << " (" << pmovie->year << ")\n";**

**return 0;**

**}**

**Задача 11. Да се напише програма използвайки структура, която по зададен час, минути, секунди да изписва броят на секундите**

**#include<iostream>**

**using namespace std;**

**struct Time{**

**int hours;**

**int minutes;**

**int seconds;**

**};**

**int toSeconds(Time now);**

**int main(){**

**Time t;**

**while (cin>>t.hours>>t.minutes>>t.seconds){**

**cout<<"Total seconds:"<<toSeconds(t)<<endl;**

**}**

**return 0;**

**}**

**int toSeconds(Time now) {**

**return 3600\*now.hours+60\*now.minutes+now.seconds;**

**}**

**Резултат:**

**3 45 67**

**Total seconds:13567**

**Задача 12: Създайте структура Employee със съответните полета nID,nAge,Fwage.**

**#include <iostream>**

**using namespace std;**

**struct Employee**

**{**

**int nID;**

**int nAge;**

**float fWage;**

**};**

**void PrintInformation(Employee sEmployee)**

**{**

**cout << "ID: " << sEmployee.nID << endl;**

**cout << "Age: " << sEmployee.nAge << endl;**

**cout << "Wage: " << sEmployee.fWage << endl << endl;**

**}**

**int main()**

**{**

**Employee sJoe; // create an Employee struct for Joe**

**sJoe.nID = 14;**

**sJoe.nAge = 32;**

**sJoe.fWage = 24.15;**

**Employee sFrank; // create an Employee struct for Frank**

**sFrank.nID = 15;**

**sFrank.nAge = 28;**

**sFrank.fWage = 18.27; // Print Joe's information**

**PrintInformation(sJoe); // Print Frank's information**

**PrintInformation(sFrank);**

**return 0;**

**}**

**Резултат:**

**ID: 14**

**Age: 32**

**Wage: 24.15**

**ID: 15**

**Age: 28**

**Wage: 18.27**

**Задача 13. Да се напише програма, която иска от потребителя да напише любимият си филм и годината на излизане по екраните, като същевременно показва любимият филм на програмиста**

**В C ++ език за програмиране , <sstream> е част от C ++ Standard библиотека . Това е заглавния файл , който предоставя шаблони и типове, които позволяват на синхрона между поточни буфери и низови обекти.**

**#include <iostream>**

**#include <string>**

**#include <sstream>**

**using namespace std;**

**struct movies\_t {**

**string title;**

**int year;**

**} mine, yours;**

**void printmovie (movies\_t movie);**

**int main ()**

**{**

**string mystr;**

**mine.title = "2001 A Space Odyssey";**

**mine.year = 1968;**

**cout << "Enter title: ";**

**getline (cin,yours.title);**

**cout << "Enter year: ";**

**getline (cin,mystr);**

**stringstream(mystr) >> yours.year;**

**cout << "My favorite movie is:\n ";**

**printmovie (mine);**

**cout << "And yours is:\n ";**

**printmovie (yours);**

**return 0;**

**}**

**void printmovie (movies\_t movie)**

**{**

**cout << movie.title;**

**cout << " (" << movie.year << ")\n";**

**}**

**Enter title: Avatar**

**Enter year: 2010**

**My favorite movie is:**

**2001 A Space Odyssey (1968)**

**And yours is:**

**Avatar (2010)’**

**Каква е разликата между структура и класове?**

**Членовете на структурата са public по подразбиране, докато членовете на класа са private.**