



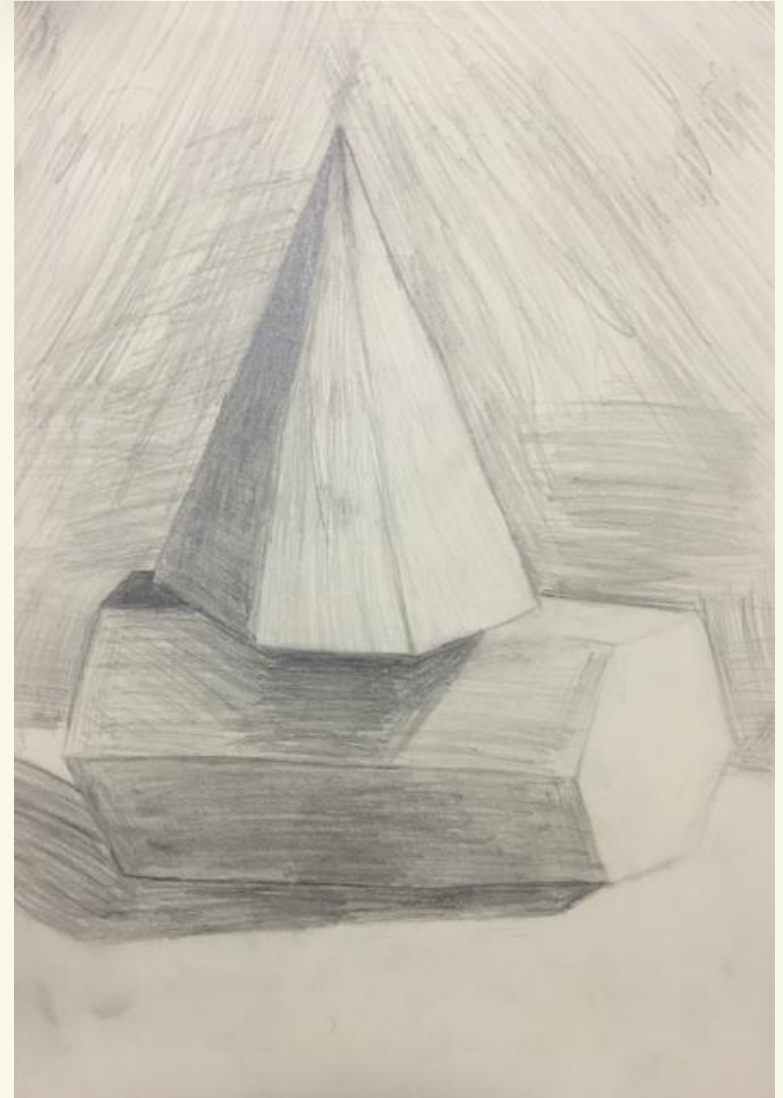
自定义类型-

结构体/共用体/枚举类型

周达天

dtzhou@bjtu.edu.cn

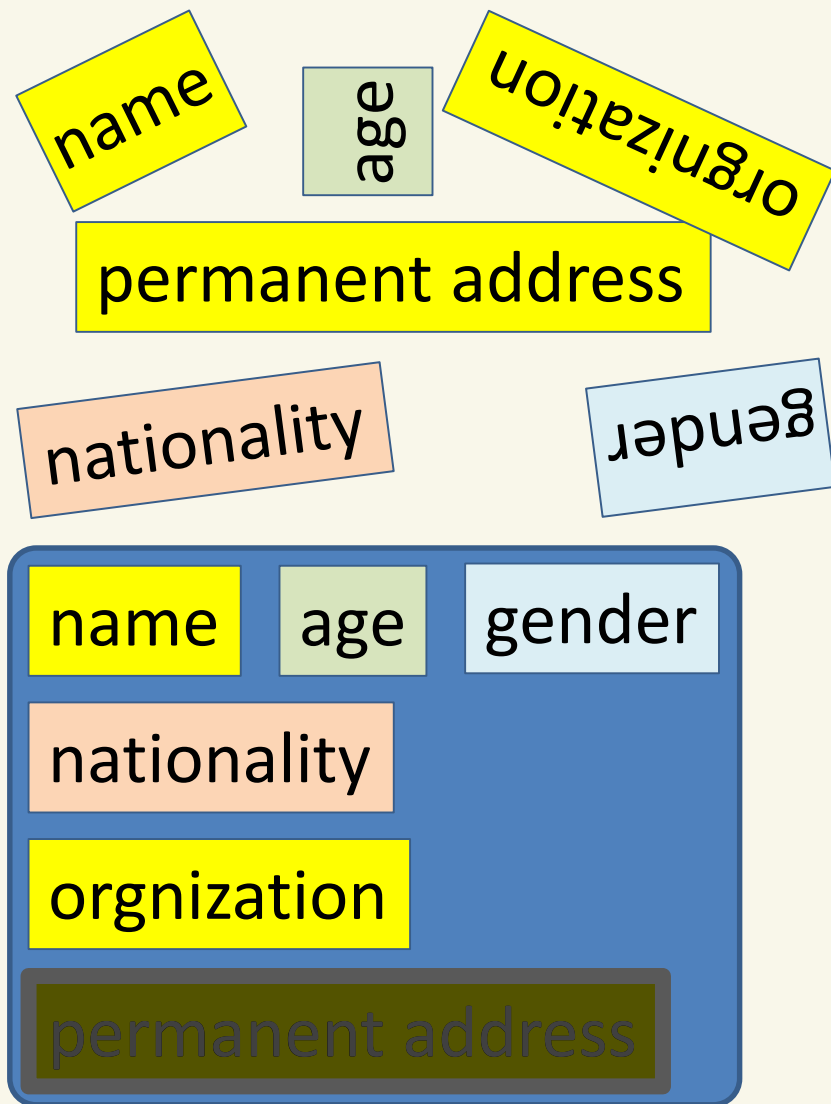
2023.05.10/12



结构体的优点

C语言中**结构体**是**用户自定义**的数据类型。使用结构体能让我们**把不同类型的数据组合在一起**。使用结构体还能让我们构造**复杂**或者说**包含更多含义**的数据类型。某种程度上讲,结构体有点像数组,但数组只能存储相同类型的数据。

Structure is a **user-defined** datatype in C language which allows us to **combine data of different types together**. Structure helps to construct a **complex** data type which is **more meaningful**. It is somewhat similar to an Array, but an array holds data of similar type only. But structure on the other hand, can store data of any type, which is practical more useful.



声明

```
struct [struct_name]
{
    //member
    <type> va1;
    <type> va2;
    <type> va3;
    ...
} [struct_variables];
```

```
struct Student{
    char name[25];
    int age;
    char branch[10];
    char gender;
};
```

```
struct Student S1, S2;
```

分离式

```
struct Student{
    char name[25];
    int age;
    char branch[10];
    char gender;
} S3, S4;
```

一体式

