

Спецификация к лабораторной работе №2

1. Общая информация

1.1. Класс `ArithmeticExpression` осуществляет расшифровку и вычисление корректно записанного арифметического выражения. Выражение передается в конструктор в виде строки символов.

1.2. Класс `InvolutionExpression` наследуется от класса `ArithmeticExpression` и производит операцию возведения числа в степень. Основание и степень передаются в конструктор в виде двух чисел: `double` и `int` соответственно, либо через бинарный файл, созданный отдельной программой, написанной на языке C, в этом случае в конструктор передается имя файла.

Язык реализации — Java.

2. Интерфейс

2.1. `ArithmeticExpression`

2.1.1. `ArithmeticExpression(double constant)` — инициализирует класс как константу

2.1.2. `ArithmeticExpression(String expression)` — инициализирует класс арифметическим выражением

2.1.2.1. throws `Throwable`

2.1.3. `calculate()` - вычисляет выражение, переданное классу в конструкторе, либо возвращает переданную в конструкторе константу

2.1.3.1. throws `Throwable` — сообщение об ошибке при вычислении выражения

2.2. `InvolutionExpression`

2.2.1. `InvolutionExpression(String fileName)` — инициализирует класс значениями из бинарного файла `fileName`

2.2.1.1. throws `FileNotFoundException`, `IOException`, `Throwable`

2.2.2. `InvolutionExpression(double constant)` — наследует `ArithmeticExpression(double constant)` без изменений

2.2.3. `InvolutionExpression(double base, int power)` — инициализирует класс значениями основания и степени числа

2.2.4. `calculate()` — унаследованный метод

3. Функциональные требования

3.1. `ArithmeticExpression`

3.1.1. Если при инициализации выражение содержит недопустимые символы (любые символы кроме ASCII 40-43, 45-57), либо символы располагаются в недопустимой последовательности (два арифметических знака подряд, открывающая скобка сразу после цифры, и т. д.), то конструктор бросает исключение типа `Throwable` с сообщением «Incorrect expression»

3.1.2. Если вычисленное значение правой части выражения деления по модулю меньше или равно минимальному положительному значению типа `double`, то функция бросает исключение типа `Throwable` с сообщением «Calculation error: division by zero»

3.1.3. Если введенное выражение корректно и при вычислении не происходит ошибок, то вычисление должно возвращать корректный результат.

3.2 `InvolutionExpression`

3.2.1. Если при инициализации указанного файла не существует, он является директорией или не может быть открыт для чтения по какой-либо иной причине, то конструктор должен бросить исключение типа `FileNotFoundException`

3.2.2. Если при чтении файла возникает ошибка ввода-вывода, то конструктор должен бросить исключение типа `IOException`

3.2.3. Если файл существует и корректно читается, либо инициализация производилась двумя числами напрямую, то вычисление должно возвращать корректный результат

4. Реализация

4.1. `ArithmeticExpression`

При инициализации выражением производится проверка на наличие недопустимых символов и на корректность выражения, после чего выражение делится на два по одной из арифметических операций верхнего уровня. При этом предпочтение отдается операциям сложения и вычитания. В классе сохраняется тип операции, по которой проходит разделение и создается два объекта `ArithmeticExpression`, инициализируемые левой и правой частью выражения соответственно. Если выражение состоит из одного числа (возможно наличие скобок), то класс инициализируется константой. Таким образом, выражение рекурсивно разбирается в бинарное дерево. Вычисление проводится в обратной последовательности.

4.2. `InvolutionExpression`

Инициализация может проводиться передачей основания и степени непосредственно, либо через бинарный файл. Во втором случае значения извлекаются из файла и инициализация продолжается по первому варианту. Выражение инициализируется как константа (при степенях 0 и 1), либо как выражение произведения с константой в левой части и возведением числа в степень в правой (со степенью на единицу меньше)