

C++ 程序设计 I

徐东

xu.dong.sh@outlook.com

信息与计算科学

数学系

上海师范大学

2018 年 8 月 23 日

内容

- 1 选择语句
- 2 关系表达式
- 3 布尔数据类型 (bool)
- 4 逻辑表达式
- 5 其他

计算圆的周长和面积

```
1    #include <iostream>
2    using namespace std;
3
4    int main(){
5        const double PI = 3.14;
6
7        double r = 0;
8
9        cout << "输入圆的半径: ";
10       cin >> r;
11
12       cout << "圆的周长: " << 2 * PI * r << endl;
13       cout << "圆的面积: " << PI * r * r << endl;
14
15       return 0;
16   }
```

计算圆的周长和面积

```
1    #include <iostream>
2    using namespace std;
3
4    int main(){
5        const double PI = 3.14;
6
7        double r = 0;
8
9        cout << "输入圆的半径: ";
10       cin >> r;
11
12       cout << "圆的周长: " << 2 * PI * r << endl;
13       cout << "圆的面积: " << PI * r * r << endl;
14
15       return 0;
16   }
```

- 只有在半径大于零的情况下，计算圆周长和面积才有意义。

计算圆的周长和面积

.....

```
cout << "输入圆的半径 : ";
```

```
cin >> r;
```

```
cout << "圆的周长 : " << 2 * PI * r << endl;
```

```
cout << "圆的面积 : " << PI * r * r << endl;
```

计算圆的周长和面积

.....

```
cout << "输入圆的半径 : ";
```

```
cin >> r;
```

//只有在半径(r)大于零的情况下,才去执行下述两条语句!

```
cout << "圆的周长 : " << 2 * PI * r << endl;
```

```
cout << "圆的面积 : " << PI * r * r << endl;
```

计算圆的周长和面积

.....

```
cout << "输入圆的半径 : ";
```

```
cin >> r;
```

//只有在半径(r)大于零的情况下,才去执行下述两条语句!

```
cout << "圆的周长 : " << 2 * PI * r << endl;
```

```
cout << "圆的面积 : " << PI * r * r << endl;
```

- 有条件地执行某些语句

计算圆的周长和面积

.....

```
cout << "输入圆的半径 : ";
```

```
cin >> r;
```

//只有在半径(r)大于零的情况下,才去执行下述两条语句!

```
cout << "圆的周长 : " << 2 * PI * r << endl;
```

```
cout << "圆的面积 : " << PI * r * r << endl;
```

- 有条件地执行某些语句
- *if* 语句

选择语句

- **if** 语句

```
if(条件){
```

在条件成立的情况下，需要执行的（若干）语句；

```
}
```

选择语句

- if 语句

```
if(条件){
```

在条件成立的情况下，需要执行的（若干）语句；

```
}
```

- 单向选择语句

选择语句

- if 语句

```
if(条件){
```

在条件成立的情况下，需要执行的（若干）语句；

```
}
```

- 单向选择语句

- 条件

选择语句

- if 语句

```
if(条件){
```

在条件成立的情况下，需要执行的（若干）语句；

```
}
```

- 单向选择语句

- 条件

- 条件表达式

- 逻辑表达式

选择语句

- if 语句

```
if(条件){
```

在条件成立的情况下，需要执行的（若干）语句；

```
}
```

- 单向选择语句

- 条件

- 条件表达式 简单条件

- 逻辑表达式

选择语句

- if 语句

```
if(条件){
```

在条件成立的情况下，需要执行的（若干）语句；

```
}
```

- 单向选择语句

- 条件

- 条件表达式 简单条件
- 逻辑表达式 复杂条件

简单的条件判断

- 只有在半径(r)大于零的情况下，计算输出圆的周长和面积。

简单的条件判断

- 只有在半径(r)大于零的情况下，计算输出圆的周长和面积。
- 条件

半径大于 0

简单的条件判断

- 只有在半径(r)大于零的情况下, 计算输出圆的周长和面积。
- 条件

半径大于 0

- C++ 的形式

$$r > 0$$

简单的条件判断

- 只有在半径(r)大于零的情况下, 计算输出圆的周长和面积。
- 条件

半径大于 0

- C++ 的形式

$r > 0$

- 条件表达式

简单的条件判断

- 只有在半径(r)大于零的情况下, 计算输出圆的周长和面积。
- 条件

半径大于 0

- C++ 的形式

$r > 0$

- 条件表达式
- $>$ 条件运算符

条件表达式

符号	意义	例子
<	小于(<)	$a < b$
<=	小于等于(\leq)	$a \leq b$
>	大于(>)	$a > b$
>=	大于等于(\geq)	$a \geq b$
==	相等(=)	$a == b$
!=	不相等(\neq)	$a != b$

