Факультет программной инженерии и компьютерной техники

Программирование

Лабораторная работа №8

Вариант 311301

Преподаватель: Николаев Владимир Вячеславович

Выполнил: Кульбако Артемий Юрьевич

Р3112

# Текст задания.

Доработать программу из [лабораторной работы №7](https://se.ifmo.ru/courses/programming#lab7) следующим образом:

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

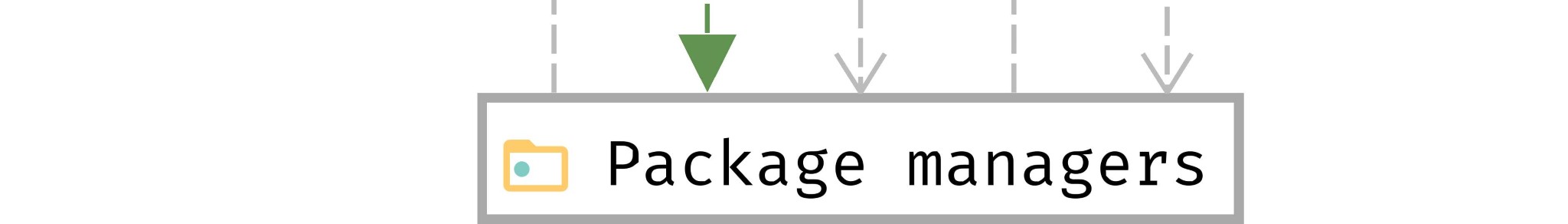
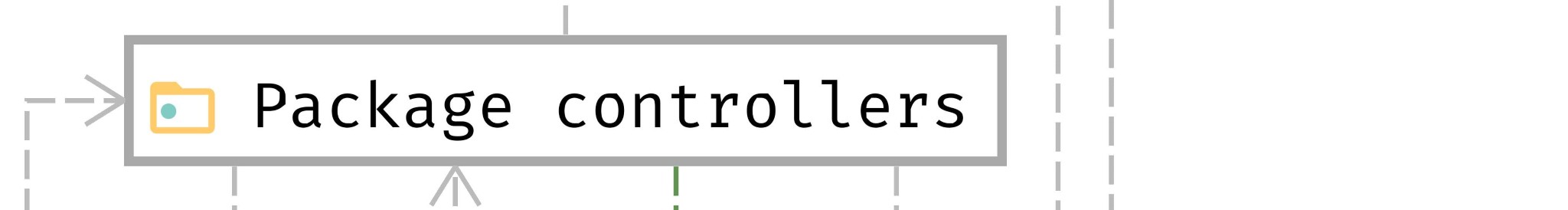
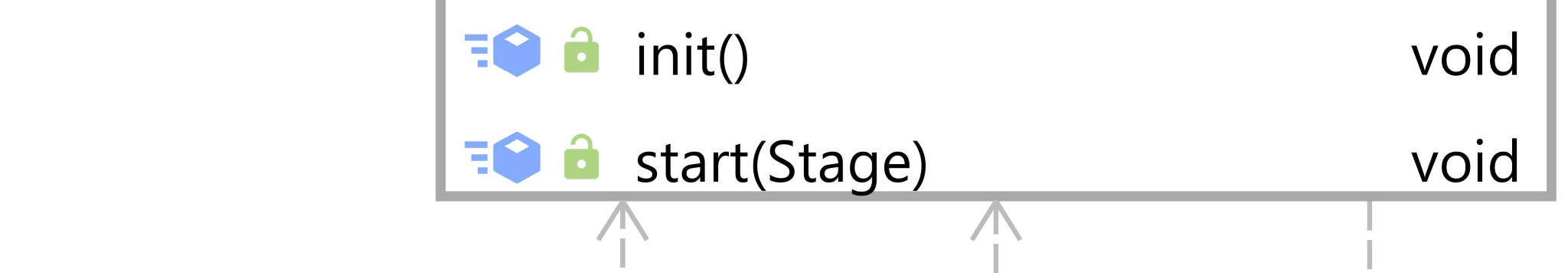
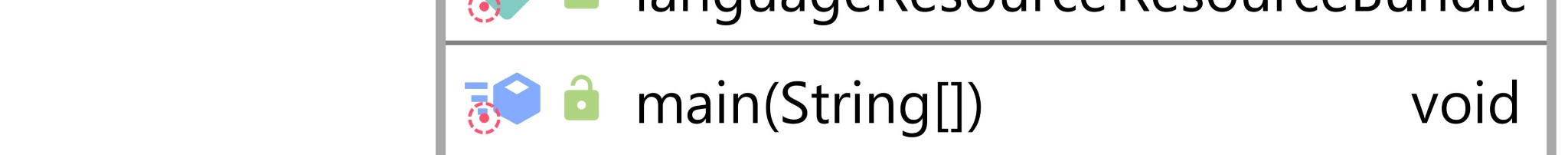
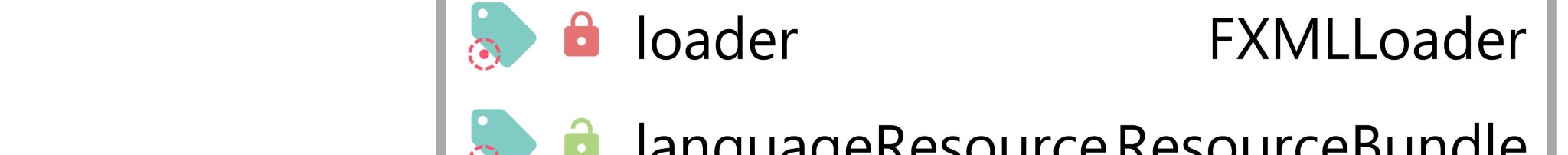
1. Окно с авторизацией/регистрацией.
2. Отображение текущего пользователя.
3. Таблица, отображающая все объекты из коллекции
   1. Каждое поле объекта - отдельная колонка таблицы.
   2. Строки таблицы можно фильтровать/сортировать по значениям любой из колонок. Сортировку и фильтрацию значений столбцов 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 Swing
2. Графический интерфейс клиентской части должен поддерживать **русский**, **нидерландский**, **украинский** и **испанский (Пуэрто-Рико)** 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**.

