```java

import java.util.Scanner;

```

- 这行代码导入了 `java.util.Scanner` 类，用于读取控制台输入。

```java

public class MathExpression {

```

- 这是类声明的开始。`MathExpression` 是我们创建的类名。

```java

public static void main(String[] args) {

```

- 这是主方法的声明，它是程序执行的入口点。`String[] args` 是命令行参数的数组，但在这个程序中我们不使用它。

```java

Scanner scanner = new Scanner(System.in);

```

- 这里创建了一个 `Scanner` 对象，名为 `scanner`，用于从标准输入（控制台）读取数据。

```java

int N = Integer.parseInt(scanner.nextLine());

```

- 这行代码读取控制台输入的第一行，它应该是一个整数 `N`，表示接下来要读取的表达式的数量。`scanner.nextLine()` 读取整行输入，然后 `Integer.parseInt` 将字符串转换为整数。

```java

for (int i = 0; i < N; i++) {

```

- 这是一个 `for` 循环，从 `i = 0` 开始，一直到 `i < N`。循环的每一次迭代都处理一个数学表达式。

```java

String line = scanner.nextLine();

```

- 在循环内，这行代码读取下一行输入，这行输入应该包含一个数学表达式。

```java

String[] parts = line.split(" ");

```

- 使用 `split(" ")` 方法将输入行按空格分割成字符串数组 `parts`。这个数组应该包含三个元素：两个数字和一个运算符。

```java

if (parts.length != 3) {

System.out.println("Invalid input format");

continue;

}

```

- 这个 `if` 语句检查 `parts` 数组是否有三个元素。如果没有，输出错误消息，并使用 `continue` 跳到下一个循环迭代。

```java

try {

```

- `try` 块开始，用于捕获并处理可能发生的异常。

```java

int a = Integer.parseInt(parts[0]);

char operator = parts[1].charAt(0);

int b = Integer.parseInt(parts[2]);

```

- 这三行代码尝试从 `parts` 数组中解析两个整数 `a` 和 `b`，以及一个字符 `operator`。如果这些转换无效，将引发 `NumberFormatException`。

```java

int result = 0;

```

- 初始化一个整数 `result`，用于存储操作的结果。

```java

switch (operator) {

```

- `switch` 语句开始，基于 `operator` 变量的值选择执行不同的代码块。

```java

case '+':

result = a + b;

break;

```

- 如果 `operator` 是 `+`，执行加法操作，并将结果存储在 `result` 中。

```java

case '-':

result = a - b;

break;

```

- 如果 `operator` 是 `-`，执行减法操作。

```java

case '\*':

result = a \* b;

break;

```

- 如果 `operator` 是 `\*`，执行乘法操作。

```java

case '/':

if (b != 0) {

result = a / b;

} else {

System.out.println("Division by zero");

continue;

}

break;

```

- 如果 `operator` 是 `/`，执行除法操作。在除以 `b` 之前，先检查 `b` 是否为零，以避免除零错误。

```java

default:

System.out.println("Invalid operator");

continue;

}

```

- 如果 `operator` 不是有效的运算符，则输出错误消息并跳过当前循环迭代。

```java

System.out.println(result);

```

- 打印计算结果。

```java

} catch (NumberFormatException e) {

System.out.println("Invalid number format");

}