关系表达式

表达式	结果
$3 < 6$	true
$6 \leq 6$	true
$6 > 7$	false
$\text{days} \geq 7$?
$3 == 3$	true
$3 != 6$	true

- 其中, *days* 是一个变量。

关系表达式

- 关系表达式代表简单的条件判断
- 关系表达式的结果
 - true
 - false
- 关系运算符是二元运算符
 - ==
 - >=
 - <=

计算圆的周长和面积

```
.....  
cout << "输入圆的半径 : ";  
cin >> r;  
if(r > 0){  
    cout << "圆的周长 : " << 2 * PI * r << endl;  
    cout << "圆的面积 : " << PI * r * r << endl;  
}
```

计算圆的周长和面积

.....

```
cout << "输入圆的半径 : ";
```

```
cin >> r;
```

```
if(r > 0){
```

```
    cout << "圆的周长 : " << 2 * PI * r << endl;
```

```
    cout << "圆的面积 : " << PI * r * r << endl;
```

```
}
```


计算圆的周长和面积

```
.....  
cout << "输入圆的半径 : ";  
cin >> r;  
if(r > 0){  
    cout << "圆的周长 : " << 2 * PI * r << endl;  
    cout << "圆的面积 : " << PI * r * r << endl;  
}
```

- 关键字 *if* 后的小括号 ()

计算圆的周长和面积

```
.....  
cout << "输入圆的半径 : ";  
cin >> r;  
if(r > 0){  
    cout << "圆的周长 : " << 2 * PI * r << endl;  
    cout << "圆的面积 : " << PI * r * r << endl;  
}
```

- 关键字 *if* 后的小括号 ()
- 用于隔离外部代码的大括号 { }

布尔型

- 布尔型代表“真、假”、“成立、不成立”等概念。
- bool
 - true
 - false
- 可以使用 `int` 代替 `bool`
 - `C++` 向下兼容 `C` 语言

布尔型

- *bool* 和 *int* 的互换

$bool \Rightarrow int$	$int \Rightarrow bool$
$true \rightarrow 1$	非 0 $\rightarrow true$
$false \rightarrow 0$	0 $\rightarrow false$

计算圆的周长和面积

- 只计算半径在 $(0, 10]$ 之间的圆的周长和面积

计算圆的周长和面积

- 只计算半径在 $(0, 10]$ 之间的圆的周长和面积

计算圆的周长和面积

- 只计算半径在 $(0, 10]$ 之间的圆的周长和面积
- 复杂条件的实现

计算圆的周长和面积

- 只计算半径在 $(0, 10]$ 之间的圆的周长和面积
- 复杂条件的实现
 - 嵌套的 *if* 语句

计算圆的周长和面积

- 只计算半径在 $(0, 10]$ 之间的圆的周长和面积
- 复杂条件的实现
 - 嵌套的 *if* 语句
 - 逻辑表达式

半径 $\in (0, 10]$: 嵌套的 *if* 语句

```
.....
```

```
cin >> r;
```

```
cout << "圆的周长 : " << 2 * PI * r << endl;
```

```
cout << "圆的面积 : " << PI * r * r << endl;
```

半径 $\in (0, 10]$: 嵌套的 *if* 语句

```
.....
```

```
cin >> r;
```

```
if(r > 0){
```

```
    cout << "圆的周长 : " << 2 * PI * r << endl;
```

```
    cout << "圆的面积 : " << PI * r * r << endl;
```

```
}
```

半径 $\in (0, 10]$: 嵌套的 *if* 语句

```
.....  
cin >> r;  
if(r > 0){  
    if(r <= 10){  
        cout << "圆的周长 : " << 2 * PI * r << endl;  
        cout << "圆的面积 : " << PI * r * r << endl;  
    }  
}
```

