

1. (20 分)

设计一个三维图形求体积和显示的应用，要求：

- 创建一个基类 `Shape`，
- 从 `Shape` 基类派生出两个子类：
 - `Sphere` (球)
 - `Cuboid` (长方体)，
- 从 `Cuboid` 类派生出一个子类 `Cube` (立方体)。

`main` 函数部分的代码如下：

```
int main() {
    Shape* shapes[3];

    shapes[0] = new Sphere(3);
    shapes[1] = new Cuboid(2, 3, 4);
    shapes[2] = new Cube(5);

    double totalVolume = 0.0;

    for (int i = 0; i < 3; ++i) {
        shapes[i]->display();
        totalVolume += shapes[i]->volume();
    }

    std::cout << "Total volume of all shapes: " << totalVolume << std::endl;

    // 请补充完整代码。

    return 0;
}
```

题目2：模板类实现矩阵操作（20 分）

要求：

设计一个模板类 `Matrix`，实现带有非类型形参 `Rows`（行数）和 `Cols`（列数）的矩阵类，要求：

1. “包含构造函数、析构函数；”
2. “能通过索引对指定的行列的元素进行修改；”
3. “能够进行矩阵相加操作；”
4. “能够实现矩阵输出，在屏幕上将矩阵的数据值进行输出。”

测试主函数如下：

```
int main() {
    Matrix<int, 2, 2> mat1;
    Matrix<int, 2, 2> mat2;

    try {
        mat1.at(0, 0) = 1; mat1.at(0, 1) = 2;
        mat2.at(1, 0) = 3; mat2.at(1, 1) = 4;

        Matrix<int, 2, 2> sum = mat1 + mat2;

        std::cout << "Matrix Addition Result:" << std::endl;
        sum.out();

        mat1.at(1, 2) = 3; // 越界测试
    } catch (exception &e) {
        cout << e.what();
    }

    return 0;
}
```

题目3：分数类的加法与约分运算（20 分）

要求：

编写一个分数类 `Fraction`，使其支持以下操作：

1. “能实现两个分数对象的加法；”
2. “构造函数中需实现约分处理（约成最简形式）；”
3. “实现两个分数的输出，格式为 `a/b`；”
4. “实现 `int` 与 `Fraction` 的加法，支持类型转换输出。”

主函数调用如下：

```
void main() {  
    cout << "test 1: ";  
    Fraction f1(1, 3), f2(7, 15);  
    cout << f1 << "+" << f2 << "=" << f1 + f2 << endl;  
  
    cout << "test 2: ";  
    Fraction f3(4, 5), f4(2, 5);  
    cout << f3 << "+" << f4 << "=" << f3 + f4 << endl;  
  
    cout << "test 3: ";  
    int i = 1;  
    Fraction f6 = i + f1;  
    cout << i << "+" << f1 << "=" << f6 << endl;  
  
    Fraction f7(2, 15), f8;  
    f8 = f4 + f7;  
    cout << f4 << "+" << f7 << "=" << f8 << endl;  
}
```