# 目录

[目录 1](#_Toc520984716)

[第9章 C语言-联合体&枚举 2](#_Toc520984717)

[9.1 C语言联合体概述 2](#_Toc520984718)

[9.2 联合体变量的定义与初始化 2](#_Toc520984719)

[9.2.1 联合体变量定义 2](#_Toc520984720)

[9.3 联合体变量的使用 2](#_Toc520984721)

[9.4 联合体数组的定义与初始化 2](#_Toc520984722)

[9.5 联合体数组的使用 2](#_Toc520984723)

[9.6 枚举概述 2](#_Toc520984724)

[9.7 枚举定义 2](#_Toc520984725)

[9.8 枚举使用 2](#_Toc520984726)

# C语言-联合体&枚举

## C语言联合体概述

C语言中的构造数据类型。

学过的结构体发现，构造结构体数据类型需要一个关键字：struct

联合体也是构造数据类型中的一种，与结构体相似，也需要一个关键字：union

联合体作用：多种数据类型共用同一块内存空间，起到了不浪费空间的目的。

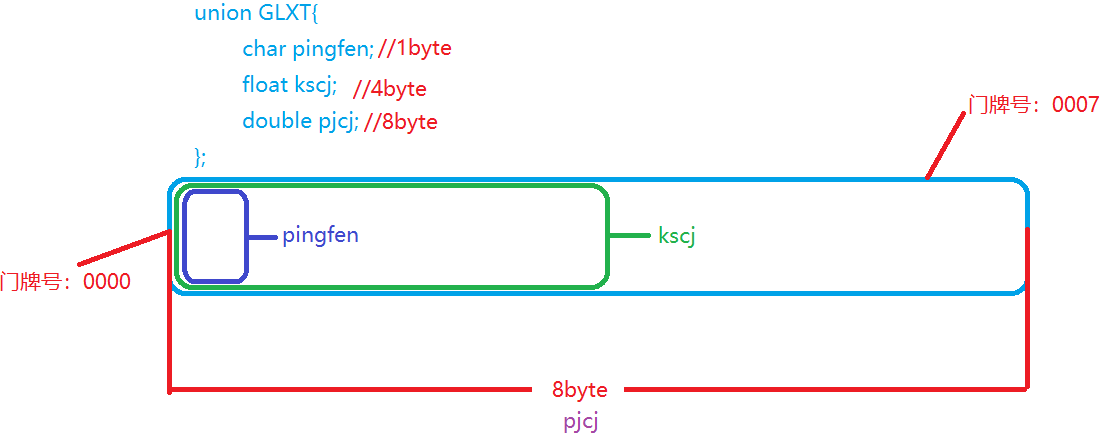
联合体的构造：与结构体的构造一致，换一个关键字即可。

|  |
| --- |
| union GLXT{  char pingfen;  float kscj;  }; |

联合体与结构体区别：

结构体中的每个成员都会占用相应数据类型的空间，并且会进行数据类型的对齐工作。

联合体中的每个成员不会占用相应数据类型的空间，总共占用最大数据类型的空间。



## 联合体变量的定义与初始化

### 联合体变量定义

示例：

|  |
| --- |
| 形式一：  union STU{  char pingfen;  float kscj;  };//构造联合体数据类型  union STU student\_one;// 联合体变量定义  形式二：  union STU{  char pingfen;  float kscj;  }student\_two;//构造联合体数据类型，并且定义一个联合体变量  形式三：  union {  char pingfen;  float kscj;  }student\_three;//构造联合体数据类型，并且定义一个联合体变量  形式一与形式二还可以定义其它变量，因为有联合体构造名称；  形式三中不能再定义其它变量，因为没有联合体构造名称。  特殊形式：  使用数据类型重定义。  typedef union {  char pingfen;  float kscj;  }STUDENT;//构造联合体数据类型  STUDENT student\_four;//定义结构体变量 |

## 联合体变量的使用

示例：

|  |
| --- |
| union GLXT{  char pingfen;  float kscj;  double pjcj;  };  int main(void)  {  union GLXT glxt = {'1'};  unsigned char \* varp = &glxt;    glxt.pingfen = 'a';  printf("%c\r\n", glxt.pingfen);  printf("%#x\r\n",\*varp);    glxt.kscj = 10.2568;  printf("%.6f\r\n", glxt.kscj);  printf("%#x\r\n",\*varp);    glxt.pjcj = 100.2658545665;  printf("%.10f\r\n", glxt.pjcj);  printf("%#x\r\n",\*varp);  printf("%#x\r\n",\*(varp + 7));    glxt.kscj = 1000.2655;  printf("%.6f\r\n", glxt.kscj);  printf("%#x\r\n",\*varp);  printf("%#x\r\n",\*(varp + 7));    return 0;  } |

