Лабораторная работа 4

Создание графического интерфейса: swing.

**Задача**. Для задачи из лабораторной работы 3 создать графический интерфейс.

Добавьте к проекту из лабораторной работы 3 еще один java-класс с именем Program. В нем будет метод main, запускающий «форму». Для формы создайте класс, например, с именем GUI, и унаследуйте его от JFrame. Экземпляр формы нужно создать в методе main.

Перейдем к созданию графического интерфейса (страшная форма на рис. 1).

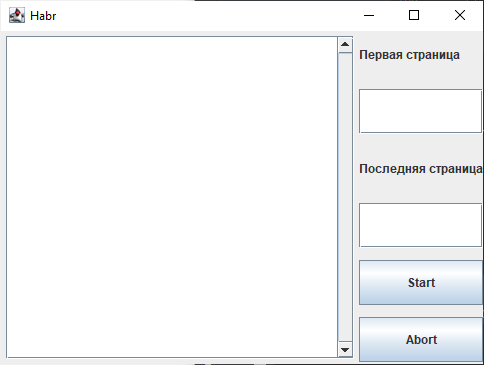


Рисунок 1. Окно программы

В конструктор класса GUI добавьте следующие строки:

JFrame.*setDefaultLookAndFeelDecorated*(**true**);  
setTitle(**"Habr"**);  
setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

Для создания правой части создадим панель и установим ей табличную разметку на один столбец:

JPanel rightPanel = **new** JPanel();  
GridLayout layout = **new** GridLayout(0, 1, 5, 12);  
rightPanel.setLayout(layout);

Далее нужно создать элементы интерфейса: надписи, поля ввода, кнопки. Для надписей можно не создавать переменных:

rightPanel.add(**new** JLabel(**"Первая страница"**));

Полям ввода и кнопкам нужны имена:

JTextField textStart = **new** JTextField(10);  
JTextField textEnd = **new** JTextField(10);  
JButton startButton = **new** JButton(**"Start"**);  
JButton abortButton = **new** JButton(**"Abort"**);

Добавьте эти элементы на правую панель. Саму панель нужно поместить на справа на форму:

add(rightPanel, BorderLayout.***EAST***);

При нажатии на кнопку считаются значения полей и напустится парсинг. Для этого нужно описать для парсера переменную (можно объявить его полем класса) и инстанцировать в конструкторе.

Кнопке startButton добавьте слушателя, обрабатывающего нажатие мыши:

startButton.addДЕЙСТВИЕListener(**new** НУЖНЫЙListener() {  
 @Override  
 **public void** МЕТОД(НУЖНЫЙEvent e) {  
 **int** start = // получить число из textStart  
 **int** end = // получить число из textEnd  
 // задать значения для ParserSettings

// запуск парсинга

}  
});

Слушателя для кнопки abortButton добавьте самостоятельно.

Отображать результаты поиска будем в текстовом поле:

JTextArea **textArea**;

Его нужно добавить в центр формы. Для вывода туда информации будем использовать возможности класса ParserWorker. Добавим классу GUI возможность реализовать интерфейсы ParserWorker.OnCompleted и ParserWorker.OnNewDataHandler<ArrayList<String>. В методе OnNewData реализовать представление полученных из сети данных.

@Override  
**public void** OnNewData(Object sender, ArrayList<String> args) {  
 // добавить строки из списка args в textArea  
}

В методе OnCompleted вывести сообщение о конце загрузки:

JoptionPane.*showMessageDialog*(**this**, **"Загрузка закончена"**);

Чтобы методы сработали нужно добавить класс GUI в списки парсера:

**parser**.**onCompletedList**.add(**this**);  
**parser**.**onNewDataList**.add(**this**);

Запустите программу. Проверьте правильность работы.

**Задание**.

1. Измените интерфейс для задания 1 лабораторной работы 3 (заголовок новости, текст новости и картинка).