**Проект по теме «Создание графического**

**приложения c пользовательским интерфейсом на JavaFX»**

**Указания к выполнению задания:**

* Все классы должны удовлетворять[**Code Conventions for the Java Programming Language**](http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html) и принципам **SOLID**

Задание:

1. Создать модель данных **User** с полями: идентификатор (целое число), имя пользователя, дата регистрации (**dd.MM.yyyy HH:mm:ss**), электронная почта, возраст, страна, отправлено ли письмо (название поля: **isSend** - **true/false, исключить** данное поле из **json,** как на загрузку, так и на выгрузку)
2. Создать **репозиторий SendUserIdRepository**, конструктор которого **не** **принимает** на вход **аргументов**, производит загрузку данных в список целых чисел из файла формата **JSON** **send.json**. Имя данного файла сделать статической константой в классе **Constants** пакета **util**. **Файл** вручную создавать **не надо**, изначально его вообще не будет существовать, ваша программа в будущем сама будет его создавать в случае отсутствия и добавлять в него нужные ей данные в п. 6. Исключение в конструкторе об отсутствии файла необходимо заигнорировать и не перебрасывать в сигнатуру
3. Создать **репозиторий**, конструктор которого принимает на вход объект типа **File** и производит инициализацию списка пользователей из этого файла. Далее в этом конструкторе проставить для каждого объекта в поле **isSend** значение true, если его id есть в списке репозитория **SendUserIdRepository**
4. Произвести **отображение** всех пользователей в таблице по нажатию на опцию

**File->Open**. Выбрать созданный **вручную** файл с пользователями. Программа должна **отметить** галочкой в первой колонке таблицы тех пользователей, кому были уже **отправлены** письма. Саму галочку сделать неизменяемой, написав следующий код в блоке инициализации после поля с CheckBox-ом:

{  
 btn.setDisable(true);  
 btn.setStyle("-fx-opacity: 1;");  
}

1. Для **каждой** **строки** таблицы в крайней колонке должна быть кнопка «**Отправить»,** по нажатию на которую происходит открытие нового окна с ожиданием закрытия и **передачей** данных о пользователе, к которому она относится
2. В **новом** окне сделать поля для ввода **темы** письма(textField) и **текста** письма(textArea). По нажатию на кнопку «Отправить» в этом окне произвести отправку письма пользователю и вывода сообщения об успехе, сохранив id этого пользователя в файл **send.json** через репозиторий. Для **отправки** **сообщения** использовать код из приложения к проекту
3. **После** закрытия окна с отправлением сообщения необходимо произвести **инициализацию** **таблицы** заново для того, чтобы галочка об отправке **отобразилась** в первой колонке автоматически