

# ФУНКЦИИ



видове параметри

# Видове параметри

- **предавани по стойност** – промяната им във функцията остава само там; при извикването на функцията могат да бъдат променливи, константи, изрази. Декларират се така:  
*тип1 име1, тип2 име2, ....*
- **предавани по адрес** – могат да бъдат само променливи. Промяната на стойността им във функцията води до промяна на стойността на съответните променливи във извикващата я функция. Декларират се така:  
*тип1 &име1, тип2 &име2, ....*

# Кога кои се използват?

- **предавани по стойност** – в повечето случаи
- **предавани по адрес** – когато функцията се налага да върне повече от един резултат. Тогава обикновено функцията връща информация дали е изпълнена задачата, а в параметри се връщат конкретните стойности на решението

## Пример: видове параметри

```
void swap(int a, int b) {  
    int tmp = a;  
    a = b;  
    b = tmp; }  
  
void swap(int &a, int &b) {  
    int tmp = a;  
    a = b;  
    b = tmp; }
```

```
void main() {  
    int i = 10;  
    int j = 20;  
    swap( i, j );  
    cout<<i<<" ";  
    cout<<j<< endl; }
```

# Вярно или невярно?

*Напишете функция за работа с точки, позволяваща въвеждане на точка от клавиатурата*

```
void read(void) {  
    int x, y;  
    cout<<"x:";  
    cin>>x;  
    cout<<"y:";  
    cin>>y;  
}
```

# Невярно!

*Напишете функция за работа с точки, позволяваща въвеждане на точка от клавиатурата и връщаща я към главната програма*

```
void read(void) {  
    int x, y;  
    cout<<"x:";  
    cin>>x;  
    cout<<"y:";  
    cin>>y;  
}
```

# Вярно или невярно?

*Напишете функция за работа с точки, позволяваща въвеждане на точка от клавиатурата*

```
void read(int x, int y) {  
    cout<<"x:";  
    cin>>x;  
    cout<<"y:";  
    cin>>y;  
}
```

# Невярно!

*Напишете функция за работа с точки, позволяваща въвеждане на точка от клавиатурата и връщаща я към главната програма*

```
void read(int x, int y) {  
    cout<<"x:";  
    cin>>x;  
    cout<<"y:";  
    cin>>y;  
}
```



# Вярно!

*Напишете функция за работа с точки, позволяваща въвеждане на точка от клавиатурата (и връщаща я чрез параметрите)*

```
void read(int &x, int &y) {  
    cout<<"x:";  
    cin>>x;  
    cout<<"y:";  
    cin>>y;  
}
```

# Вярно или невярно?

*Напишете функция за работа с точки, позволяваща въвеждане на точка от клавиатурата*

```
point read() {  
    cout<<"x:";  
    cin>>a.x;  
    cout<<"y:";  
    cin>>a.y;  
    return a;  
}
```

# Вярно!

*Напишете функция за  
работа с точки,  
позволяваща  
въвеждане на точка  
от клавиатурата  
(и връщаща я като  
резултат)*

```
point read() {  
    point a;  
    cout<<"x:";  
    cin>>a.x;  
    cout<<"y:";  
    cin>>a.y;  
    return a;  
}
```

# Вярно или невярно?

*Напишете функция,  
върщаща като  
резултат решението  
на уравнението  
 $ax+b=0$*

```
double lineino (double a,double b) {  
    return -b/a;  
}
```

# Невярно!

*Напишете функция,  
върщаща като  
резултат  
решението на  
уравнението  
 $ax+b=0$*

```
double lineino (double a,double b) {  
    return -b/a;  
}
```

*А за  $a=0$ ?*

*А за  $b=0$ ?*

# Вярно или невярно?

*Напишете функция,  
върщаща като  
резултат решението  
на уравнението  
 $ax+b=0$*

```
double lineino (double a,double b) {  
    if (a==0)  
        if(b==0)  
            return 1;  
        else return 2;  
    else return -b/a;  
}
```

# Невярно!

*Напишете  
функция, връщаща  
като резултат  
решението на  
уравнението  
 $ax+b=0$*

*За  $a=0$  и  $b=0$ ?*

*За  $a=1$  и  $b=-1$ ?*

```
double lineino (double a,double b) {  
    if (a==0)  
        if(b==0)  
            return 1;  
        else return 2;  
    else return -b/a;  
}
```

# Вярно!

*Напишете функция,  
върщаща като  
резултат решението  
на уравнението  
 $ax+b=0$*

```
int lineino (double a, double b,  
             double &x) {  
    if (a==0)  
        if(b==0)  
            return 1;  
        else return 2;  
    else {  
        x=-b/a;  
        return 0;  
    }  
}
```



Край

