

Упражнение 7 – Структури от данни

Понятие за структура

Структурата в програмният език C е потребителски тип данни. Това означава, че програмистът може да създаде свой собствен тип данни. Освен това структурата се води сложен тип данни – това означава, че структурата в себе си може да включва множество променливи от различни типове данни.

Структурите като типове данни се използват в програмирането, за да обединят много типове данни под едно име. Целта е да се даде възможност на програмистите да могат да опишат реални обекти от ежедневието им, като обединят всички техни характеристики в едно. Като пример може да бъде разгледан всеки един предмет без значение какъв е: човек, машина, животно, растение, ежедневна вещ.

Нека се определят характеристиките на обекти нощна лампа: нощната лампа може да се опише с тези характеристики: име на модел, цвят, височина. Също така, същата нощна лампа може да бъде разгледана като част от магазин може да има допълнителни характеристики като цена, производител и т.н.

Проблемът с определянето на характеристиките на класа от обекти, които ще бъдат описани, е възложена на самия програмист. Като пример може да бъде посочено описанието на един цвят – той може да бъде описан по голям брой правилни начини, които да се разливат коренно: цвят може да бъде описан с име – „зелен“, „червен“, „син“, „жълт“. Освен това може да бъде описан и чрез условна номерация: 0 – зелен, 1 – червен, 2 – син, 3 – жълт. Може да бъде представен и според съставните си цветове с определена степен на участие: червен: 255 червено, 0 зелено, 0 синьо; жълт: 255 червено, 255 зелено, 0 син. Изборът изцяло зависи от програмиста.

Освен това една структура може да включва в себе си други структури, по този начин да се изгради по-завършена представа за дадените обекти.

Пример: да се опише като структура автомобил: автомобилът се състои от модел, марка, бр. врати, цвят, тип на купето, двигател. От своя страна двигателят има следните характеристики: брой цилиндри, обем, мощност, брой клапани и т.н. Вижда се, че всеки един предмет може да се опише структурно неограничено за всяка една своя характеристика. От тук се поставя и въпросът със т. нар степен на абстракция – до какво ниво трябва да се стигне с описанието на обектите? Отговорът – зависи от предназначението на тези обекти за целите на програмата.

Структурите са предвестник на класовете в обектно-ориентираните езици. Структурите се състоят само от данни, описващи характеристиките. Всяка характеристика се нарича поле.

Дефиниране на структури в C

```
struct <име_на_структурата>
{
    <тип_на_поле_1> <име_на_поле_1>;
    <тип_на_поле_2> <име_на_поле_2>;
    <тип_на_поле_3> <име_на_поле_3>;
    <тип_на_поле_4> <име_на_поле_4>;
    ...
    <тип_на_поле_N> <име_на_поле_N>;
};
```

По-горе е показано дефинирането на структурита в C, като всеки тип на полета може да е число, указател, символ, указател, масив, символен низ, друга структура и т.н.

```
struct Avtomobil
{
    char marka[20];
    char model[20];
    float obem_na_dvigatel;
    int br_vrati;
    float moshtnost_na_dvigatel;
    float vazdushno_saprotivlenie;
};
```

Употреба на структури

Структурите се използват като нормални променливи с леки особености:

Дефиниране на променливи

Променлива от тип структура се прави по следния начин:

```
struct <име_на_структурата> <име_на_променливата>;
struct Avtomobil kola1;
```

Достъп до полетата

След като е дефинирана променлива от даден тип структура тя трябва да може да се използва, но тъй като е сложен тип, трябва да се използват самите полета: достъп до полетата се осъществява след името на променливата се запише името на полето от структурата. Само така може да се осъществи достъп до полетата както за четене, така и за запис на стойности:

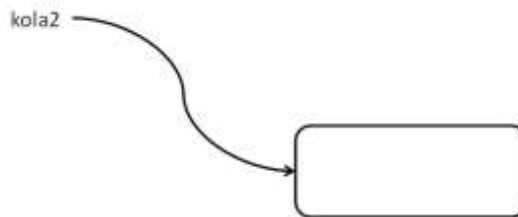
```
struct Avtomobil kola1;
kola1.obem_na_dvigatel = 1.6;
kola1.moshtnost_na_dvigatel = 90;
kola1.br_vrati = 4;
kola1.vazdushno_saprotivlenie = 0.46;
strcpy(kola1.marka, "Toyota");
strcpy(kola1.model, "Corola");
printf("Marka: %s, Model: %s\n", kola1.marka, kola1.model);
printf("Obem: %f, Moshtnost: %f\n", kola1.obem_na_dvigatel, kola1.moshtnost_na_dvigatel);
printf("Broy vrati: %d\n", kola1.br_vrati);
printf("Vazdushno saprotivlenie: %f\n", kola1.vazdushno_saprotivlenie);
```

Масиви от структури

Масивите от структури не различават като дефиниция и употреба от масивите на елементарните типове данни – отново има размер, достъп до елементите на масива чрез индекс. След като бъде достъпен всеки елемент от масива, той се разглежда подобно на горния пример.

Указатели към структури

Както беше споменато на предишното упражнение указателите „сочат“ към полета от паметта. Тук аналогията е същата. Указател към тип структура „сочи“ полето памет, където се намират данните. За това, когато се достъпват полетата се използва „->“ вместо “.”.



```
struct Avtomobil kola1, *kola2;
kola2 = (struct Avtomobil)calloc(1, sizeof(struct Avtomobil));
kola2->obem_na_dvigatel = 1.8;
kola2->moshtnost_na_dvigatel = 110;
kola2->br_vrati = 2;
kola2->vazdushno_saprotivlenie = 0.33;
strcpy(kola2->marka, "Mazda");
strcpy(kola2->model, "3");
printf("Marka: %s, Model: %s\n", kola2->marka, kola2->model);
printf("Obem: %f, Moshtnost: %f\n", kola2->obem_na_dvigatel, kola2->
moshtnost_na_dvigatel);
printf("Broy vrati: %d\n", kola2->br_vrati);
printf("Vazdushno saprotivlenie: %f\n", kola2->vazdushno_saprotivlenie);
```

Потребителски тип данни

В програмния език C има възможност да се дефинират потребителски типове данни, чието име да се използва. За целта се използва запазената дума typedef последвана от стария тип и името на новия тип:

```
typedef unsigned long size_type;

struct Avtomobil
{
    char marka[20];
    char model[20];
    float obem_na_dvigatel;
    int br_vrati;
    float moshtnost_na_dvigatel;
    float vazdushno_saprotivlenie;
};
typedef struct Avtomobil Avtomobil;
```

```
typedef struct {  
    float uskorenie;  
    float vartyasht_moment;  
} SportenAvtomobil;
```

Задачи за изпълнение

1. Да се дефинира структура по избор и да се въведат от клавиатурата полетата на една променлива и да се изведат на екрана.
2. Да се дефинира масив от структурата и да се изведе едно от полетата по избор на всичките елементи в масива в обратен ред.