# Диаграмма классов разработанной программы.

## Клиентский модуль:



# Выводы по работе.

В процессе выполнения лабораторной работы был дополнен пользовательский модуль из лабораторной №7 согласно варианту задания. Основная сложность заключалась в большом объёме компонентов и методов, которые можно использоваться при разработке GUI на Java, и модификации логики программы таким образом, чтобы не пришлось затрагивать серверный модуль, так как до этого его работа была завязана на передаче исключительно строк. Неожиданно, но пункт «Объекты от разных пользователей должны быть нарисованы разными цветами» вызвал самые большие сложности. На основе целочисленного значения уникального номера пользователя необходимо было сформировать цвет фигуры, а это три компонента – red, green, blue, при этом цвета должны были различаться сильно, а не лишь оттенком. Наиболее оптимальным решением оказалось формировать 6-битную хэш-последовательность, которая и задаёт цвет фигуры в виде 16-ричного кода. Сложности были с объектом **javafx.scene.control.TableView<S>**, который предназначен для формирования таблицы из коллекции. Столбы **TableView** заполняются с помощью метода **setCellValueFactory**, который крайне схож с **setCellFactory**, что можно не заметить, и, что со мной и произошло. Проблемы были и с исключениями, относящимися к JavaFX коду. Они неочивидны, из-за того, что на вершине стека исключений находится совсем не то исключение, которое породило ошибку. Отдельного упоминания стоит исключение **Caused by:**[**javafx.fxml.LoadException**](http://docs.oracle.com/javafx/2/api/javafx/fxml/LoadException.html)**: Root value already specified**, связанное с возникновением тега **root** в FXML коде. При беглом взгляде на FXML код этот тег крайне сложно обнаружить (есть целая статья про проблему с этим тегом <https://java-no-makanaikata.blogspot.com/2012/11/javafx-fxml-root-value-already-specified.html>).