#### Цель практической работы

Научиться:

* использовать механизм исключений с помощью их создания и «отлова» в коде;
* писать конфигурацию для логирования событий при исполнении программы.

#### Что нужно сделать

Выполните задание в репозитории java\_basics в проекте ExceptionsDebuggingAndTesting/homework\_1/ConsoleCustomerList.

1. Проект ConsoleCustomerList — консольное приложение, в котором можно хранить список клиентов. Запустите его, введите в консоль слово help и нажмите Enter. Программа выведет список доступных команд и примеры их выполнения.
2. Поэкспериментируйте с программой: попробуйте ввести разные значения и команды, которые не будут соответствовать примерам. Добейтесь того, чтобы программа «вылетела», то есть произошёл Exception.
3. Напишите в классе CustomerStorage проекта ConsoleCustomerList все варианты защиты от некорректных данных, которые вам удастся придумать и обнаружить. Создайте собственные классы исключений, которые должны выбрасываться:
   * при некорректном количестве компонентов в переданной строке с данными;
   * при неверном формате номера телефона;
   * при неправильном формате email.
4. Защитите программу от преждевременного завершения метода addCustomer() — напишите код, который будет «отлавливать» исключения и выводить эту информацию в консоль в понятном пользователю виде и в лог (см. ниже).
5. Подключите библиотеку Log4j2 к проекту:

<dependency>

<groupId>org.apache.logging.log4j</groupId>

<artifactId>log4j-api</artifactId>

<version>2.18.0</version>

</dependency>

<dependency>

<groupId>org.apache.logging.log4j</groupId>

<artifactId>log4j-core</artifactId>

<version>2.18.0</version>

</dependency>

Пример использования библиотеки:

import org.apache.logging.log4j.\*;

public class Main {

private static final Logger logger =

LogManager.getLogger(Main.class);

public static void main(String[] args) {

logger.log(Level.WARN, "Сообщение об ошибке");

}

}

1. Сделайте два отдельных лога в папке logs проекта с помощью Log4j2 и настройте конфигурацию библиотеки Log4j следующим образом:
   * **logs/queries.log** — заполняется информацией обо всех запросах к приложению;
   * **logs/errors.log** — заполняется информацией обо всех ошибках (возникших исключениях со всеми деталями).

Подробнее о том, как настраивать конфигурацию, читайте [в материале](https://www.notion.so/9b1faeff94634d758fb558ac6fc9d2a9).

#### Критерии оценки

**Принято:**

* Метод реализован.
* Код метода чистый.
* Тесты выполняются успешно.
* Логи пишутся корректно.
* В каждый лог-файл выводится только соответствующий тип информации.

**На доработку:**

* Задание не выполнено.

#### Материалы для изучения на русском языке

* [Исключения в Java. Часть 1](https://skillbox.ru/media/base/isklyucheniya_v_java_chast_1/) – https://skillbox.ru/media/base/isklyucheniya\_v\_java\_chast\_1/
* [Исключения в Java. Часть 2](https://skillbox.ru/media/base/isklyucheniya-v-java-chast-2/) – https://skillbox.ru/media/base/isklyucheniya-v-java-chast-2/
* [Исключения в Java: catch под лупой. Часть 3](https://skillbox.ru/media/base/isklyucheniya-v-java-catch-pod-lupoy-chast-3/) - https://skillbox.ru/media/base/isklyucheniya-v-java-catch-pod-lupoy-chast-3/
* [Что спрашивают на собеседованиях об исключениях](https://javastudy.ru/interview/exceptions/) - https://javastudy.ru/interview/exceptions/