先定义（头文件，文件头），再使用

```
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>


/* Minimal definition of disklabel, so we don't have to include
 * asm/disklabel.h (confuses make)
 */
#ifndef MAXPARTITIONS
#define MAXPARTITIONS 8          /* max. # of partitions */
#endif

#ifndef u8
#define u8 unsigned char
#endif

#ifndef u16
#define u16 unsigned short
#endif

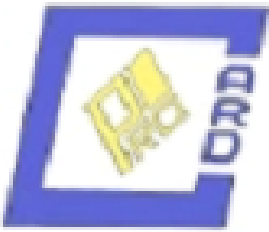
#ifndef u32
#define u32 unsigned int
#endif

struct disklabel {
    u32 d_magic;
    u16 d_type, d_subtype;
    u8 d_typename[16];
    u8 d_packname[16];
    u32 d_sectsize;
    u32 d_nsectors;
    u32 d_ntracks;
    u32 d_ncylinders;
    u32 d_secpercyl;
    u32 d_secprtunit;
    u16 d_sparespertrack;
    u16 d_sparespercyl;
};
```



```
struct disklabel {
    u32 d_magic;
    u16 d_type, d_subtype;
    u8 d_typename[16];
    u8 d_packname[16];
    u32 d_sectsize;
    u32 d_nsectors;
    u32 d_ntracks;
    u32 d_ncylinders;
    u32 d_secpercyl;
    u32 d_secprtunit;
    u16 d_sparespertrack;
    u16 d_sparespercyl;
};
```


