Software Development Project

CS 478

Fall, 2014

You are to create an Integrated Development Environment (IDE) for and Introductory Programming Language. The language is a graphical language. The development language, operating system, and target language (see Testing) are the developers choice, within reason!

**The Environment**

The environment will be similar to modern IDEs – jGrasp, Visual Studio, etc. It should have a main window for program development, with “appropriate” menu items across the top. Along one side (the right?) will be a list of currently known objects (modules, programs). This list should be kept as a directory structure. These windows may be resized by the user.

There will need to be “pop-up” windows for actions like saving a program, executing a program, etc. The language definition will also necessitate new “pop-up” windows.

**User Interaction**

The user will not “write” a program in the usual sense. Programming will be accomplished by a set of templates. When a user wishes to use one of the templates, they will indicate this by choosing the template with a “hot key.” For example, an assignment statement may be chosen by “Alt-A”.

Each template will have text to display as well as “spots” for user supplied information. I.e., the statement will be syntactically correct with holes for additional information. In the case of an assignment, the user must supply the expression to calculate and the variable to receive the new value. The assignment might be displayed to the user as:

Calculate ∆ and save the result as ∆

The user would “click” on the appropriate ∆, which would open a window to permit the user to supply the needed information. The first ∆ would require an “expression” and the second ∆ would expect a variable name. Users should be able to minimize/maximize these pop-up windows to make additions/changes. Furthermore a window may be recursive: an IF statement can contain and IF statement, which can contain an IF statement, etc.

Note: ∆ was chosen as a quick display. A better choice might be a “box”.

What templates are available and what elements are part of the templates are defined by the language in use.

**The Languages**

A language will be defined for the user (possible chosen by the user). The definition will be in a file (in a known directory). The templates will consist of an appropriate hot key, a name, and the display template. For example:

A, assignment, Calculate ∆ and save the result as ∆

Other elements (like expressions, conditions, etc.) will probably be stored as BNF notation.

**Testing**

Users can test their program by choosing the appropriate menu item. The user program will then be converted to an actual programming language (developer’s choice), compiled, and executed. The input/output for the user program will be managed by the IDE.

**File Input/Output**

Your IDE will be expected to read in a language definition, and then interacted with the user to create, test, load, save, etc. their programs.

**The language Files**

Each language file will describe a different language. For now you can think of these different languages as starting with a very simple language (input and output only?) and then additional languages becoming more and more complex.

Each file should consist of the language elements – templates, additional definitions – and rules for conversion to an actual programming language for execution.

**The User Files**

The user may save/load their creations anywhere available but the current directory is displayed for the user to simply click their choice.