Problem Statement
Often in classroom presentations, the default font sizes used by Java in its panels and widgets are too small for students in the back of the classroom to see easily. It would be nice to have a simple way to increase these defaults uniformly.

More generally, a faculty member or student may want to test how a particular GUI looks with different look-and-feel settings. It would also be nice to do this easily.

Solution Overview
The class `LookAndFeelTools` has two static methods that solve the above problems. We consider the second problem first. If the first line of the `main` program is the following:

```java
LookAndFeelTools.showSelectLookAndFeelDialog();
```

then a dialog of the following form is presented:

![Select Look and Feel Dialog](image1)

The first line lists the look and feel objects available on the system and allows the user to select one via a radio button. The system default look and feel is the one initially checked. The second line allows the user to choose a uniform increment or decrement to the dozens of font sizes that characterize the selected look-and-feel. Since a look-and-feel may vary font sizes depending on the widget, this adjustment will be done on a font-by-font basis to preserve relative size relationships. The default adjustment is `None` but the snapshot shows a selected adjustment of `+4`.

Here are some snapshots of the same GUI shown in the three look-and-feel settings we had available on our system. Each uses an increment of `+4` to emphasize the change in font size.

![Circle Sample](image2)
We now turn to the simpler problem of adjusting the default font size without changing the default look-and-feel. For this, a dialog is unnecessary. Instead, one simply issues a command of the form:

```java
LookAndFeelTools.adjustAllDefaultFontSizes(2);
```
as the first line of the `main` program. In our classrooms, an increment of +2 seems to make students happy. The exact value of the increment will of course vary from school to school.

For convenience, in the vanilla `Methods.java` file used to start the Java Power Framework, we place both of the above commands in the `main` program with comments. The user, faculty member or student, can decide which one, if any, to uncomment and use.

**Experience with the Solution**
The solution works like a charm and seems to be the kind of tool that would be pedagogically useful for both faculty and students.

**API Documentation & Related Materials**
The main JPT site to access documentation, code, and the jpt.jar:
http://www.ccs.neu.edu/jpt/