先定义（头文件，文件头），再使用



```
#define VPTB    ((unsigned long *) 0x20000000)
#define L1      ((unsigned long *) 0x200802000)

void
pal_init(void)
{
    unsigned long i, rev;
    struct percpu_struct * percpu;
    struct pcb_struct * pcb_pa;

    /* Create the dummy PCB. */
    pcb_va->ksp = 0;
    pcb_va->usp = 0;
    pcb_va->ptbr = L1[1] >> 32;
    pcb_va->asn = 0;
    pcb_va->pcc = 0;
    pcb_va->unique = 0;
    pcb_va->flags = 1;
    pcb_va->res1 = 0;
    pcb_va->res2 = 0;
    pcb_pa = find_pa(VPTB, pcb_va);

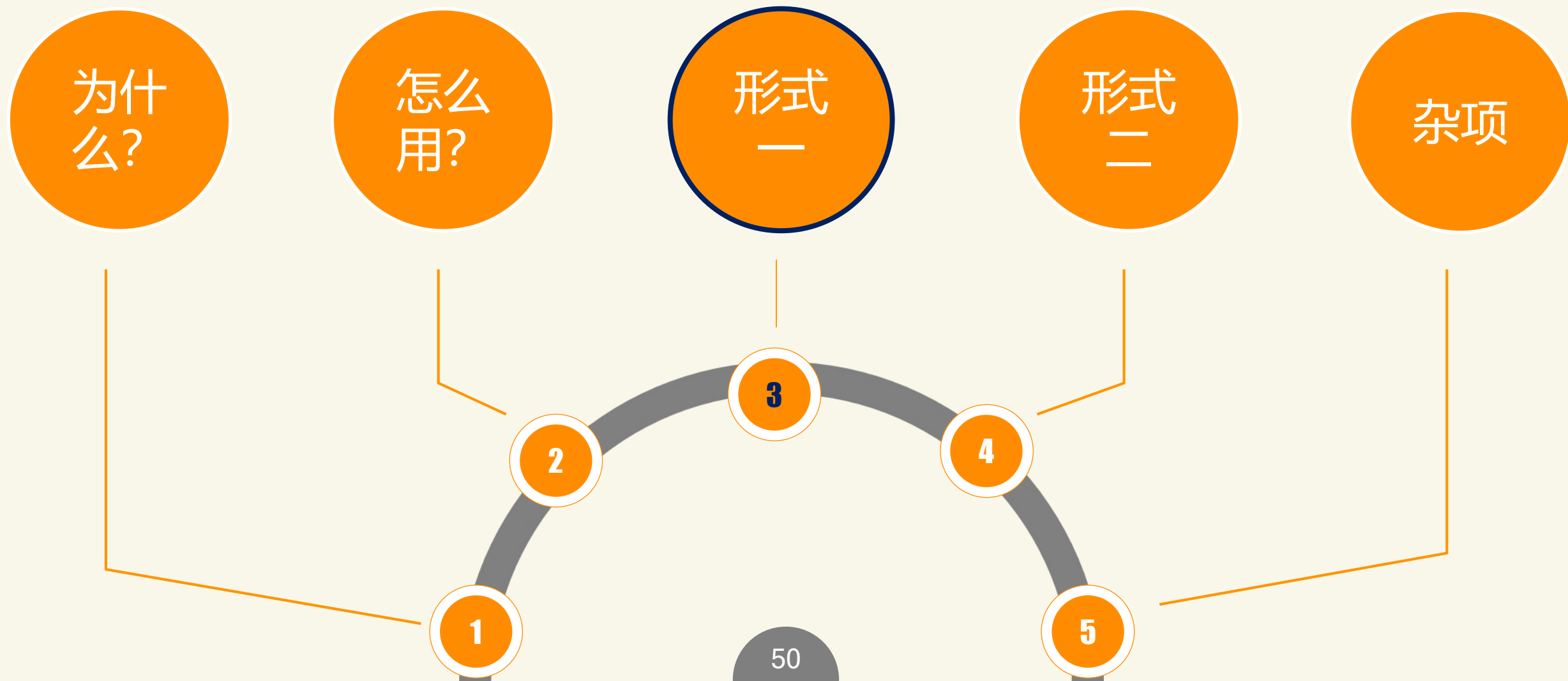
    /*
     * a0 = 2 (OSF)
     * a1 = return address, but we give the asm the vaddr of the PCB
     * a2 = physical addr of PCB
     * a3 = new virtual page table pointer
     * a4 = KSP (but the asm sets it)
     */
    srm_printk("Switching to OSF PAL-code .. ");

    i = switch_to_osf_pal(2, pcb_va, pcb_pa, VPTB);
    if (i) {
        srm_printk("failed, code %ld\n", i);
        __halt();
    }

    percpu = (struct percpu_struct *)
        (INIT_HWRPB->processor_offset + (unsigned long) INIT_HWRPB);
}
```

```
/* Create the dummy PCB. */
pcb_va->ksp = 0;
pcb_va->usp = 0;
pcb_va->ptbr = L1[1] >> 32;
pcb_va->asn = 0;
pcb_va->pcc = 0;
pcb_va->unique = 0;
pcb_va->flags = 1;
pcb_va->res1 = 0;
pcb_va->res2 = 0;
pcb_pa = find_pa(VPTB, pcb_va);
```

