一、数及其运算

【大纲内容】

【1】自然数、整数、有理数、实数及其算术运算(加、减、乘、除)

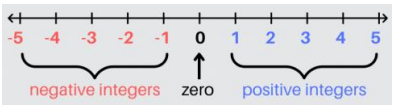
【1】进制与进制转换：二进制、八进制、十进制、十六进制

（一）数的分类（Types of Numbers）

1. 自然数（Natural Numbers）

- 自然数是我们用来数东西的正整数，**从 0 开始**：0，1，2，3，4.....
- 包含所有正整数和零，不包括负数。
- 用来计数，比如苹果有几个。
- 就是你每天用来数玩具、数糖果的数字，0、1、2、3，数到你有多少。

2. 整数（Integers）



整数包括所有**正整数、零和负整数**：...，-3，-2，-1，0，1，2，3，...

- 除了数东西用的正数，还包括表示“少了”的负数，比如欠朋友 3 颗糖就是 -3。

3. 有理数（Rational Numbers）

- 有理数是可以表示为两个整数**分数（Fraction）形式**的数，例如：1/2，-3/4，5，0（5/1, 0/1）
- 就像把蛋糕切成几块，吃了几块，数字表示“几分之几”。

4. 实数（Real Numbers）

实数包括所有**有理数和无限不循环小数**，比如：

- π（3.14159...），√2（约 1.414）
- 不仅有整数和分数，还有圆周率这样的特别数字。

（二）四则运算（Arithmetic Operations）

1. 加法（Addition，+）

- 定义：把两个数**合并**起来，求总数。
- 例子：3 + 2 = 5

2. 减法（Subtraction，-）

- 定义：求一个数比另一个数**少多少**。
- 例子：5 - 2 = 3

3. 乘法（Multiplication，×）

- 定义：把一个数**重复相加**几次。
- 例子：3 × 4 = 12（3 加 4 次，或 4 加 3 次）

4. 除法（Division，÷）

- 定义：把一个数**平均分成几份**。
- 例子：12 ÷ 3 = 4

5、简单 C++示例（演示基本算术运算）

```
#include <iostream>
using namespace std;
int main() {
    int a = 10; // 整数
```

```
    int b = 3;
    cout << "加法：" << a << " + " << b << " = " << a + b << endl;
    cout << "减法：" << a << " - " << b << " = " << a - b << endl;
    cout << "乘法：" << a << " * " << b << " = " << a * b << endl;
    cout << "除法：" << a << " / " << b << " = " << a / b << endl;
    cout << "余数：" << a << " % " << b << " = " << a % b << endl;
    return 0;
}
```

（三）进制与进制转换（Number Bases and Base Conversion）

1、什么是进制？（What is Number Base?）

进制是表示数字的方式，不同进制用不同的“基数”（base）表示数字。

- **十进制（Decimal，Base 10）**是我们日常生活最常用的，用 0~9 共 10 个数字表示。
- **二进制（Binary，Base 2）**只用 0 和 1 两个数字，计算机的语言。
- **八进制（Octal，Base 8）**用 0~7 共 8 个数字。
- **十六进制（Hexadecimal，Base 16）**用 0……9 和 A……F（10~15）共 16 个数字。

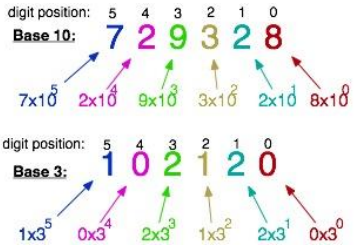
举例说明

- 进制就像不同的数字语言，十进制是我们平时用的数字，二进制是电脑用的语言。
- 就像不同国家用不同的字母或符号，数字也有不同的表达方法。

2、四种常见进制介绍（Common Number Bases）

（1）十进制（Decimal）

- **基数是 10**，数字用 0~9。
- 比如：数字“234”表示 2×10^2+3×10^1+4×10^0=200+30+4



（2）二进制（Binary）

- **基数是 2**，数字只有 0 和 1。

- 计算机内部数据全用二进制。
- 例子：二进制 101 表示 1×2^2+0×2^1+1×2^0=4+0+1=5。

（3）八进制（Octal）

- **基数是 8**，数字用 0~7。
- 例子：八进制 27 表示 2×8 ^1+7×8 ^0=16+7=23。

（4）十六进制（Hexadecimal）

- **基数是 16**，数字用 0……9 和 A……F（A=10, B=11,...F=15）。
- 例子：十六进制 1A 表示 1×16 ^1 +10×16 ^0 =16+10=26。

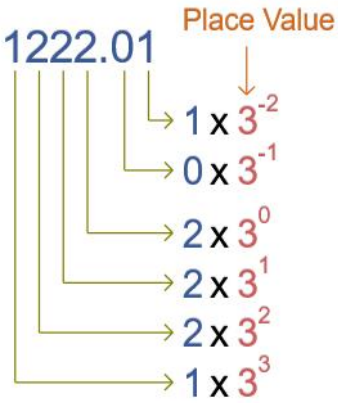
3、进制转换方法（How to Convert Between Bases）

（1）十进制转其他进制（Decimal to Other Base）

用**除基数取余法**：不断除以目标进制的基数，把余数倒着写。

（2）其他进制转十进制（Other Base to Decimal）

按位乘以基数的幂次相加。



（3）二进制与八进制、十六进制转换

- **二进制转八进制**：**每 3 位**二进制转成 1 位八进制。
- **二进制转十六进制**：**每 4 位**二进制转成 1 位十六进制。

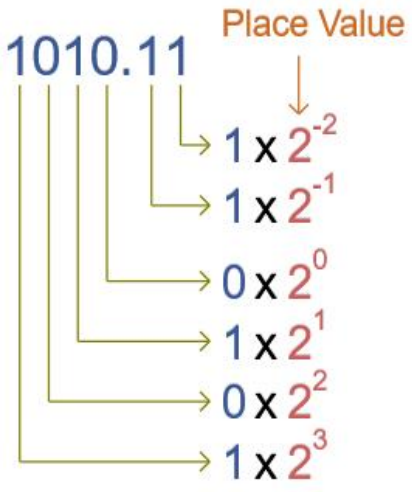
（4）C++代码示例：十进制转二进制

```
#include <iostream>
#include <string>
using namespace std;
string decimalToBinary(int n) {
    string binary = "";
    if (n == 0) return "0";
    while (n > 0) {
        binary = char((n % 2) + '0') + binary;
```

2	153	(Remainder)
2	76 - 1	(Remainder)
2	38 - 0	(Remainder)
2	19 - 0	(Remainder)
2	9 - 1	(Remainder)
2	4 - 1	(Remainder)
2	2 - 0	(Remainder)
	1 - 0	(Remainder)

```
            n /= 2;
        }
        return binary;
    }
    int main() {
        int num = 13;
        cout << "十进制 " << num << " 转二进制是 " << decimalToBinary(num) << endl;
    }
```

（5）C++代码示例：二进制转十进制




```
    #include <iostream>
    #include <string>
    using namespace std;
    int binaryToDecimal(string binary) {
        int decimal = 0;
        for (char bit : binary) {
            decimal = decimal * 2 + (bit - '0');
        }
        return decimal;
    }
    int main() {
        string binary = "1101";
        cout << "二进制 " << binary << " 转十进制是 " << binaryToDecimal(binary) << endl;
    }
```