



UNIVERSIDADE ESTADUAL DE MS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
AEDI

Struct + exemplos

Prof. Nilton

```
typedef struct Ponto {  
    int x;  
    int y;  
} TPonto;
```

```
void adicionaPonto1(TPonto p, int x, int y) {  
    p.x=x;  
    p.y=y;  
}
```

```
...  
TPonto p1={0,0};
```

```
adicionaPonto1(p1,50,20);  
printf("\n%d %d",p1.x,p1.y);
```

```
typedef struct Ponto {  
    int x;  
    int y;  
} TPonto;
```

```
TPonto adicionaPonto2(TPonto p, int x, int y) {  
    p.x=x;  
    p.y=y;  
    return(p);  
}
```

```
TPonto p1={0,0};
```

```
p1=adicionaPonto2(p1,88,44);  
printf("\n%d %d",p1.x,p1.y);
```

```
typedef struct Ponto {  
    int x;  
    int y;  
} TPonto;
```

```
void adicionaPonto3(TPonto *p, int x, int y) {  
    p->x=x;  
    p->y=y;  
}
```

```
...  
TPonto p1={0,0};
```

```
adicionaPonto3(&p1,5,100);  
printf("\n%d %d",p1.x,p1.y);
```

```
typedef struct Ponto {  
    int x;  
    int y;  
} TPonto;
```

```
TPonto *adicionaPonto4(TPonto *p, int x, int y) {  
    p->x=x;  
    p->y=y;  
    return(p);  
}
```

```
TPonto p1={0,0};  
TPonto *p2;
```

```
p2=adicionaPonto4(&p1,75,8);  
printf("\n%d %d    %d %d",p2->x,p2->y,p1.x,p1.y);
```

```
typedef struct Ponto {  
    int x;  
    int y;  
} TPonto;
```

```
TPonto *adicionaPonto5(TPonto *p, int x, int y) {  
    TPonto *paux=p;  
    paux->x=x;  
    paux->y=y;  
    return(paux);  
}
```

```
TPonto p1={0,0};  
TPonto *p2;
```

```
p2=adicionaPonto5(&p1,15,80);  
printf("\n%d %d    %d %d",p2->x,p2->y,p1.x,p1.y);
```

```
typedef struct Ponto {  
    int x;  
    int y;  
} TPonto;
```

```
TPonto adicionaPonto2(TPonto p, int x, int y) {  
    p.x=x;  
    p.y=y;  
    return(p);  
}
```

```
TPonto p1={0,0};  
TPonto p3;
```

```
p3=adicionaPonto2(p1,1,2);  
printf("\n%d %d    %d %d",p3.x,p3.y,p1.x,p1.y);
```

```
typedef struct Ponto {  
    int x;  
    int y;  
} TPonto;
```

```
TPonto adicionaPonto6(TPonto p, int x, int y) {  
    TPonto paux=p;  
    paux.x=x;  
    paux.y=y;  
    return(paux);  
}
```

```
...  
TPonto p1={0,0};  
TPonto p3;
```

```
p3=adicionaPonto6(p1,5,6);  
printf("\n%d %d    %d %d",p3.x,p3.y,p1.x,p1.y);
```



```
typedef struct Ponto {  
    int x;  
    int y;  
} TPonto;
```

```
void adicionaPonto7(TPonto *p, int x, int y) {  
    adicionaPonto8(&p,x,y);  
}
```

```
void adicionaPonto8(TPonto *p, int x, int y) {  
    p->x=x;  
    p->y=y;  
}
```

```
TPonto p1={0,0};
```

```
adicionaPonto7(&p1,10,11);  
printf("\n%d %d",p1.x,p1.y);
```

```
typedef struct Ponto {
    int x;
    int y;
} TPonto;
```

```
void adicionaPonto7(TPonto *p, int x, int y) {
    adicionaPonto8(&p,x,y);
}
```

```
void adicionaPonto8(TPonto *p, int x, int y) {
    p->x=x;
    p->y=y;
}
```

```
TPonto p1={0,0};
```

```
adicionaPonto7(&p1,10,11);
printf("\n%d %d",p1.x,p1.y);
```

| Rotina | Variável | Endereço | Conteúdo |
|---|----------|----------|----------------------|
| main | p1 | 1000 | p1.x = 0 p1.y = 0 |
| adicionaPonto7(&p1,10,11); | | | |
| adicionaponto7 | *p | 1001 | 1000 |
| adicionaponto7 | x | 1002 | 10 |
| adicionaponto7 | y | 1003 | 11 |
| adicionaPonto8(&p,x,y); | | | |
| adicionaponto8 | *p | 1004 | 1001 |
| adicionaponto8 | x | 1005 | 10 |
| adicionaponto8 | y | 1006 | 11 |
| p->x=x; p->y=y; de adicionaPonto8, onde ? | | | |

