



The Future of Navigation

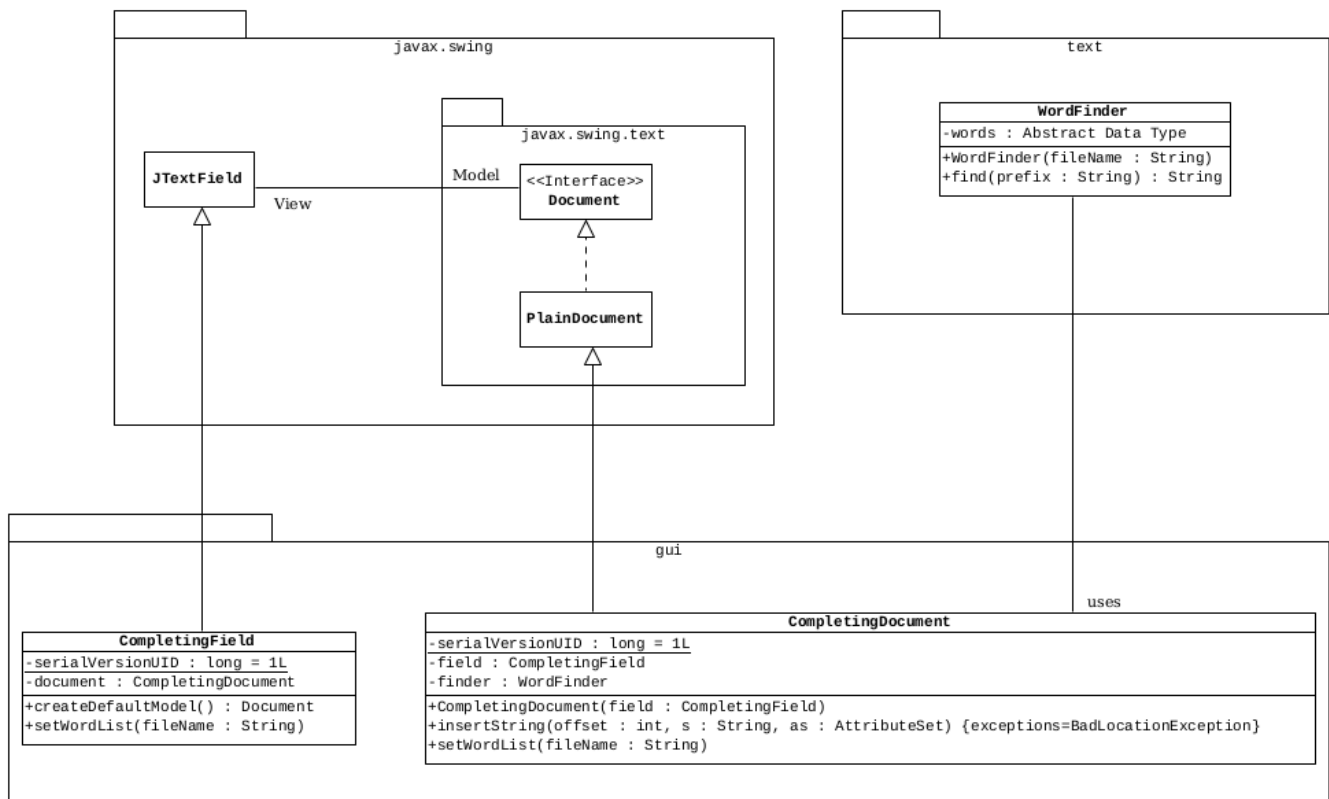
Address Entry System v1

Purpose

The address entry system is a collection of components that makes it easier for users to enter addresses. Version 1 of the system consists of components that perform typing completion.

Design

The design of the system is summarized in the following UML class diagram:



Note that the details of the components that are part of Java's Swing API have been omitted from this diagram.





The Future of Navigation

Specifications

This section contains design specifications for the components above.

The WordFinder Class

A `WordFinder` object is used to find a word or words that begin with a particular set of letters (called the *prefix*). Note that the term "word" is used loosely, since a "word" may contain spaces and/or punctuation characters.

The constructor must read the list of valid words from a file. It must assume this file is in the current working directory (i.e., the name of the file must not include a path).

The `find()` method must return the first word (alphabetically, in the list of valid words) that begins with the given prefix (or `null` if there is no such word).

The CompletingDocument Class

A `CompletingDocument` is the model component (in the sense of the Model-View-Controller Pattern).

The `setWordList()` method must construct a `WordFinder` using the given file name.

The `insertString()` method must determine what the user has typed, use the `WordFinder` to find a word that begins with what the user has typed, put that word into the model, and select the text that was not typed by the user in the view. For example, if the user has typed `Ma` and the valid word list contains `Madison`, the model must contain `Madison` and the view must display `Madison` (so that if the user continues typing the text that was inserted by the system, the `dison`, is overwritten).

The CompletingField Class

The `CompletingFieldClass` is the view (and controller) component (in the sense of the Model-View-Controller Pattern).

The `createDefaultModel()` method must create and return a `CompletingDocument` object (i.e., the model).

The `setWordList()` method must invoke the `setWordList()` method in the model, if the model has already been created.