练习：

|  |
| --- |
| flag：1表示老师；0表示学生  如果为老师，存储评分A~E;如果为学生存储考试成绩：float  存储好了以后，将信息输出  struct GLXT{  char name[20];  int age;  int flag;  union {  char pingfen;  float kscj;  }sx\_ls;  };  #define GLXT\_NUM 4  struct GLXT{  char name[20];  int age;  int flag;  union {  char pingfen;  float kscj;  }sx\_ls;  };  int main(void)  {  struct GLXT glxt[GLXT\_NUM] = {"0"};  int vari = 0;    for(vari = 0; vari < GLXT\_NUM; vari++)  {  printf("请输入姓名、年龄、标志位：");  scanf("%s%d%d", glxt[vari].name, &glxt[vari].age, &glxt[vari].flag);    if(glxt[vari].flag)  {  printf("请输入老师的评分：");  getchar();//录入一次----录入的是空格或回车    glxt[vari].sx\_ls.pingfen = getchar();//录入了评分字母  }  else  {  printf("请输入学生的成绩：");    scanf("%f", &glxt[vari].sx\_ls.kscj);  }  }      for(vari = 0; vari < GLXT\_NUM; vari++)  {  printf("姓名：%s\t年龄：%d\t标志位：%d\r\n", glxt[vari].name, glxt[vari].age, glxt[vari].flag);    if(glxt[vari].flag)  {  printf("老师的评分：%c\r\n",glxt[vari].sx\_ls.pingfen);  }  else  {  printf("学生的成绩：%.6f\r\n",glxt[vari].sx\_ls.kscj);  }  }    return 0;  } |

## 联合体数组的定义与初始化

### 联合体数组的定义

结构体数组一致。

联合体数据类型 数组名[数组长度];

示例：

|  |
| --- |
| union GLXT{  char pingfen;  float kscj;  double pjcj;  };  union GLXT glxt[5];//联合体数组的定义 |

### 联合体数组的初始化

示例：

|  |
| --- |
| union GLXT{  char pingfen;  float kscj;  double pjcj;  };  union GLXT glxt[5] = {{.kscj = 10.234444}, {.pjcj = 34333.988789}, {.pingfen = 'A'}};//联合体数组的定义并且初始化 |

## 联合体数组的使用

示例：

|  |
| --- |
| union GLXT{  char pingfen;  float kscj;  double pjcj;  };  union GLXT glxt[5] = {{.kscj = 10.234444}, {.pjcj = 34333.988789}, {.pingfen = 'A'}};//联合体数组的定义并且初始化  int main(void)  {  int vari = 0;    for(vari = 0; vari < 5; vari++)  {  printf("%c\r\n",glxt[vari]);  }    return 0;  } |

## 枚举概述

数学中的枚举：将某些事情一一列举出来。

C语言中的枚举：表示一件事发生可以产生的结果所有的情况列举出来。

一周：星期日~星期六。

枚举也是一种构造数据类型：同样有个关键字：enum；

形式与结构体、联合体一样。

## 枚举定义

示例：

|  |
| --- |
| enum WEEK{  SUN,//0  MON,//1  TUS = 4454,//4454  WEN,//4455  THU,//4456  FRI,  STA  };  int main(void)  {    enum WEEK week;//枚举类型的变量只能赋枚举值  return 0;  } |

## 枚举使用

示例：

|  |
| --- |
| enum WEEK{  SUN,//0  MON,//1  TUS,//2  WEN,//3  THU,//4  FRI,//5  STA//6  };  int main(void)  {    enum WEEK week;//枚举类型的变量只能赋枚举值    int year = 2018, month = 10, day = 26;  int days = 0;    switch(month)  {  case 12: days += 30;  case 11: days += 31;  case 10: days += 30;  case 9: days += 31;  case 8: days += 31;  case 7: days += 30;  case 6: days += 31;  case 5: days += 30;  case 4: days += 31;  case 3: days += 28;  case 2: days += 31;  case 1: days += 0;  }    days += day;    week = days % 7;    switch(week)  {  case SUN: printf("星期天\r\n");break;  case MON: printf("星期一\r\n");break;  case TUS: printf("星期二\r\n");break;  case WEN: printf("星期三\r\n");break;  case THU: printf("星期四\r\n");break;  case FRI: printf("星期五\r\n");break;  case STA: printf("星期六\r\n");break;  }    return 0;  } |