Cale Gallaway

iEx6

Programming Environment Survey

1. A good programming environment should highlight the sections of code that don't compile so that you can more easily recognize it and fix it. If possible, the programming environment should provide possible solutions to the problems it finds in your code. A good programming environment should also make it easy to navigate through your code; a couple examples of this are allowing the user to click on variable names and going to where those variables are defined and then used and also searching through multiple files at once. A good programming environment should also recognize when you are accessing a class and list the methods inside that class so that you don't have to remember all those names. Eclipse does all of these things which is why it has been one of my favorite programming environments to date.
2. I like the way the compiled program highlights things when all those blue lines show up. I like the way the compiler will tell me what contract is broken when an error occurs even though it doesn't always says where and it can be hard to decipher what the contract means. I also like that I can define properties and it will test them.
3. I disliked the fact that when it finds error in my code that it doesn't give any kind of clue on how to fix it and that sometimes the error messages were sometimes difficult to decipher. I also disliked that I could never get the debug mode to run properly; every time I tried to enter debug mode it would crash. I also disliked how it would open one file per window; it was sometimes hard to keep track of the three or four windows I would have open at one time.

4. I would chose Eclipse:

There are so many development environments out there that it can be hard to choose the one that is right for you. The choice can rely in part on what programming language has been selected to develop in. In my opinion Eclipse is the best choice for our programming environment. It supports a number of languages, provides excellent file storage and navigation, and helps the developer in writing code.

Eclipse supports a variety of programming languages. Eclipse supports object oriented languages such as Java and C++ as well as functional languages like lisp. This allows the developer to work on multiple projects within the same development environment. This can be very useful for developing software applications that are required to interface with multiple languages. Eclipse even supports database languages such as SQL.

Eclipse also provides excellent file storage and navigation. It supports code repositories. These are very useful for when there are multiple developers working on the same project. Developers can easily set their project to update from repositories on a schedule or do it manually. Eclipse makes it very it to navigate through multiple projects. You can open a class by simply selecting the open declaration option on any reference to that class. This is very useful when looking at code that is foreign to you. In Eclipse a developer can easily run searches by strings or regular expressions in order to find code in large projects.

Eclipse can also aide developer in generating code. Eclipse has the options to generate blocks of code such as getters and setters or surrounding code in a try/catch block. This can be very helpful and time saving, allowing the developer to work on the more intricate parts of the code.