Санкт-Петербургский Национальный

Исследовательский Университет

Информационных Технологий, Механики и Оптики

Лабораторная работа №8

по дисциплине

«Программирование»

Вариант - 25904

Выполнил: Давтян Давид Арменович

Санкт-Петербург

2019 г.

**ЗАДАНИЕ**

Доработать программу из [ЛР №7](https://se.ifmo.ru/courses/programming" \l "lab7) следующим образом:

Заменить консольный клиент на клиент с графическим интерфейсом пользователя(GUI).   
В функционал клиента должно входить:

1. Окно с авторизацией/регистрацией.
2. Отображение текущего пользователя.
3. Таблица, отображающая все объекты из коллекции
   1. Каждое поле объекта - отдельная колонка таблицы.
   2. Строки таблицы можно фильтровать/сортировать по значениям любой из колонок. Сортировку и фильтрацию значений столбцов реализовать с помощью Streams API.
4. Поддержка всех команд из предыдущих лабораторных работ.
5. Область, визуализирующую объекты коллекции
   1. Объекты должны быть нарисованы с помощью графических примитивов с использованием [Graphics](https://docs.oracle.com/javase/8/docs/api/java/awt/Graphics.html), [Canvas](https://docs.oracle.com/javase/8/javafx/api/javafx/scene/canvas/Canvas.html) или аналогичных средств графической библиотеки.
   2. При визуализации использовать данные о координатах и размерах объекта.
   3. Объекты от разных пользователей должны быть нарисованы разными цветами.
   4. При нажатии на объект должна выводиться информация об этом объекте.
   5. При добавлении/удалении/изменении объекта, он должен **автоматически** появиться/исчезнуть/измениться  на области как владельца, так и всех других клиентов.
   6. При отрисовке объекта должна воспроизводиться согласованная с преподавателем **анимация**.
6. Возможность редактирования отдельных полей любого из объектов (принадлежащего пользователю). Переход к редактированию объекта возможен из таблицы с общим списком объектов и из области с визуализацией объекта.
7. Возможность удаления выбранного объекта (даже если команды remove ранее не было).

Перед непосредственной разработкой приложения **необходимо** согласовать прототип интерфейса с преподавателем. Прототип интерфейса должен быть создан с помощью средства для построения прототипов интерфейсов (mockplus, draw.io, etc.)

1. Интерфейс должен быть реализован с помощью библиотеки JavaFX
2. Графический интерфейс клиентской части должен поддерживать **русский**, **румынский**, **итальянский** и **испанский (Доминиканская Республика)** языки / локали. Должно обеспечиваться корректное отображение чисел, даты и времени в соответстии с локалью. Переключение языков должно происходить без перезапуска приложения. Локализованные ресурсы должны храниться в **файле свойств**.

**Вопросы к защите лабораторной работы:**

1. Компоненты пользовательского интерфейса. Иерархия компонентов.
2. Базовые классы Component, Container, JComponent.
3. Менеджеры компоновки.
4. Модель обработки событий. Класс-слушатель и класс-событие.
5. Технология JavaFX. Особенности архитектуры, отличия от AWT / Swing.
6. Интернационализация. Локализация. Хранение локализованных ресурсов.
7. Форматирование локализованных числовых данных, текста, даты и времени. Классы NumberFormat, DateFormat, MessageFormat, ChoiceFormat

**ВОПРОСЫ К ЗАЩИТЕ**

1. Классы для представления даты и времени из пакета java.time. Преобразование темпоральных величин.
2. Взаимодействие с базами данных. Протокол JDBC.
3. JavaMail API. Протоколы SMTP, IMAP, POP3.
4. Шаблоны проектирования. GoF-паттерны.

**ИСХОДНЫЙ КОД**

<https://github.com/david-d25/programming-lab8>

**КАК ЗАПУСТИТЬ**

**Сборка:** bash build.sh

**Генерация javadoc:** bash build-javadoc.sh

**Запуск:** bash run-server.sh

bash run-client.sh

**Очистка:** bash clear.sh

**КАК ЗАПУСТИТЬ НА HELIOS**

**Введите ‘18’ после нужного сценария.**

**Например:** bash build.sh 18

**ДИАГРАММА КЛАССОВ**

**ВЫВОД**

Спустя много дней и ночей, проведённых за этой лабораторной, я осознал, что мне следует тщательнее подбирать структуру приложения, чтобы не пожалеть потом об этом. Ну, и ещё я познакомился с базами данных. Не сказал бы, что эти громадные конструкции красивые, но, по крайней мере, они безопасны.