半径 $\in (0, 10]$: 嵌套的 *if* 语句

```
.....  
cin >> r;  
if(r > 0){  
    if(r <= 10){  
        cout << "圆的周长 : " << 2 * PI * r << endl;  
        cout << "圆的面积 : " << PI * r * r << endl;  
    }  
}
```

- 先判断外层条件；再决定是否继续判定内层条件。

逻辑表达式

- 复杂条件

$$r \in (0, 10]$$

- 逻辑表达式

逻辑表达式

- 复杂条件

$$r \in (0, 10]$$

- 逻辑表达式

$$r > 0$$

逻辑表达式

- 复杂条件

$$r \in (0, 10]$$

- 逻辑表达式

$$r > 0 \quad r \leq 10$$

逻辑表达式

- 复杂条件

$$r \in (0, 10]$$

- 逻辑表达式

$$r > 0 \ \&\& \ r \leq 10$$

逻辑表达式

- 复杂条件

$$r \in (0, 10]$$

- 逻辑表达式

$$r > 0 \ \&\& \ r \leq 10$$

- `&&` 逻辑运算符 (逻辑与)

半径 $\in (0, 10]$: 逻辑表达式

.....

```
cin >> r;
```

```
if(r > 0 && r < 10){
```

```
    cout << "圆的周长 : " << 2 * PI * r << endl;
```

```
    cout << "圆的面积 : " << PI * r * r << endl;
```

```
}
```

逻辑表达式

- 逻辑运算符

- 逻辑与 (并且) &&
- 逻辑或 (或者) ||
- 逻辑非 (取反) !

- 逻辑表达式

- 由若干 (关系) 表达式通过逻辑运算符连接而成的表达式
(形成复杂的条件判断)
- 逻辑表达式的结果是 bool 值

逻辑表达式的计算 (真假值表)

表达式	值	说明
表达式1 && 表达式2	true	表达式1与表达式2都为true
表达式1 表达式2	true	表达式1与表达式2不全为false
!表达式1	true	表达式1为false

逻辑表达式的计算 (真假值表)

表达式	值	说明
表达式1 && 表达式2	true	表达式1与表达式2都为true
表达式1 表达式2	true	表达式1与表达式2不全为false
!表达式1	true	表达式1为false

● 短路运算问题

逻辑运算符

- 优先级 (从高到低排列)

① !

② &&

③ ||

逻辑运算符

- 优先级 (从高到低排列)

① !

② &&

③ ||

- 结合性

- 右结合 !

- 左结合 && ||

逻辑运算符

- 优先级 (从高到低排列)

- ① !

- ② &&

- ③ ||

- 结合性

- 右结合 !

- 左结合 && ||

- 通过添加小括号()明确逻辑运算的关系

判断闰年

● 判断闰年的条件

年份能被 4 整除但不能被 100 整除

或者

年份能被 100 整除又能被 400 整除

注意

- 区分 `=` 和 `==`
- 编译器以条件表达式的结果是否为 `0` 来判断条件是否成立
- 短路运算
 - 确定整个逻辑表达式的最终值之后，立即结束整个表达式的执行。
 - 逻辑与 `&&`
 - 逻辑或 `||`

注意

- 判断两个浮点数相等的错误方式

```
if(0.3 == 0.1 + 0.1 + 0.1){  
    cout << "equal" ;  
}
```

注意

- 判断两个浮点数相等的错误方式

```
if(0.3 == 0.1 + 0.1 + 0.1){  
    cout << "equal" ;  
}
```

- 浮点数不是实数（不精确）

注意

- 判断两个浮点数相等的错误方式

```
if(0.3 == 0.1 + 0.1 + 0.1){  
    cout << "equal" ;  
}
```

- 浮点数不是实数（不精确）
- 正确方式：判断条件 $|a - b| < \epsilon$ 是否成立

格式化输出布尔值

```
1  cout << "default_bool_values:"  
2      << true << "\t" << false << endl; //默认  
3  
4  cout << "alpha_bool_values:" << std::boolalpha  
5      << true << "\t" << false << endl; //格式化  
6  
7  cout << "no_reset_bool:"  
8      << true << "\t" << false << endl; //没有重置  
9  
10 cout << std::noboolalpha << "reset_bool:"  
11     << true << "\t" << false << std::endl; //重置
```

Q & A