Towards a Web Learning Environment for Supporting Object Oriented Programming Courses

Carlos R. Jaimez-González, Wulfrano A. Luna-Ramírez

Departamento de Tecnologías de la Información, Universidad Autónoma Metropolitana -
Cuajimalpa, Av. Constituyentes No. 1054, Col. Lomas Altas, C.P. 11950, México D.F.
{cjaimez,wluna}@correo.cua.uam.mx

Abstract. Computer programs written in the Java programming language require the Java development kit in order to be compiled and run; this can be accomplished through a command line or using an integrated development environment on the computer where the program is being written. This paper presents a web learning environment, which is composed of an online code editor, initially for the Java programming language, but it can be extended to other object-oriented programming languages, such as C++. This web environment is designed with features that support students during the process of learning this programming paradigm, such as allowing them running the code edited, showing the output of the program, showing the list of exceptions occurred during compilation and pointing out the place of the exception. All these features are accessible through the web environment, which can be used by students through a web browser on any computer with Internet access.

Keywords. Java programming language, object-oriented programming, web learning environment, online compilation, interactive web environment.

1 Introduction

The Information Technologies and Systems undergraduate program at our faculty has an Object-Oriented Programming course, additionally to other courses that require the use of object-oriented programming languages. It could be of great support for students to have a web learning environment that helps them in the process of learning object-oriented programming, such as its main principles and concepts (classes, objects, methods, attributes, inheritance, polymorphism, etc.), and make transparent the compilation and execution of programs.

A web learning environment for supporting object-oriented programming courses is being developed, which uses the Java programming language, but it can be extended to other object-oriented languages, such as C++. This web environment will support the process of teaching and learning for object-oriented programming courses. The reason for using Java is simply because most of the lecturers use Java as the programming language for their courses. There are some projects available on the web, which are tools for supporting programming from a web platform, but some of them are very simple [8], expensive [6][7], or they do not provide the possibility of work-

Advances in Computer Science and Engineering
Research in Computing Science 67, 2013 pp. 33–41
Paper Recived 31-08-2013 and Acepted 21-10-2013

ing with Java [5]. Some others provide the possibility of working with several files at the same time [7][8], or compiling online and running offline [4].

Computer programs written in the Java programming language require the Java Development Kit (JDK) in order to be compiled and run; this can be accomplished through a command line or using an Integrated Development Environment (IDE) on the computer where the program is being written. Figure 1 shows the process of edition, compilation, and execution of Java programs. The complete process starts with the creation of .java files with the source code, the use of the editor for writing the program (source files), the delivery of the files for compilation, the actual compilation, the generation of the .class files (in case there are no syntax errors), the execution of the program through the Java Virtual Machine (JVM), and the visualization of the program output.

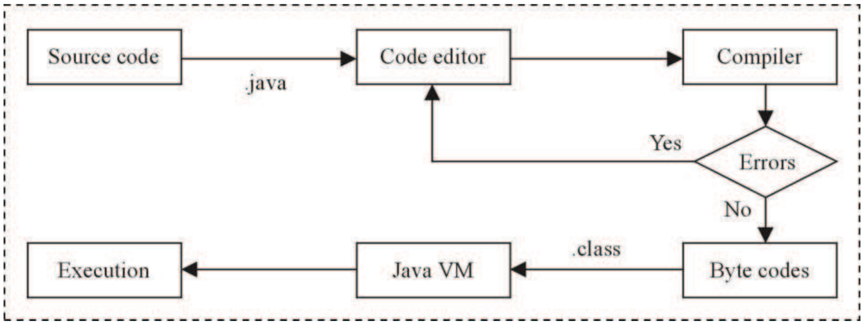


Fig. 1. Process of edition, compilation and execution of Java programs.

The web environment is composed of an online code editor for object-oriented programming languages, such as Java or C++. The environment has features to help students during the learning process of this programming paradigm; such as allowing them executing the edited code; showing them the output of the program; showing them the list of errors during compilation, and where they are inside the code. Students do not need to leave the web learning environment to use all these features, because they are built in the environment. A remote compiler will also be provided by the environment, so students do not need to install any. This way, they can learn and practice object-oriented programming while studying a tutorial, without the need of navigating between different applications (source files, web browser, books, etc.).

