

# 02 - 变量与输入输出

## C++ 程序设计基础

SOJ 信息学竞赛教练组

2024 年 5 月 16 日

# 目录

1 复习回顾

2 变量

3 数据类型

4 常量

5 类型转换

6 总结

# 程序的基本结构

```
1 #include <iostream>    // 头文件
2
3 using namespace std;    // 命名空间
4
5 // 程序的主体
6 int main() {            // 程序的入口：主函数
7     // 程序的内容
8
9     return 0;           // 程序的出口
10 }
```

# 输出语句

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     cout << "Hello, world!" << endl;
7
8     return 0;
9 }
```

# 输出语句

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     cout << "Hello, world!" << endl;
7     cout << 1 << " + " << 2 << " = " << 3 << endl;
8
9     return 0;
10 }
```

# 格式控制

```
1 #include <iostream>
2 #include <iomanip>
3
4 using namespace std;
5
6 int main() {
7     cout << fixed << setprecision(2) << 3.14159 << endl;
8
9     return 0;
10 }
```

# 格式控制

```
1 #include <iostream>
2 #include <iomanip>
3
4 using namespace std;
5
6 int main() {
7     cout << fixed << setprecision(2) << 3.14159 << endl;
8
9     return 0;
10 }
```

- 输出结果：3.14

# 目录

1 复习回顾

**2 变量**

3 数据类型

4 常量

5 类型转换

6 总结



- 变量的概念
  - 变量就是在程序运行过程中，其值可以被改变的量

- 变量的概念
  - 变量就是在程序运行过程中，其值可以被改变的量
- 变量的作用
  - 程序运行过程中**存储数据**的“盒子”

- 变量的概念
  - 变量就是在程序运行过程中，其值可以被改变的量
- 变量的作用
  - 程序运行过程中**存储数据**的“盒子”
- 变量的声明
  - **类型 变量名**
  - `int a;`

- 变量的初始化
  - 变量在声明的同时设置一个初始的数值
  - **类型 变量名 = 值**
  - `int a = 2;`

- 变量的初始化
  - 变量在声明的同时设置一个初始的数值
  - **类型 变量名 = 值**
  - `int a = 2;`
- 变量的赋值
  - 在变量声明之后，可以通过 **=** 修改变量存储的数据
  - `a = 3;` // 变量 `a` 中存储的数据会从 2 变为 3

# 变量的命名规则

- 变量名只能由字母、数字或下划线构成，但不能以数字开头
  - `dis2` 或 `_dis` 是合法变量名
  - `2dis` 是非法变量名

# 变量的命名规则

- 变量名只能由字母、数字或下划线构成，但不能以数字开头
  - `dis2` 或 `_dis` 是合法变量名
  - `2dis` 是非法变量名
- 变量名区分大小写
  - `Dis` 与 `dis` 视作不同变量名

# 变量的命名规则

- 变量名只能由字母、数字或下划线构成，但不能以数字开头
  - `dis2` 或 `_dis` 是合法变量名
  - `2dis` 是非法变量名
- 变量名区分大小写
  - `Dis` 与 `dis` 视作不同变量名
- 变量名不能与 C++ 的关键字重名
  - `std` 和 `return` 是非法变量名



# 变量的输入与输出

- 当我们希望变量的值不是预先写入程序，而是通过用户输入决定时，应该怎么办？

# 变量的输入与输出

- 当我们希望变量的值不是预先写入程序，而是通过用户输入决定时，应该怎么办？
- 使用 `cin` 语句输入变量，在输入流符号 `>>` 后写上变量名
  - `int a;`  
`cin >> a;`

# 变量的输入与输出

- 当我们希望变量的值不是预先写入程序，而是通过用户输入决定时，应该怎么办？
- 使用 `cin` 语句输入变量，在输入流符号 `>>` 后写上变量名
  - `int a;`  
`cin >> a;`
- 使用 `cout` 语句输出变量，在输出流符号 `<<` 后写上变量名
  - `cout << a << endl;`

# 多个变量的使用

- 可以同时声明多个变量
  - `int a, b;`

# 多个变量的使用

- 可以同时声明多个变量
  - `int a, b;`
- 同时输入多个变量
  - `cin >> a >> b;`

# 多个变量的使用

- 可以同时声明多个变量
  - `int a, b;`
- 同时输入多个变量
  - `cin >> a >> b;`
- 同时输出多个变量
  - `cout << a << " " << b << endl;`

