

Условни оператори

доц. д-р Нора Ангелова

Какво друго ни е нужно в една програма

- Съхранение на информация (памет, типове и променливи)
- Извеждане и въвеждане на информация (входно-изходни оператори)
- Изчисления (аритметични оператори)
- Логически изрази
- ?

Какво друго ни е нужно в една програма

- Съхранение на информация (памет, типове и променливи)
- Извеждане и въвеждане на информация (входн-изходни оператори)
- Изчисления (аритметични оператори)
- Логически изрази
- Проверки?!?

Условен оператор

```
if (<условие>) {  
    <оператор>  
}
```

- if – запазена дума
- <условие> ::= <булев израз>
- <оператор> ::= <произволен оператор>

Пример:

** Когато операторът е само един {} могат да се пропуснат*

```
if (age < 3) {  
    std::cout << "It is a baby" << std::endl;  
}
```

Условен оператор

```
if (<условие>) {  
    <оператор>;  
    {<оператор>;}опц  
}
```

Пример:

* Съвкупността от оператори трябва да бъде оградена в {} скоби.

```
...  
if (age < 3) {  
    std::cout << "It is a baby" << std::endl;  
    std::cout << "The age is " << age << std::endl;  
}  
...  
...
```

Оператор if/else

```
if (<условие>) {  
    <оператор1>;  
    {<оператор1>;} опц  
} else {  
    <оператор2>;  
    {<оператор2>;} опц  
}
```

- if – запазена дума,
- <условие> - булев израз,
- <оператор1> и <оператор2> - произволни оператори,
- ако има само един оператор, скобите могат да се пропуснат.

Пример:

```
if (age < 3) {  
    std::cout << "It is a baby" << std::endl;  
} else {  
    std::cout << "age >= 3" << std::endl;  
}
```

Оператор if/else

- <оператор1> и <оператор2> са произволни оператори за управление на изчислителния процес
- Могат да бъдат условни оператори

Пример:

```
if (age > 4) {
    std::cout << "age > 4" << std::endl;
} else {
    if (age < 4) {
        std::cout << "age < 4" << std::endl;
    } else {
        std::cout << "age == 4" << std::endl;
    }
}
```

Оператор if/else

- <оператор1> и <оператор2> са произволни оператори за управление на изчислителния процес
- Могат да бъдат условни оператори

Пример:

```
if (age > 4) {  
    std::cout << "age > 4" << std::endl;  
} else if (age < 4) {  
    std::cout << "age < 4" << std::endl;  
} else {  
    std::cout << "age == 4" << std::endl;  
}
```


Тернарен оператор

- (`<условие>`) ? `<оператор1>` : `<оператор2>`

Пример:

```
int firstVar = 5;
```

```
int secondVar = 3;
```

```
int max = (firstVar > secondVar) ? firstVar : secondVar;
```

Тернарен оператор

Пример:

```
int firstVar = 5;  
int secondVar = 3;  
int max = (firstVar > secondVar) ? firstVar : secondVar;
```

■ Реализация с if

```
int firstVar = 5;  
int secondVar = 3;  
int max; // Декларацията е извън тялото на if-а
```