This web learning environment is thought to be stand-alone, which allows the student accessing it and working with it through a web browser, from any computer with Internet access. Additionally, this learning environment is planned to be incorporated to the online interactive tutorials platform, which is being developed [16].

The rest of the paper is organized as follows. Section 2 presents a brief summary of some existing tools that are relevant to this project; mainly tools for edition, compilation and execution of code. Section 3 describes the architecture of the web learning environment; the software and supporting tools used; the edition, compilation, and execution of programs; and the help section for the segments of code with errors. Finally, section 5 provides conclusions and future work.

2 Related tools and applications

This section reviews some existing tools that are related to the web learning environment being developed. Among the tools found, there is a tool that covers some of the features desired in the web learning environment; however, it is expensive, and it is thought as an online IDE and not as a tool for learning a programming paradigm. There are also some tools that are actually plug-ins, which can be used to implement some of the features planned for the web learning environment. Table 1 shows some of the existing tools, along with a brief description of their purpose.

Table 1. Tools for edition, compilation, and execution.

Name	*Description*
Compilr	Online IDE.
Codepad	Compiler/interpreter.
DJGPP Public Access	Cross-compiler service for creating small programs in DOS.
Ideone	Online compiler and debugger.
JXXX Compiler Service	Remote compilation service for Java files.
OnlineCompiler	Online compiler.
CodeMirror	JavaScript component that provides a code editor for Web.
CodePress	Online code editor, written in JavaScript.
EditArea	Code editor, written in JavaScript.

Compilr [7] is a web application that allows creating projects in different object-oriented programming languages. It provides a file browser interface, it is multi-language and allows the edition of more than one file at the same time; it has a code editor and syntax highlighting; it provides a remote compiler, and executes the code with input parameters taken from the user; it visualizes the output, and allows the execution of Java *Applets* [15]; it is similar to *Eclipse* [13] and *Netbeans* [14].

Codepad [5] is a compiler/interpreter of code that can be used by 10 different programming languages. It has no support for the Java programming language, but for others, such as Python, Ruby and C++. *It* provides a multi-language code editor, remote compilation, and it allows visualizing the output of the program after execution; its functionality is more oriented to code debugging, because it highlights the code lines where it finds an error.

DJGPP Public Access Cross-Compiler [8] is a service designed for users that need to create small programs in DOS and do not have access to a computer with a compiler. It only has support for programs in the C programming language. It is a very basic application, because it only has support for one language, and its interface is limited with only a small text area with two options: *Warnings* and *Optimize*.

Ideone [10] is an online compiler and debugger, which allows compiling and executing code in 40 programming languages. It has a code editor, syntax highlighting, multi-language, remote compiler, and it allows input from the user for the execution; it also visualizes the output of the program. It can only executes programs with one source file; it does not allow the execution of Java *Applets*.

JXXX Compiler Service [11] is a service for remote compilation of source files written in the Java programming language. This tool allows the execution of *Applets* without having installed the JDK. This tool is basically a form with remote compilation, loading of libraries, selection of different JDK, and the possibility of compiling several files at the same time. This service also includes the option to load an HTML file for visualizing *Applets*. It does not have a text editor to write code.

OnlineCompiler [12] is a web site to paste and compile code. It does not show any output from the programs, because its main functionality is to detect syntax errors of the 5 different languages it supports (C/C++, Fortran, Java, Pascal and Basic). It has a code editor, it is multi-language, and it can compile source files remotely. The web interface of this tool allows writing or pasting code on a text area, in order to compile it. The result of the compilation is a web page with a link to the source file, and another link for the executable or compiled file.

CodePress [6] is a source code editor, written in *JavaScript*, which colors the text as it is being written on the web browser. Some of its main advantages are the following: syntax highlighting, auto-complete, possibility of having multiple windows of code inside the same web page, and the support for several programming languages.

