

Домашнее задание

по теме «Ошибки компиляции и ошибки выполнения»

Формулировка задания:

Задача: Форматирование ввода числа

Ошибки случаются, и нужно уметь с ними работать. На проекте разработали библиотеку, которая обрабатывает ввод числа в формате String и возвращает значение Integer или Long. Вас попросили написать класс-преобразователь, который включает в себя методы со следующим функционалом:

- возвращает число, если всё корректно;
- генерирует ошибку, если ввод не является числом в десятичной системе счисления.

Подробное описание задачи:

1. Создать папку utils, где будут размещен класс для обработки ввода данных.
2. Напишите в классе функцию parseCount.
 - a. Аргументом функции является строковое значение, которое необходимо распарсить.
 - b. Для парсинга используйте parse из класса Integer.
 - c. Если в результате парсинга возникает ошибка, то выбрасывайте исключение с ошибкой «Невалидное значение».
 - d. Верните результат парсинга из функции.
3. Напишите в классе функцию validateCount.
 - a. Аргументом функции является значение, которое необходимо распарсить.
 - b. Попробуйте распарсить значение с помощью функции parseCount (пункт 2 текущего задания).
 - c. Если распарсить удаётся успешно, то возвращайте результат.
 - d. Перехватывайте исключение, которое может выбрасывать функция

parseCount.

е. Возвращайте ошибку из функции в случае перехвата исключения.

4. Дополнительно. Напишите в классе функцию parseNumber.

а. Аргументом функции является значение, которое необходимо распарсить.

б. Для парсинга используйте parse из класса Double.

с. Если в результате парсинга возникает ошибка, то выбрасывайте исключение с ошибкой «Невалидное значение».

д. Верните результат парсинга из функции.

5. Дополнительно. Напишите в классе функцию validateNumber. Функционал будет похож на validateCount, только функция будет работать с parseNumber.

После разработки библиотеки, смержите в ветку задачу с покупками (Домашнее задание по теме «Работа с классами и объектами. IO»). Добавьте папку utils в задачу. Проверьте функционал на вводе дробных и целых чисел.

Программа реализуется в отдельной ветке git homeworks/homework013. При сохранении состояния программы (коммиты) пишется сообщение с описанием хода работы по задаче.

В корне папки с программой должен быть файл .gitignore.

Программа локально коммитится и публикуется в репозиторий GitHub на проверку.

Дополнительный функционал

1. Используйте интерфейс ByCondition из Домашнего задания по теме: «Абстрактные классы и интерфейсы». Доработайте интерфейс на работу с Generics. Метод будет принимать обертки Integer, Double и возвращать boolean.

2. Создайте второй класс в папке utils. Сделать класс наподобие класса Sequence из Домашнего задания по теме: «Абстрактные классы и интерфейсы». Функционал класса — отклонять экземпляры Integer, Double, если они не

соответствуют условию.

3. Проверить работу класса на задаче покупателей. Отклонять условия «Отрицательное число», «Нет возможности купить товар».

Программа реализуется в отдельной ветке git homeworks/homework013Addition. При сохранении состояния программы (коммиты) пишется сообщение с описанием хода работы по задаче.

В корне папки с программой должен быть файл .gitignore.

Программа локально коммитится и публикуется в репозиторий GitHub на проверку.

Планируемый результат:

1. Ссылка на программу в репозитории GitHub;
2. Отчёт со скринами выполнения задач - постановка задачи, код задачи и результат в консоли IntelliJ Idea.

Описания плана работы:

Обработка исключений, работа с данными в Java, вынесение функционала проверок в отдельные классы.

Перечень инструментов, необходимых для реализации деятельности:

Персональный компьютер, JDK 17 (либо OpenJDK 17), IntelliJ Idea для разработки на Java, GIT, Tortoise GIT.