**HTML Editor (1)**

Сегодня мы напишем HTML редактор с графическим интерфейсом. В качестве библиотеки для  
создания графического интерфейса воспользуемся Swing. А в качестве архитектурного каркаса  
нашего приложения будем использовать MVC модель.  
1.1. Объяви класс Controller и класс View. Класс View должен быть унаследован от JFrame и реализовывать интерфейс ActionListener.  
1.2. Добавь в класс Controller поля, отвечающие за представление View view и модель HTMLDocument document. Здесь и далее, классы, которые мы не объявляли, но используем в коде скорее всего все находятся в библиотеке swing. Например, класс HTMLDocument реализован в пакете *javax.swing.text.html.*  
1.3. Добавь в класс Controller поле, которое будет отвечать за файл, который сейчас открыт в нашем редакторе (*текущий файл*) File currentFile.  
1.4. Добавь конструктор класса Controller. Он должен принимать в качестве параметра представление и инициализировать соответствующее поле класса.  
1.5. Добавь в Controller пустой метод main.  
1.6. Добавь в класс View поле Controller controller.  
1.7. Добавь в класс View сеттер и геттер для поля controller.  
1.8. Добавь пустую реализацию метода, который объявлен в интерфейсе ActionListener.

Требования:  
1. Класс Controller должен быть создан в отдельном файле.  
2. Класс View должен быть создан в отдельном файле.  
3. Класс View должен быть унаследован от JFrame и реализовывать интерфейс ActionListener.  
4. В классе Controller должны быть приватные поля: View view, HTMLDocument document, File currentFile.  
5. В классе Controller должен быть создан метод public static void main (String[] args).  
6. Класс Controller должен содержать конструктор с одним параметром, инициализирующий поле view.  
7. В классе View должно быть приватное поле Controller controller, а также сеттер и геттер к нему.  
8. В классе View должен быть метод public void actionPerformed(ActionEvent e).

### HTML Editor (2)

2.1. Добавь в контроллер и представление по методу init(), пока без реализаций. Они будут отвечать за инициализацию контроллера и представления.  
2.2. Теперь напишем в классе **Controller** метод main(). Он должен:  
2.2.1. Создавать объект представления.  
2.2.2. Создавать контроллер, используя представление.  
2.2.3. Устанавливать у представления контроллер.  
2.2.4. Инициализировать представление.  
2.2.5. Инициализировать контроллер. Контроллер должен инициализироваться после представления.  
2.3. Добавь в контроллер метод exit(), он должен вызывать статический метод exit у класса  
**System**.  
2.3.1. Метод exit в классе **Controller** не должен быть статическим.  
2.4. Добавь в представление метод exit(), он должен вызывать exit() у контроллера.

Требования:  
1. Класс Controller должен содержать метод public void init().  
2. Класс View должен содержать метод public void init().  
3. Класс Controller должен реализовывать метод public void exit().  
4. Класс View должен реализовывать метод public void exit().  
5. Реализуй, согласно условию, метод public static void main(String[] args) у класса Controller.