EditArea [9] is a free code editor, written in *JavaScript*, which is designed for editing source code inside of a <*textarea*> element; it has several functionalities that are useful, such as syntax highlighting, tabs support, well-formatted text, search and find functions, auto-indent, and the possibility of having multiple instances.

CodeMirror [4] is a *JavaScript* component that provides a code editor with syntax highlighting, indentation, search and replace. It is a code editor that can be embedded in a web page, and can be adjusted to the desired window size, which affects the way it visualizes the code.

3 The web learning environment

This section presents the architecture of the web learning environment, the software and supporting tools used for its development (edition, compilation, and execution of programs), and the help sections that are provided for the code with errors.

3.1 System architecture

The web learning environment is based on a client-server architecture. The server is in charge of the compilation and execution of programs; it also registers the output of the program and any errors or exceptions occurred. The client side shows the web interface, which is composed of a source code editor, where the user can edit the code and send it to the server side to be compiled and executed, as it is shown in Figure 2.

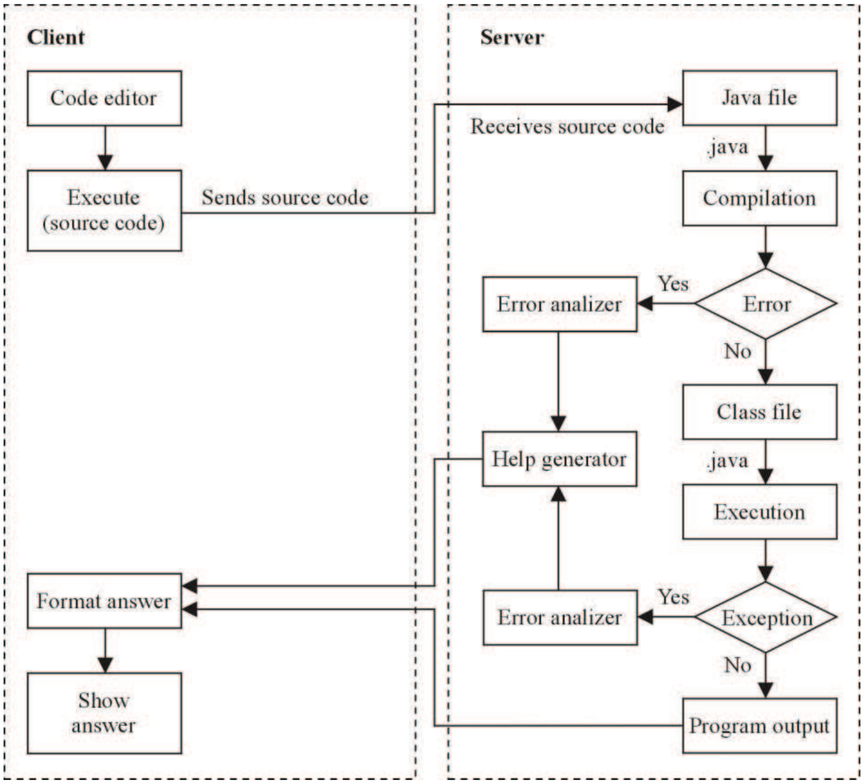


Fig. 2. Client and server sides: edition, compilation and execution of Java programs.