In function 'adicionaPonto7':

warning: implicit declaration of function 'adicionaPonto8'
warning: conflicting types for 'adicionaPonto8'

```
typedef struct Ponto {  
    int x;  
    int y;  
} TPonto;
```

```
void adicionaPonto7(TPonto *p, int x, int y) {  
    adicionaPonto8(p,x,y);  
}
```

```
void adicionaPonto8(TPonto *p, int x, int y) {  
    p->x=x;  
    p->y=y;  
}
```

```
TPonto p1={0,0};
```

```
adicionaPonto7(&p1,10,11);  
printf("\n%d %d",p1.x,p1.y);
```

```
typedef struct Ponto {
    int x;
    int y;
} TPonto;
```

```
void adicionaPonto7(TPonto *p, int x, int y) {
    adicionaPonto8(p,x,y);
}
```

```
void adicionaPonto8(TPonto *p, int x, int y) {
    p->x=x;
    p->y=y;
}
```

```
TPonto p1={0,0};
```

```
adicionaPonto7(&p1,10,11);
printf("\n%d %d",p1.x,p1.y);
```

| Rotina | Variável | Endereço | Conteúdo |
|---|----------|----------|----------------------|
| main | p1 | 1000 | p1.x = 0 p1.y = 0 |
| adicionaPonto7(&p1,10,11); | | | |
| adicionaponto7 | *p | 1001 | 1000 |
| adicionaponto7 | x | 1002 | 10 |
| adicionaponto7 | y | 1003 | 11 |
| adicionaPonto8(p,x,y); | | | |
| adicionaponto8 | *p | 1004 | 1000 |
| adicionaponto8 | x | 1005 | 10 |
| adicionaponto8 | y | 1006 | 11 |
| p->x=x; p->y=y; de adicionaPonto8, onde ? | | | |

```
typedef struct Ponto {  
    int x;  
    int y;  
} TPonto;
```

```
TPonto adicionaPonto9(int x, int y) {  
    TPonto p;  
    p.x=x;  
    p.y=y;  
    return(p);  
}
```

```
...  
TPonto p4[3];  
int i;  
  
for(i=0;i<3;i++)  
    p4[i]=adicionaPonto9(i*2,i*3);  
for(i=0;i<3;i++)  
    printf("\n%d %d",p4[i].x,p4[i].y);
```

```
typedef struct Ponto {  
    int x;  
    int y;  
} TPonto;
```

```
TPonto *adicionaPonto10(int x, int y) {  
    TPonto p;  
    p.x=x;  
    p.y=y;  
    return(&p);  
}
```

```
...  
TPonto *p4[3];  
int i;  
  
for(i=0;i<3;i++)  
    p4[i]=adicionaPonto10(i*2,i*3);  
for(i=0;i<3;i++)  
    printf("\n%d %d",p4[i]->x,p4[i]->y);
```

```
Process returned -1073741819 (0xC0000005)   execution time : 2.125 s  
Press any key to continue.
```

```
char *leNome() {  
    char nome[50];  
  
    scanf("%s",nome);  
    return(nome);  
}
```

```
...  
printf("\n%s",leNome());
```

```
Prof.Nilton  
  
(null)  
Process returned 0 (0x0)   execution time : 3.968 s  
Press any key to continue.
```

```
In function 'leNome':
```

```
warning: function returns address of local variable
```

```
typedef struct Ponto {  
    int x;  
    int y;  
} TPonto;
```

```
TPonto *adicionaPonto11(int x, int y) {  
    TPonto p[1];  
    p[0].x=x;  
    p[0].y=y;  
    return(p);  
}
```

```
...  
TPonto *p;  
  
p=adicionaPonto11(10,20);  
printf("\n%d %d",p[0].x,p[0].y);
```

```
Process returned -1073741819 (0xC0000005)   execution time : 2.203 s  
Press any key to continue.
```



```
typedef struct Ponto {  
    int x;  
    int y;  
} TPonto;
```

```
void adicionaPonto3(TPonto *p, int x, int y) {  
    p->x=x;  
    p->y=y;  
}
```

```
void printaCoordRetang(TPonto *p) {  
    int i;  
    for(i=0;i<4;i++) {  
        printf("\n%d %d",p[i].x,p[i].y);  
    }  
}
```

...

```
TPonto retangulo[4];
```

```
adicionaPonto3(&retangulo[0],0,0);  
adicionaPonto3(&retangulo[1],10,0);  
adicionaPonto3(&retangulo[2],10,20);  
adicionaPonto3(&retangulo[3],0,20);  
printaCoordRetang(retangulo);
```