通过结构体变量(ming cheng)来访问



访问结构体中的成员（通过-结构体类型变量(名称)：点儿点儿访问式）

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <windows.h>
#include <conio.h>
#include <math.h>
```

```
#define PI 3.14
```

```
struct Star {
    int x, y;
};
```

```
int main() {
    int r, alpha = 0;
```

```
    struct Star s = {0,1};
```

```
    srand((unsigned)time(NULL));
```

```
    do{
        gotoxy(0, 0);
        r=rand()%30;alpha=rand()%360;
```

```
        S.X=r*sin(alpha/180.0*PI)+40;
```

```
        S.y=r*cos(alpha/180.0*PI)+40;
```

```
        printf("The position of star is at (%d,%d)", S.X, S.y);
```

```
        gotoxy(S.X, S.y);
```

```
        printf("*");
```

```
        if (_kbhit())
```

```
            if ('C' == _getch() || 'c' == _getch())    system("Cls");
```

```
            Sleep(50);
```

```
    } while (1);
```

```
    return 0;
```

```
}
```

引出的rand的用法的问题

凯迪丽亚·地里下提
22218252

```
C:\Windows\system32\cmd.e x + v
80 7 99
C:\Users\kdly0\Desktop\高级语言程序设计>test.exe
1683685177:83 1683685177:83 1683685177:83
83 69 99
C:\Users\kdly0\Desktop\高级语言程序设计>test.exe
1683685178:87 1683685178:87 1683685178:87
87 50 63
C:\Users\kdly0\Desktop\高级语言程序设计>test.exe
1683685178:87 1683685178:87 1683685178:87
87 50 63
C:\Users\kdly0\Desktop\高级语言程序设计>test.exe
1683685178:87 1683685178:87 1683685178:87
87 50 63
C:\Users\kdly0\Desktop\高级语言程序设计>test.exe
1683685179:90 1683685179:90 1683685179:90
90 98 59
C:\Users\kdly0\Desktop\高级语言程序设计>test.exe
1683685179:90 1683685179:90 1683685179:90
90 98 59
C:\Users\kdly0\Desktop\高级语言程序设计>test.exe
1683685179:90 1683685179:90 1683685179:90
90 98 59
C:\Users\kdly0\Desktop\高级语言程序设计>test.exe
1683685179:90 1683685179:90 1683685179:90
90 98 59
C:\Users\kdly0\Desktop\高级语言程序设计>text.exe
'text.exe' 不是内部或外部命令，也不是可运行的程序
或批处理文件。
C:\Users\kdly0\Desktop\高级语言程序设计>
```