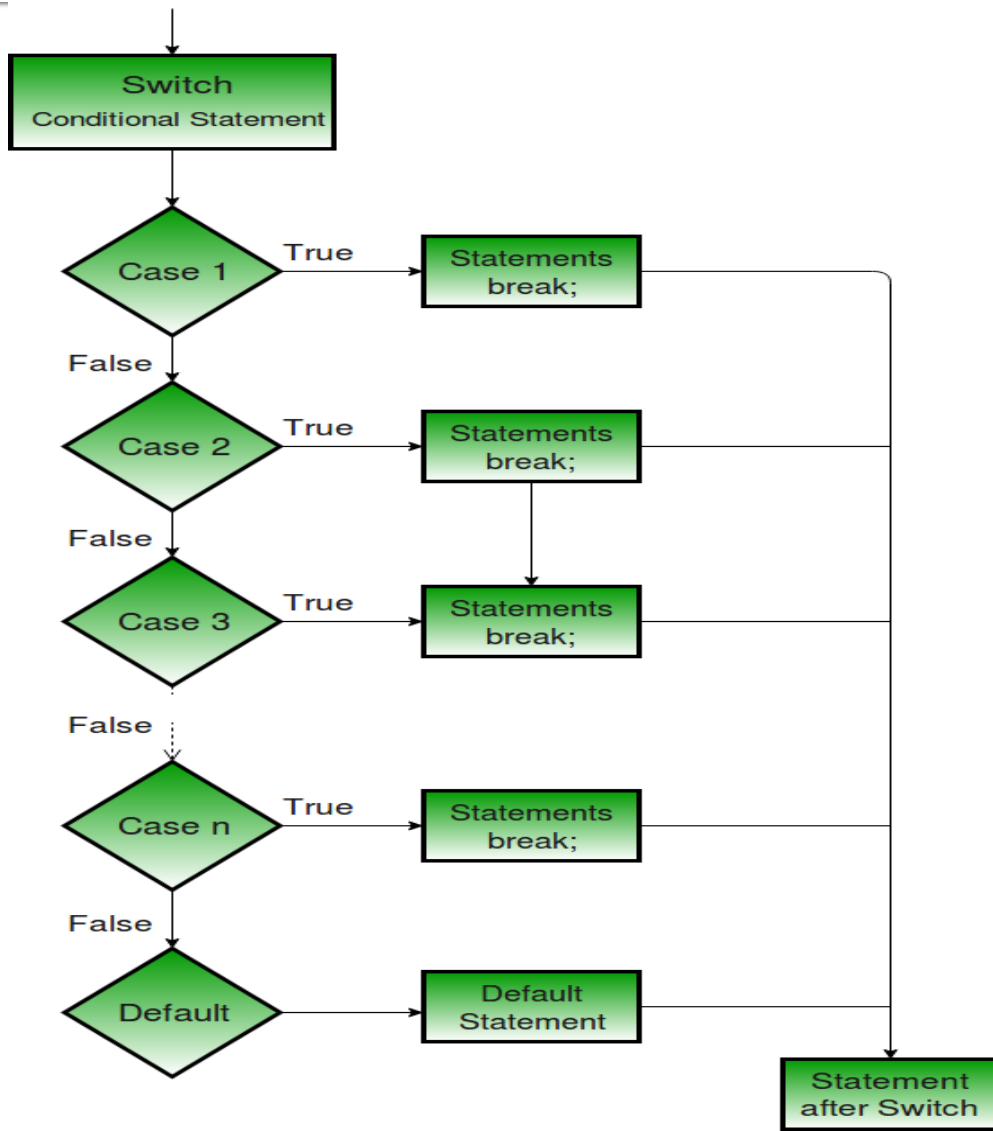
```
if (firstVar > secondVar) {  
    max = firstVar;  
} else {  
    max = secondVar;  
}
```

<http://www.cplusplus.com/articles/1AUq5Di1/>

Оператор switch

- `switch`(<израз>) {
 `case` <израз1> : <редица от оператори 1>
 `case` <израз2> : <редица от оператори 2>
 ...
 [`default`: <редица от оператори n>]_{опц}
}
- `break` – прекратява изпълнението на най-вътрешния, съдържащ го оператор `switch` или оператор за цикъл.

Оператор switch



Пример

```
int value = 1;
int counter = 1;
switch (value) {
case 2: counter += 2;
case 1: counter++;
case 3: counter += 5;
default: counter++;
}
```

```
std::cout << counter;
```

value: 1

counter: 8

Пример

```
int value = 1;
int counter = 1;
switch (value) {
case 2: counter += 2; break;
case 1: counter++; break;
case 3: counter += 5; break;
default: counter++; break;
}
```

```
std::cout << counter;
```

value: 1

counter: 2

Следва продължение...