Use Java Native Access to Determine Default Directory

Context and Problem Statement

JabRef needs to propose a file directory to a user for storing files. How to determine the "best" directory native for the OS the user runs.

Decision Drivers

- Low maintenance effort
- Follow JabRef's architectural guidelines
- No additional dependencies

Considered Options

- Use Swing's FileChooser to Determine Default Directory
- Use `user.home`
- AppDirs
- Java Native Access

Decision Outcome

Chosen option: "Java Native Access", because comes out best (see below).

Pros and Cons of the Options

Use Swing's FileChooser to Determine Default Directory

Swing's FileChooser implemented a very decent directory determination algorithm. It thereby uses `sun.awt.shell.ShellFolder`.

- Good, because provides best results on most platforms.
- Good, because also supports localization of the folder name. E.g., `~/Dokumente` in Germany.
- Bad, because introduces a dependency on Swing and thereby contradicts the second decision driver.
- Bad, because GraalVM's support Swing is experimental.

- Bad, because handles localization only on Windows.

Use `user.home`

There is `System.getProperty("user.home");`.

- Bad, because "The concept of a HOME directory seems to be a bit vague when it comes to Windows". See https://stackoverflow.com/a/586917/873282 for details.
- Bad, because it does not include `Documents`: As of 2022, `System.getProperty("user.home")` returns `c:\Users\USERNAME` on Windows 10, whereas `FileSystemView` returns `C:\Users\USERNAME\Documents`, which is the "better" directory.

AppDirs

> AppDirs is a small java library which provides a path to the platform dependent special folder/directory.

- Good, because already used in JabRef.
- Bad, because does not use `Documents` on Windows, but rather `C:\Users\<Account>\AppData\<AppAuthor>\<AppName>` as basis.

Java Native Access

- Good, because no additional dependency required, as it is already loaded by AppDirs.
- Good, because it is well maintained and widely used.
- Good, because it provides direct access to `Documents` and other system variables.