АС «**Twitter-Mini**» версия 2.0

**Задача**: дополнить функционал АС «Twitter-mini».

**Список дополнений**

**1) Избавиться от хранения данных внутри системы в виде List<>.**

Для этого наши классы, которые до этого хранили в себе данные, должны научиться работать с файлами. Данные будут храниться в 2 разных файлах: UserData.txt и PostData.txt .

**Формат хранения данных в UserData.txt:**

{id}{login}{password}{registrationDate}{type}{name}{surname}{dateOfBirth}{occupation}{foundingDate}

Одна строка — один пользователь.

Пример строки в файле для **человека**:

{1}{vadim}{12345}{2023-04-12}{PERSON}{Vadim}{Maidanov}{1995-04-01}{}{}

Пример строки в файле для **организации**:

{2}{codify}{1234}{2023-04-12}{ORGANIZATION}{CodifyAcademy}{}{}{Образовательные курсы}{2017-01-01}

**Формат хранения данных в PostData.txt:**

{id}{authorLogin}{theme}{message}{tags}{createdAt}

Одна строка — один пост.

Пример строки в файле для **публикации**:

{1}{vadim}{Первая публикация}{Моя первая публикация!!!}{впервые,публикация}{2023-04-12}

На старте, система считывает данные с файлов в нужные List<>. Дальше продолжает работу с List<>. В случае добавления пользователя или публикации, запись добавляется сначала в List<>, затем в файл. **Файл должен содержать актуальные данные обязательно.** Работоспособность проверить с помощью перезапуска системы, внесенные ранее данные должны сохраниться и должно быть возможно продолжение работы с системой.

**2) Добавить валидацию данных.**

Список валидаций:

1. **Логин пользователя** должен состоять только из латинских букв, не содежрать пробелов, цифр и спецсимволов. Не должен быть короче 3 символов и длиннее чем 10 символов.
2. **Пароль пользователя** должен состоять только из латинских букв, не должен содержать пробелов. Минимальная длина пароля 5 символов, обязательное использование: хотя бы одна заглавная буква, хотя бы одна цифра и хотя бы один спецсимвол. Пример: 12345 — неправильный пароль, 1Vad!m — правильный пароль.
3. **Дата рождения человека.** Возраст регистрируемого не должен быть менее 10 лет. К примеру дата рождения в день регистрации 2023-04-12 не должна быть позже, чем 2013-04-12.
4. **Имя, фамилия, наименование организации, род деятельности организации** не должны быть пустыми.
5. **Дата основания организации** не должна быть позже, чем текущая дата — 1 месяц.
6. **Тема публикации** не должна быть пустой и содержать менее 5 символов.
7. **Текст публикации** не должен быть пустым и содержать более 255 символов.
8. **Тег публикации** не должен содержать пробелов, иметь менее 3 символов и иметь более 15 символов.

**3) Добавить обработку исключительных ситуаций.**

Добавить обработку всех исключительных ситуаций с помощью блока try{}catch{}. Описать пакет exception, где будут собственные Exception для обработки каждой исключительной ситуации возникающей при валидации данных. Также добавить обработку исключительных ситуаций, которые могут возникнуть в системе(чтение файла, добавление публикации незарегистрированному пользователю и т. д.).

**4) Добавить новые команды: read\_users и read\_posts.**

Команда **read\_users** должна считывать пользователей из указанного файла и в случае если этого пользователя нет в системе, регистрировать его. Пример команды:

*Введите команду: read\_users*

*Файл для чтения: UserRegistry\_2023-04-12.txt*

*Считывание пользователей…*

*Всего пользователей в файле: 7, из них были добавлены: 4.*

Команда **read\_posts** должна считывать публикации из укзанного файла и добавлять их все в систему. Пример команды:

*Введите команду: read\_posts*

*Файл для чтения: PostRegistry\_2023-04-12.txt*

*Считывание публикаций…*

*Было добавлено: 10 публикаций.*

Формат файлов такой же, как и формат файлов UserData.txt и PostData.txt.

Валидации:

1. Расширение файла должно быть \*.txt.
2. Файл не должен быть пустым.
3. Формат данных в файле должен совпадать с шаблоном.

Если валидация не проходит, описать и обработать соответствующий Exception.

**Считывание данных из файла должно происходить в отдельном потоке**.

Данные из файла попадают сначала в List<>, затем переносятся в файл для хранения данных.