1. **NullPointerException**: Напишите программу, которая вызывает метод на объекте, который равен **null**, и обработайте **NullPointerException**.
2. **ArrayIndexOutOfBoundsException**: Попробуйте обратиться к элементу массива с индексом, который выходит за границы массива.
3. **ClassCastException**: Создайте список объектов разных типов и попробуйте привести их к неверному типу, затем обработайте **ClassCastException**.
4. **IllegalArgumentException**: Напишите метод, который ожидает положительное число в качестве аргумента, и передайте ему отрицательное число.
5. **UnsupportedOperationException**: Используйте методы, которые не поддерживаются в неизменяемой коллекции.
6. **InterruptedException**: Запустите поток и попробуйте прервать его с помощью **Thread.interrupt()**.
7. **StackOverflowError**: Создайте рекурсивный метод, который вызывает себя без условия завершения.
8. **OutOfMemoryError**: Создайте программу, которая попытается выделить большое количество памяти (например, создав большой список), чтобы вызвать **OutOfMemoryError**.
9. **Custom Exception**: Создайте собственное исключение, например, **MyCustomException**, и бросьте его в своей программе.
10. **ArithmeticException**: Разработайте калькулятор, который делит число на ноль, и обработайте **ArithmeticException**. Through classes (

- class Calculator,

- каждая операция отдельный класс,

- нужно гарантировать что каждый класс имеет метод perform(Integer a, Integer b),

- реализовать метод reset() который сбрасывает вычисление на 0

- в main запрашиваем операцию (multiply(\*), div(/), sum(+), sub(-)

- реализовать обработку деления на 0 при div

Для каждой из этих задач предоставьте соответствующий код, который вызывает и обрабатывает исключение. Это поможет вам понять, как обрабатывать разные типы исключений в Java.