|  |  |  |
| --- | --- | --- |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное автономное  образовательное учреждение высшего образования  «Пермский государственный национальный  исследовательский университет» | | |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по индивидуальной работе №2  по дисциплине «Язык программирования Python»  Вариант 1 | | |
|  | | Работу выполнила  студентка группы ИТ-6 1 курса  Булгакова Ю.О  «25» июня 2025 г. |
| Работу проверил  Рубцова М.Б  «\_\_\_» июня 2025 г. |
| Пермь 2025 | | |

СОДЕРЖАНИЕ

[Постановка задачи 3](#_Toc201788130)

[Алгоритм решения 4](#_Toc201788131)

[1. Разработка алгоритма 4](#_Toc201788132)

[1.1 Поддержка вложенных выражений 4](#_Toc201788133)

[1.2 Проверка корректности ввода 4](#_Toc201788134)

[1.3 Интерактивный интерфейс 4](#_Toc201788135)

[2. Выбранные структуры данных 4](#_Toc201788136)

[2.1 Класс Stack (стек) 5](#_Toc201788137)

[2.2 Класс ExpressionCalculator 5](#_Toc201788138)

[3. Логика работы 5](#_Toc201788139)

[Тестирование 7](#_Toc201788140)

[Код программы 10](#_Toc201788141)

[Инструкция по применению стилей и оформлению работы 11](#_Toc201788142)

# Постановка задачи

Используя структуру стека, вычислить выражение, содержащее две операции: поиск минимума (обозначается m(,)) и поиск максимума (обозначается М(,)). Операции могут быть вложенными, например, M(15, m(16,8)) (в данном выражении ищем минимум из 16 и 8, а потом ищем максимум от результата m(16,8) и 15, ответ 15). В качестве аргументов могут использоваться только целые положительные числа. Строка, вводимая пользователем, корректна и не содержит пробелов.

# Алгоритм решения

1. Разработка алгоритма  
Требовалось разработать программу-калькулятор, способную вычислять математические выражения с операциями нахождения минимума (min) и максимума (max). Программа должна удовлетворять следующим требованиям:

## 1.1 Поддержка вложенных выражений

Обработка сложных конструкций, например: M(15, m(16, 8)) → 15.

Корректная интерпретация приоритета операций.

## 1.2 Проверка корректности ввода

Обработка ошибок:

Неправильное количество аргументов (например, m(5,)).

Недопустимые символы (например, k(1, 2)).

Несбалансированные скобки (например, m(5,10).

## 1.3 Интерактивный интерфейс

Возможность ввода выражений вручную.

Вывод результата или сообщения об ошибке.

Команда для выхода из программы (выход, exit, quit)

# **2.** Выбранные структуры данных

Для реализации калькулятора использованы:

## 2.1 Класс Stack (стек)

Основан на списке Python (self.\_items).

Операции:

push(item) – добавление элемента в конец списка.

pop() – удаление и возврат последнего элемента.

peek() – просмотр верхнего элемента без удаления.

is\_empty() – проверка на пустоту.

size() – количество элементов.

Обоснование выбора:

Стек идеально подходит для обработки вложенных выражений благодаря принципу LIFO (Last In, First Out).

Позволяет корректно обрабатывать скобки и порядок операций.

## 2.2 Класс ExpressionCalculator

Содержит словарь операций: {'m': min, 'M': max}.

Методы:

\_apply\_operation(stack) – извлекает аргументы из стека, проверяет их количество и применяет операцию.

calculate(expression) – парсит выражение, используя стек, и возвращает результат.

# 3. Логика работы

Разбор выражения:

Посимвольный обход строки.

Игнорирование пробелов и запятых.

Обработка чисел (последовательность цифр → целое число).

Фиксация операций (m или M) и скобок.

Вычисление вложенных операций:

При обнаружении ) извлекаются аргументы до ближайшей (.

Проверяется, что аргументов ровно 2.

Применяется соответствующая операция (min или max).

Обработка ошибок:

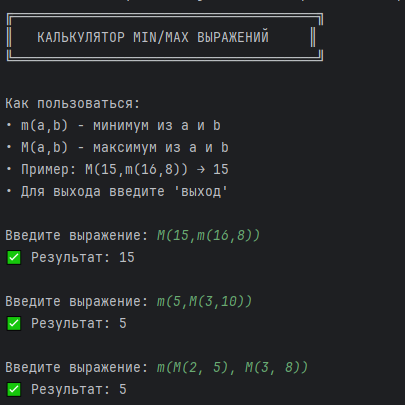
Неправильное количество аргументов → ValueError.

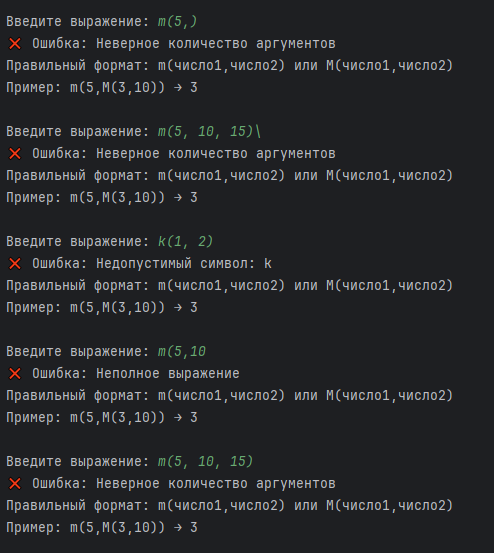
Неизвестная операция → ValueError.

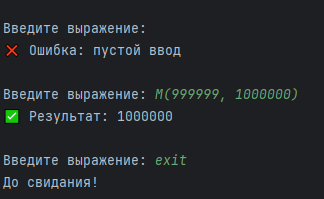
Недопустимые символы → ValueError.

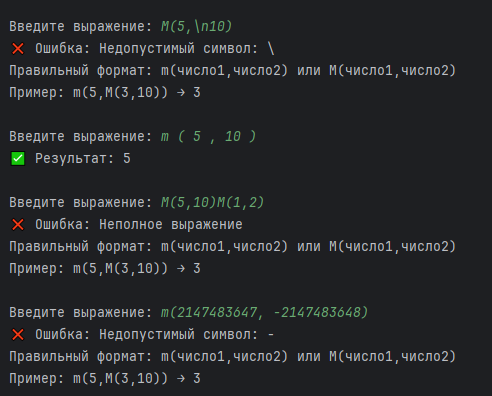
Несбалансированные скобки → ValueError.

# Тестирование

Корректные выражения:  
 

Ошибочные выражения:  
 

Граничные случаи:  
 

Особые случаи:  
 

# Код программы

Код

# Инструкция по применению стилей и оформлению работы

Для оформления частей отчёта следует использовать заранее созданные стили. Все стили, которые могут пригодиться, начинаются с «ЛР. 1 …».



**ЛР. 1 Обычный** – для оформления текста задания и алгоритма решения.

**ЛР. 1 Текст программы** – для оформления кода программы.

**ЛР. 1 Заголовок 1** – заголовок первого уровня (для того, чтобы озаглавить основные разделы отчета).

**ЛР. 1 Заголовок 2** – заголовок второго уровня (для того, чтобы озаглавить подразделы).

Для того, чтобы перенести текст следующего блока на другую страницу, необходимо воспользоваться инструментов «Разрыв страницы» в разделе «Вставка».

