Java实验报告

软件学院 软件工程4班 2113850 李鹏

——实验题目：Java实现表达式求值及其异常处理

1. 简介

本实验针对表达式求值的功能进行了完善，主要包括以下内容：

* 设计异常处理机制，在合适的位置抛出和处理异常；
* 定义异常体系，并根据异常类型进行相应的处理；
* 记录计算日志和异常日志。
* 本实验以 Java 语言作为开发工具，通过编写代码实现表达式计算、异常处理和日志记录等功能。其中，初级表达式计算是指只包含数值和简单运算符（加、减、乘、除、取模）的表达式计算；带有变量定义的表达式计算是在初级表达式计算的基础上，增加了自定义变量的功能。

1. 异常处理
2. 异常定义方式和异常体系设计方式：

本实验设计了三种异常类型：

* ExpressionException：表达式错误异常；
* TypeDefineException：类型定义异常；
* UnsupportedOperatorException：不支持的操作异常。

这三种异常均继承自 Exception 类，在需要抛出异常时可以通过 throw 关键字将其抛出，并在需要处理异常时使用 try-catch 块进行捕获。

1. 异常处理和异常抛出：

在表达式计算过程中，可能会发生各种异常情况，例如表达式格式错误、类型定义错误等。这些异常需要在程序中进行捕获和处理，避免程序崩溃或计算结果错误。

在本实验中，异常处理主要包含以下几个方面：

* 在 judgeExpression() 方法中，检测括号是否匹配。如果不匹配，则抛出 ExpressionException 异常；
* 在 calculate() 方法中，对表达式进行解析和计算。如果表达式中包含未定义的变量，则抛出 TypeDefineException 异常；如果表达式使用了不支持的操作符，则抛出 UnsupportedOperatorException 异常。

3 异常体系设计：

在本实验中，异常体系设计如下：

异常体系设，ExpressionException、TypeDefineException 和 UnsupportedOperatorException 继承自 Exception 类，分别表示表达式错误、类型定义错误和不支持的操作错误。这些异常均具有 getMessage() 方法和 toString() 方法，可以在需要记录日志时输出异常信息。

1. 日志记录

除了异常处理之外，我们还需要记录计算日志和异常日志，以便在程序出现问题时排查错误。

* 设置日志级别为 INFO，即只记录一般信息和警告信息；
* 将日志输出到控制台和文件中，文件名为 calc.log；
* 对异常信息进行特殊处理，记录在异常日志中。

1. 日志设置方式

在 Java 中，java.util.logging 是一个标准的日志框架，无需额外引入依赖，可以直接使用。我们需要先对其进行相关配置，具体步骤如下：

1.1 创建 Logger 对象

首先，我们需要创建一个 Logger 对象，用于进行日志记录。通常情况下，我们会在类中定义一个静态变量来表示 Logger，然后在构造函数或静态代码块中进行初始化。例如：

import java.util.logging.Logger;

public class ExampleClass {

private static final Logger logger = Logger.getLogger(ExampleClass.class.getName());

public ExampleClass() {

// ...

}

// ...

}

在上述代码中，我们使用 getLogger() 方法创建了一个 Logger 对象，并指定了该类的名称。Logger 对象需要通过 getName() 方法获取其名称。

1.2 配置日志属性

在代码中，我们可以通过以下方式配置日志属性：

import java.util.logging.Level;

import java.util.logging.LogManager;

public class ExampleClass {

static {

LogManager.getLogManager().getLogger("").setLevel(Level.ALL);

}

public ExampleClass() {

// ...

}

// ...

}

在上述代码中，我们使用 LogManager.getLogManager() 方法获取 LogManager 对象，并通过 getLogger("") 获取根 Logger 对象。然后，我们通过 setLevel() 方法设置该对象的日志级别为 Level.ALL，表示记录所有级别的日志信息。

1. 日志记录方式

在代码中，我们可以使用 Logger 对象进行日志记录。Logger 对象提供了多个方法，用于记录不同级别的日志信息。常用的方法包括：

log(Level level, String msg)：输出指定级别的日志；

info(String msg)：输出 INFO 级别的日志；

warning(String msg)：输出 WARNING 级别的日志；

severe(String msg)：输出 SEVERE 级别的日志。

这些方法都接受一个 String 类型的参数 msg，表示要记录的日志信息。例如，我们可以在程序中加入如下代码：

java

logger.log(Level.SEVERE, "This is a severe message");

logger.info("This is an info message");

logger.warning("This is a warning message");

当程序运行时，会将以上内容输出到配置文件中指定的目的地中（控制台或文件）。