## 例 2.1：输出长方形的长和宽 I

### 编程题

- 编写程序，由用户输入两个整数  $a, b$ ，代表长方形的长和宽，再输出长和宽。
- 输出分两行，第一行输出长，第二行输出宽。
- 样例输入  
5 3
- 样例输出  
length of rectangle: 5  
width of rectangle: 3

## 例 2.1：输出长方形的长和宽 I

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int a, b;
7     cin >> a >> b;
8     cout << "length of rectangle: " << a << endl;
9     cout << "width of rectangle: " << b << endl;
10
11     return 0;
12 }
```



# 目录

1 复习回顾

2 变量

**3 数据类型**

4 常量

5 类型转换

6 总结

# 变量的数据类型

## 整数类型

- `int`:  $-2^{31} \sim 2^{31} - 1$  (近似值  $2.1 \times 10^9$ )
- `long long`:  $-2^{63} \sim 2^{63} - 1$  (近似值  $9.2 \times 10^{18}$ )
- `int a = 123;`  
`cout << a << endl; // 输出 123`

# 变量的数据类型

## 整数类型

- `int`:  $-2^{31} \sim 2^{31} - 1$  (近似值  $2.1 \times 10^9$ )
- `long long`:  $-2^{63} \sim 2^{63} - 1$  (近似值  $9.2 \times 10^{18}$ )
- `int a = 123;`  
`cout << a << endl;` // 输出 123
- `int b = 123456789123;`  
`cout << b << endl;` // 输出错误, 整数溢出

# 变量的数据类型

## 整数类型

- `int`:  $-2^{31} \sim 2^{31} - 1$  (近似值  $2.1 \times 10^9$ )
- `long long`:  $-2^{63} \sim 2^{63} - 1$  (近似值  $9.2 \times 10^{18}$ )
- `int a = 123;`  
`cout << a << endl;` // 输出 123
- ~~`int b = 123456789123;`~~  
~~`cout << b << endl;`~~ // 输出错误, 整数溢出
- `long long b = 123456789123;`  
`cout << b << endl;` // 输出 123456789123

# 变量的数据类型

## 浮点数类型（实数/小数）

- double: 双精度浮点数
- ```
double d = 2.33;  
cout << d << endl; // 输出 2.33
```

# 变量的数据类型

## 字符类型

- char: 键盘上的字母、数字及符号
- ```
char c = '!';  
cout << c << endl; // 输出 !
```

## 布尔类型

- `bool`: `true` / `false` (真 / 假)
- `bool t = true;`  
`cout << t << endl;` // 输出 1
- `bool f = false;`  
`cout << f << endl;` // 输出 0

## 例 2.2：输出长方形的长和宽 II

### 编程题

- 编写程序，由用户输入两个实数  $a, b$ ，代表长方形的长和宽，再输出长和宽。
- 输出分两行，第一行输出长，第二行输出宽。结果保留 2 位小数。
- 样例输入  
5.727 3.472
- 样例输出  
length of rectangle: 5.73  
width of rectangle: 3.47



## 例 2.2：输出长方形的长和宽 II

