

Исключения в программировании и их обработка (семинары)

Урок 1. Обработка ошибок в программировании

Методы с исключениями

Реализуйте 3 метода, чтобы в каждом из них получить разные исключения.

- Метод `arrayOutOfBoundsException` - Ошибка, связанная с выходом за пределы массива
- Метод `divisionByZero` - Деление на 0
- Метод `NumberFormatException` - Ошибка преобразования строки в число

Важно: они не должны принимать никаких аргументов

Разность двух массивов

Реализуйте метод `subArrays`, принимающий в качестве аргументов два целочисленных массива `a` и `b`, и возвращающий новый массив `c`, каждый элемент которого равен разности элементов двух входящих массивов в той же ячейке.

Если длины массивов не равны - верните нулевой массив длины 1.

Напишите свой код в методе `subArrays` класса `Answer`. Метод `subArrays` принимает на вход два параметра:

- `int[] a` - первый массив
- `int[] b` - второй массив

Пример

```
a = new int[]{4, 5, 6};  
b = new int[]{1, 2, 3};
```

Вывод: `[3, 3, 3]`

```
a = new int[]{4, 5, 6};  
b = new int[]{1, 2, 3, 5};
```

Вывод: `[0]`

Частное двух массивов

Реализуйте метод `divArrays`, принимающий в качестве аргументов два целочисленных массива `a` и `b`, и возвращающий новый массив `c`, каждый элемент которого равен частному элементов двух входящих массивов в той же ячейке.

Если длины массивов не равны - верните нулевой массив длины 1.

Важно: При выполнении метода единственное исключение, которое пользователь может увидеть - `RuntimeException`, т.е. ваше.

Напишите свой код в методе `divArrays` класса `Answer`. Метод `divArrays` принимает на вход два параметра:

- `int[] a` - первый массив
- `int[] b` - второй массив

Пример

```
a = new int[]{12, 8, 16};  
b = new int[]{4, 2, 4};
```

Вывод: `[3, 4, 4]`

```
a = new int[]{12, 8, 16, 25};  
b = new int[]{4, 2, 4};
```

Вывод: `[0]`

Урок 2. Исключения и их обработка

Float

Напишите программу, которая проверяет, является ли введенная текстовая строка числом с плавающей точкой (`float`).

Программа пытается преобразовать введенную строку в число `float`, и если это успешно, она выводит полученное число.

Если преобразование не удалось, программа выдаёт сообщение об ошибке и возвращает специальное значение `Float.NaN`, чтобы обозначить ошибку.

На входе:

'3.14'

'Ivan'

На выходе:

3.14

Деление элементов массива на число

Напишите программу, которая вычисляет значение выражения `intArray[8] / d`, где `intArray` - массив целых чисел, а `d` - делитель.

Программа проверяет, имеется ли в массиве `intArray` элемент с индексом 8, и если нет, выводит сообщение о невозможности выполнения операции.

Также программа проверяет, равен ли делитель `d` нулю, и если да, выводит соответствующее сообщение.

На входе:

'1 2 3 4 5 6 7 8 9'

'1'

На выходе:

`intArray[8] / d = 9 / 1 = 9.0`

9.0

Деление

Напишите программу для выполнения арифметической операции деления двух целых чисел `a` и `b`.

При этом программа должна проверить, что делитель `b` не равен нулю, и выполнить деление только в этом случае.

Если `b` равен нулю, программа должна вывести сообщение о невозможности выполнения операции и вернуть результат равный нулю.

После выполнения операции деления, программа также должна вывести сумму чисел `a` и `b` с помощью метода `printSum`.

Если аргументы не переданы через командную строку, используйте значения по умолчанию.

На входе:

'12'

'5'

На выходе:

17

2.4

Символ 'а'

Напишите программу, которая принимает символ `a` в качестве аргумента и выполняет следующую проверку:

если символ `a` равен пробелу " ", программа должна выбрасывать исключение с сообщением "Пустая строка введена."

В противном случае программа должна возвращать сообщение "Ваш ввод - [символ]", где [символ] заменяется на введенный символ `a`.

На входе:

'0'

На выходе:

Result: Your **input** was - 0

Урок 3. Продвинутая работа с исключениями в Java

Напишите приложение, которое будет запрашивать у пользователя следующие данные в произвольном порядке, разделенные пробелом:
Фамилия Имя Отчество датарождения номертелефона пол

Форматы данных:

фамилия, имя, отчество - строки

датарождения - строка формата dd.mm.yyyy

номертелефона - целое беззнаковое число без форматирования

пол - символ латиницей f или m.

Приложение должно проверить введенные данные по количеству. Если количество не совпадает с требуемым, вернуть код ошибки, обработать его и показать пользователю сообщение, что он ввел меньше и больше данных, чем требуется.

Приложение должно попытаться распарсить полученные значения и выделить из них требуемые параметры. Если форматы данных не совпадают, нужно бросить исключение, соответствующее типу проблемы. Можно использовать встроенные типы java и создать свои. Исключение должно быть корректно обработано, пользователю выведено сообщение с информацией, что именно неверно.

Если всё введено и обработано верно, должен создаваться файл с названием, равным фамилии, в него в одну строку должны записаться полученные данные, вида

<Фамилия><Имя><Отчество><датарождения> <номертелефона><пол>

Однофамильцы должны записаться в один и тот же файл, в отдельные строки.

Не забудьте закрыть соединение с файлом.

При возникновении проблемы с чтением-записью в файл, исключение должно быть корректно обработано, пользователь должен увидеть стектрейс ошибки.