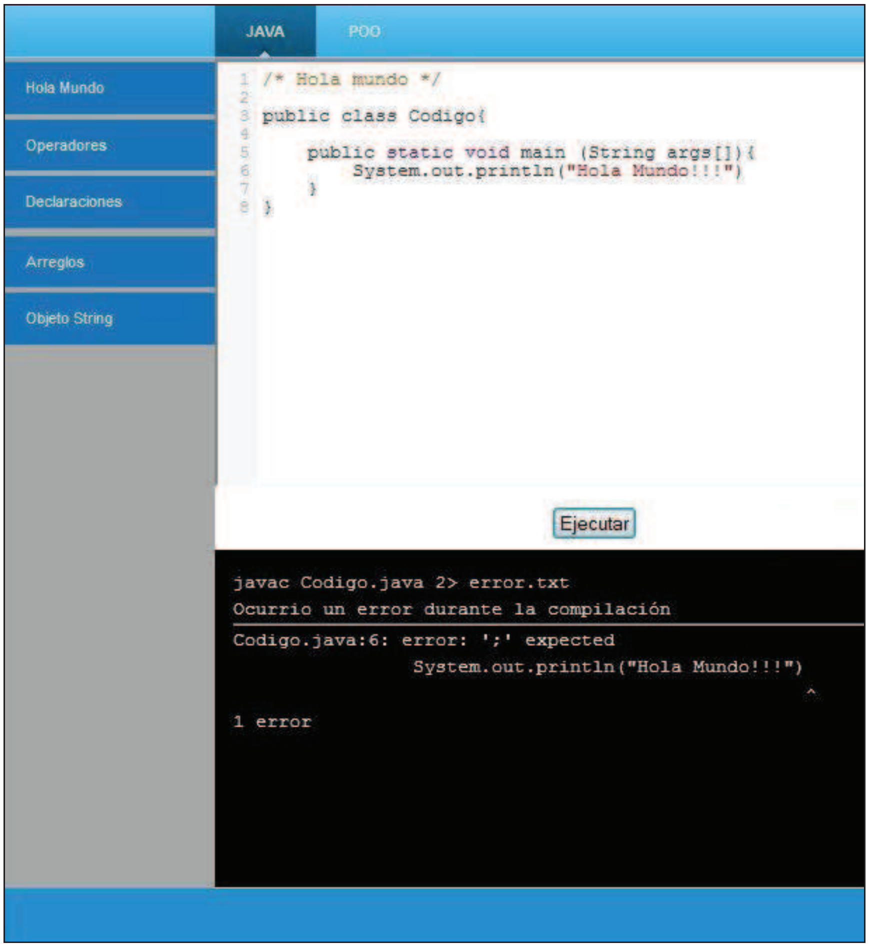
3.2 Software and supporting tools

The web environment uses Apache Tomcat as a web server and container of Java Server Pages; it is also used for the dynamic generation of content, and database connectivity. Java is the programming language for the business classes; it also has the *JavaCompiler* class, which is used for the compilation of programs on the server side. For the web user interface it is used HTML, CSS, and JavaScript: HTML for the structure of the content, CSS for the presentation of the content, and JavaScript for the interaction with the user and for the asynchronous method calls to the server.

3.3 Edition

After having reviewed several code editors with syntax highlighting [4][6][9], *CodeMirror* was chosen, because it is the most complete editor from the ones evaluated. Although the web learning environment is only using syntax highlighting and auto-indentation, in the future it can use more features of *CodeMirror*, such as auto-complete, pre-visualization of HTML, auto-complete of XML tags, etc.

Figure 3 shows the initial version of the web interface, where it can be observed three panels. The left panel shows some examples of the syllabus established for the environment; the top panel shows the code editor, and the bottom panel shows the console for compilation and execution of programs. The top panel has an example of a Java class, which is also shown in Figure 4.

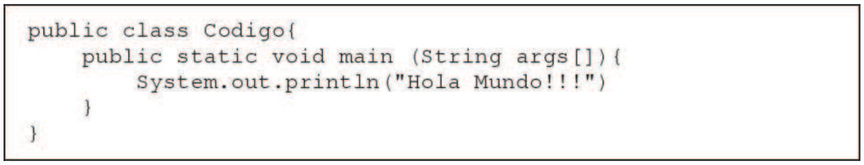


```
/* Hola mundo */

public class Codigo{

    public static void main (String args[]){
        System.out.println("Hola Mundo!!!")
    }
}
```

```
javac Codigo.java 2> error.txt
Ocurrio un error durante la compilación
Codigo.java:6: error: ';' expected
                System.out.println("Hola Mundo!!!")
                                                   ^

1 error
```

Fig. 3. Interface of the web learning environment, with examples and code editor.