```
1 #include <iostream>
2 #include <iomanip>
3
4 using namespace std;
5
6 int main() {
7     double a, b;
8     cin >> a >> b;
9     cout << "length of rectangle: ";
10    cout << fixed << setprecision(2) << a << endl;
11    cout << "width of rectangle: ";
12    cout << fixed << setprecision(2) << b << endl;
13
14    return 0;
15 }
```

# 目录

1 复习回顾

2 变量

3 数据类型

**4 常量**

5 类型转换

6 总结

- 常量的概念
  - 常量就是在程序运行过程中，其值不能被改变的量

- 常量的概念
  - 常量就是在程序运行过程中，其值不能被改变的量
- 常量的作用
  - 存储在程序运行过程中不能被修改的数据，更易于维护
  - 例如：圆周率  $PI$

- 常量的概念
  - 常量就是在程序运行过程中，其值不能被改变的量
- 常量的作用
  - 存储在程序运行过程中不能被修改的数据，更易于维护
  - 例如：圆周率  $PI$
- 常量的声明
  - 声明常量与声明变量类似
  - 但需要在数据类型前加 `const` 关键字
  - `const double PI = 3.14159;`

- 常量必须在声明的同时初始化

# 注意事项

- 常量必须在声明的同时初始化
- 不能被重新赋值
  - `const double PI;`  
`PI = 3.14159;`

# 注意事项

- 常量必须在声明的同时初始化
- 不能被重新赋值
  - ~~const double PI;~~  
~~PI = 3.14159;~~



# 注意事项

- 常量必须在声明的同时初始化
- 不能被重新赋值
  - ~~const double PI;~~  
~~PI = 3.14159;~~
- 也不能输入
  - const double PI;  
cin >> PI;

# 注意事项

- 常量必须在声明的同时初始化
- 不能被重新赋值
  - ~~const double PI;~~  
~~PI = 3.14159;~~
- 也不能输入
  - ~~const double PI;~~  
~~cin >> PI;~~

- 字面量的概念
  - C++ 源代码中的固定数值，是预先写入程序中的数据
  - `int a = 2;`

- 字面量的概念
  - C++ 源代码中的固定数值，是预先写入程序中的数据
  - `int a = 2;`
- 字面量的数据类型
  - `cout << 3 << endl;` // 3 默认是整数类型
  - `cout << 3.0 << endl;` // 3.0 默认是浮点数类型

# 目录

1 复习回顾

2 变量

3 数据类型

4 常量

**5 类型转换**

6 总结

# 类型转换

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     long long a = 123456789123;
7     int b = a;
8     cout << a << endl;
9     cout << b << endl;
10
11     return 0;
12 }
```

# 类型转换

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     long long a = 123456789123;
7     int b = a;
8     cout << a << endl;
9     cout << b << endl;
10
11     return 0;
12 }
```

- 以上程序能编译通过吗?
- 运行结果是两行 123456789123 吗?

- 任何基本数据类型之间都可以相互转换，常见的有：
  - `int <=> long long`
  - `int <=> double`
  - `int <=> bool`



- int 和 long long 的数据范围不同（容量不同）

# 类型转换

- int 和 long long 的数据范围不同（容量不同）
- int => long long
  - 数据正常转换

# 类型转换

- int 和 long long 的数据范围不同（容量不同）
- int  $\Rightarrow$  long long
  - 数据正常转换
- long long  $\Rightarrow$  int
  - 超过 int 类型上限 ( $2.1 \times 10^9$ ) 时会**溢出**，导致数据不准确
  - 其它情况下，数据正常转换

- double 和 int 的区别在于是否存在小数部分

# 类型转换

- double 和 int 的区别在于是否存在小数部分
- double  $\Rightarrow$  int
  - 删除小数部分
  - 2.9  $\Rightarrow$  2

# 类型转换

- double 和 int 的区别在于是否存在小数部分
- double  $\Rightarrow$  int
  - 删除小数部分
  - 2.9  $\Rightarrow$  2
- int  $\Rightarrow$  double
  - 添加小数部分
  - 5  $\Rightarrow$  5.0

- bool 类型表示 true / false, 实际存储的是数值 1 / 0

# 类型转换

- bool 类型表示 true / false, 实际存储的是数值 1 / 0
- bool => int
  - false => 0
  - true => 1



# 类型转换

- bool 类型表示 true / false, 实际存储的是数值 1 / 0
- bool  $\Rightarrow$  int
  - false  $\Rightarrow$  0
  - true  $\Rightarrow$  1
- int  $\Rightarrow$  bool
  - 0  $\Rightarrow$  false
  - 非 0  $\Rightarrow$  true

# 类型转换

- 直接赋值时，会产生隐式类型转换
  - `int a = 2.9;`  
`cout << a << endl;` // 输出 2

# 类型转换

- 直接赋值时，会产生隐式类型转换
  - `int a = 2.9;`  
`cout << a << endl; // 输出 2`
- 也可以通过 **(类型) 变量名** 的形式进行强制类型转换
  - `double b = 2.9;`  
`cout << b << endl; // 输出 2.9`  
`cout << (int)b << endl; // 输出 2`

# 目录

1 复习回顾

2 变量

3 数据类型

4 常量

5 类型转换

**6 总结**

- 变量与常量
  - 声明、初始化
  - 赋值
  - 输入输出
- 数据类型与类型转换

# Thank you!