```
#include"stdio.h"
#include"time.h"
#include
```

```
int main(){
    int a,z,c,sa,sz,sc;
    int d,e,f;
```

```
sa=(unsigned)time(NULL);
srand(sa);
a=rand()%100;
```

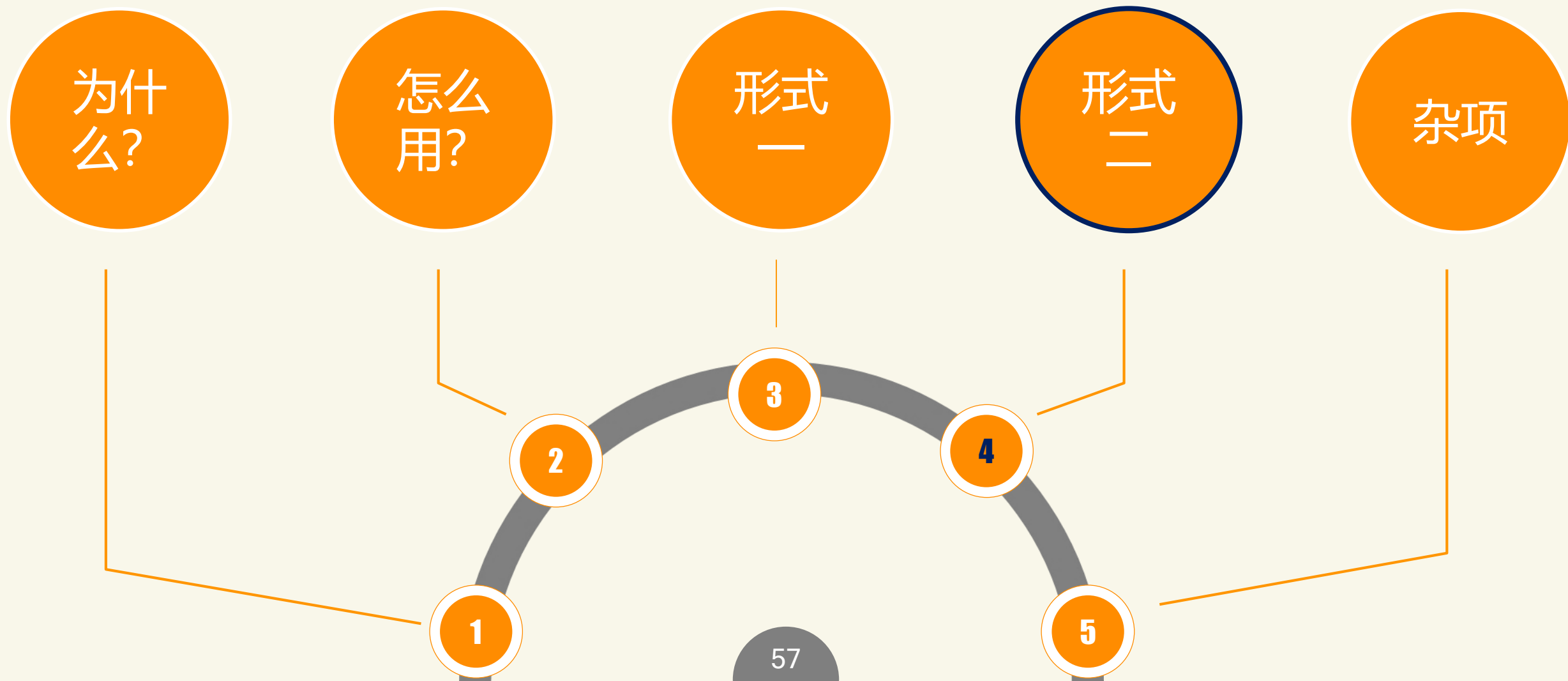
```
sz=(unsigned)time(NULL);
```

```
srand(sz);
z=rand()%100;
```

```
sc=(unsigned)time(NULL);
srand(sc);
c=rand()%100;
printf("%d:%d %d:%d
%d:%d\n",sa,a,sz,z,sc,c);
```

```
srand((unsigned)time(NULL));
d=rand()%100;
e=rand()%100;
f=rand()%100;
printf("%d %d %d",d,e,f);
return 0;
}
```


通过结构体变量的地址（指针）来访问



访问结构体中的成员（通过-结构体类型指针：箭头儿访问式）

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <windows.h>
#include <conio.h>
#include <math.h>
```

```
#define PI 3.14
```

```
struct Star {
    int x, y;
};
```

```
int main() {
    int r, alpha = 0;
```

```
    struct Star s0 = {0,1}, *s=&s0;
```

与形式1进
行对比，找
找两种形式
的差异

```
srand((unsigned)time(NULL));
```

```
do{
    gotoxy(0, 0);
    r=rand()%30;alpha=rand()%300;
    S->X=r*sin(alpha/180.0*PI)+40;
    S->Y=r*cos(alpha/180.0*PI)+40;
    printf("The position of star is at (%d,%d)", S->X, S->Y);
    gotoxy(S->X, S->Y);
    printf("*");
    if (_kbhit())
        if ('C' == _getch() || 'c' == _getch())    system("Cls");
    Sleep(50);
} while (1);

return 0;
}
```

关键词

return	unsigned	default	union	const
int	signed	break	auto	static
double	sizeof	for	register	void
float	if	do	volatile	
char	else	while	<u>typedef</u>	
short	switch	continue	enum	
long	case	<u>struct</u>	extern	

作业（一）设计练习

- 代码书写：访问结构体成员的方式一，访问结构体成员的方式二
- 读课本P.184-190



不是個大人嘛