Figure 4 shows an example with a syntax error. Due to this error, it can be observed in Figure 3 the error message at the bottom panel.



```
public class Codigo{
    public static void main (String args[]){
        System.out.println("Hola Mundo!!!")
    }
}
```

Fig. 4. Java class example with a compilation error.

3.4 Compilation and execution

There is a Java interface called *JavaCompiler*, which allows invoking programs; it provides methods for compiling the source code generated in the code editor, and also for capturing the errors generated during compilation and execution. Inside the web learning environment, this interface is implemented on the server side to provide compilation and execution services.

3.5 Help

This is one of the most difficult tasks. The web environment will offer information about the errors occurred during compilation, or exceptions occurred during execution. The environment will have a menu, from which there will be well-commented examples with brief explanations, which will serve as support for students for familiarizing and understanding the object-oriented programming paradigm. This is accomplished through the Java programming language, using simple examples of the topics covered for the web environment, such as those found in [1], [2], or [3].

The editor area will provide the help material. The main features of this editor are the following: 1) edition of code with syntax highlighting; 2) online compilation and execution of source code with only one click; 3) visualization in the web interface of compilation or execution errors, with an option to see an example related with the error; 4) mapping of errors inside the source code (highlighting the line where the error occurred), 5) viewer of applets.

4 Conclusions and future work

This paper presented a web learning environment, which is composed of an online code editor, initially for the Java programming language, but it can be extended to other object-oriented programming languages, such as C++. This web environment is designed with features that support students during the process of learning this programming paradigm, such as allowing them running the code edited, showing the output of the program, showing the list of exceptions occurred during compilation and pointing out the place of the exception. All these features are accessible through the web environment, which can be used by students through a web browser on any computer with Internet access. The web learning environment is under development, although there is an initial working version. Additionally, it is planned that this environment could be incorporated to an online tutorials platform.

Further work is needed to finish the web environment with the complete functionality to compile and execute programs; as well as the functionality to implement the help for students, which will show the exact place of the error, along with examples to reinforce the knowledge acquired; and the possibility to include a reference to the Java API corresponding to the topics covered. It is needed to carry out tests and evaluations to determine the effectivity of the web learning environment in the teaching-learning process in the object-oriented programming courses.

References

1. Eckel, B. Thinking in Java (4th Edition), Prentice Hall (2006).
2. Flanagan, D. Java in a Nutshell (5th Edition), O'Reilly Media (2005).
3. Arnold, K., Gosling, J., Holmes, D. The Java programming language (4th Edition), Addison Wesley Professional.
4. CodeMirror. Available at: http://codemirror.net/. Last access in July 2013.
5. Codepad. Available at: http://codepad.org/. Last access in July 2013.
6. CodePress. Available at: http://codepress.sourceforge.net/. Last access in July 2013.
7. Compilr. Available at: http://www.compilr.com/. Last access in July 2013.
8. DJGPP Public Access Cross-Compiler. Available at: http://www.delorie.com/djgpp/compile/. Last access in July 2013.
9. EditArea. Available at: http://www.cdolivet.com/editarea/. Last access in July 2013.
10. IdeOne. Available at: http://ideone.com/. Last access in July 2013.
11. Java Compiler. Available at: http://www.innovation.ch/java/java_compile.html. Last access in July 2013.
12. Online compiler. Available at: http://www.onlinecompiler.net/. Last access in July 2013.
13. Eclipse. Available at: http://www.eclipse.org/org/. Last access in July 2013.
14. NetBeans. Available at: http://netbeans.org/. Last access in July 2013.
15. Oracle Corporation Applets. Available at: http://java.sun.com/applets/. Last access in July 2013.
16. Jaimez-González, C., Sánchez-Sánchez, C., Zepeda-Hernández, S. Creating and Administering Interactive Online Tutorials and Performance Evaluation Tests Through a Novel Web Platform. International Journal for Cross-Disciplinary Subjects in Education (IJCDSE), Volume 2, Issue 3, ISSN: 2042 6364 (Online), pp. 447-455, 2011.