Text Editors vs IDEs for Beginner Programmers

The University of Texas at Dallas

800 W Campbell Rd, Richardson, TX 75080

nia210000@utdallas.edu

I. INTRODUCTION

A. Industry Background

1). The Ever-Evolving Popularity of Programming: In the current day, technology has evolved beyond imagination. We are at the peak of engineering; creating self-driving cars, automated waitresses at restaurants, and home devices that tell us about the weather with a simple acknowledgment. This amazing technology came to fruition due to the hard work of programmers and engineers. For an inanimate object to interact with humans, someone had to program it. Today, just about anyone can learn how to program thanks to the many resources available online and elsewhere. This was not always the case. In the past, there was no wireless internet, commonly referred to as Wi-Fi. Early software developers learned to code via books in the library and by trial and error. This mindset has allowed programmers to learn efficiently and effectively. Victor Anjos, Chief Executive Office of Kunin Inc., stated that "we had to do the work, and we had to learn – or we'd fail." [1]. Things did not come easy for those who set the foundation for what is software engineering today. In the past, to receive recognition in the field, you needed to have the education to support your program. Today, the focus is more on the ability of a programmer rather than their education. Fellow software developer Elye, states "Even if you don't have a thesis or degree, if you've built an app used by many people or have a popular blog on development, you're regarded as an expert in the field." [2]. If it is the quality of code that a programmer writes that is most important, then it is essential that all programmers learn to write quality code.

2). The Popularity of the Java Language: There are many programming languages for a beginner programmer to choose from to begin learning how to code. One of the more famous of those languages is Java. Java was developed by James Gosling in 1995 with hopes “to create a new language that would allow consumer electronic devices to communicate with each other.” [3]. The capabilities of Java were “revolutionary.” Its ease of use and its versatility has allowed it to remain one of the most popular programming languages of all time. The number of Java developers varies. A recent study has revealed that “the number of software developers who use Java reached 8 million software developers worldwide.” [4]. Java is used worldwide to build applications and introduce students to the world of programming. Its platform independence has gained notoriety as well. Being able to run the same program across different computers is important in the software development industry. According to TIOBE, Java currently ranks third overall as the most popular programming language. [5]. Its popularity does not appear to be diminishing anytime soon, thus it is important for programmers to learn how to code in Java, as it may be required in their occupation.

3). The History of Text Editors and IDEs: For a programmer to program something, there must be a place to write code in. Text Editors were first introduced to debug punch card programs of the early computer. The first official Text Editor & Corrector, as it was called, was released in 1962. [6]. The revolutionary technology allowed developers to write, type, and create their code. Text editors were used to compose the code, then it would be compiled separately. In the 1980s, developers would have to write down the errors by hand, then go back to the text editor and reconfigure their code. [7]. This process remained steady for many years until the first

IDE was created. An IDE stands for Integrated Development Environment, and it was created with the insight of making coding easier for the programmer. The first notable IDE to gain recognition was Visual Basic in 1991. This IDE allowed the programmer to drag and drop items, thus leading to a faster, and easier coding process. [7]. Today, text editors and IDEs have changed drastically. There are many more features in both, giving the developers many choices to use when writing code.

B. Purpose

With the evolution of text editors and IDEs, there has been an ongoing debate on which is superior to the other, especially regarding beginner programmers. The goal of this report is to provide insight into the two and allow the reader to distinguish the difference between an IDE and a text editor. The audience for this report would be Professors of introductory programming courses and individuals interested in learning how to code and what to use. I will examine whether text editors or IDEs are a better option for Java development and analyze each option regarding the user interface, features, accessibility, and performance.

C. Problem Statement

A programmer cannot call themselves a programmer if they do not know how to code. There is a debate on whether the use of an IDE hinders a programmer's ability to fully grasp and comprehend what they are typing. Since an IDE was created with the mindset of helping the programmer, some believe that beginners should learn to code in a text editor, where they must write their code without the use of syntax aids and autofill. Hence, it is important to bring to light the benefits and downfalls of the two and analyze what is better for what situation.

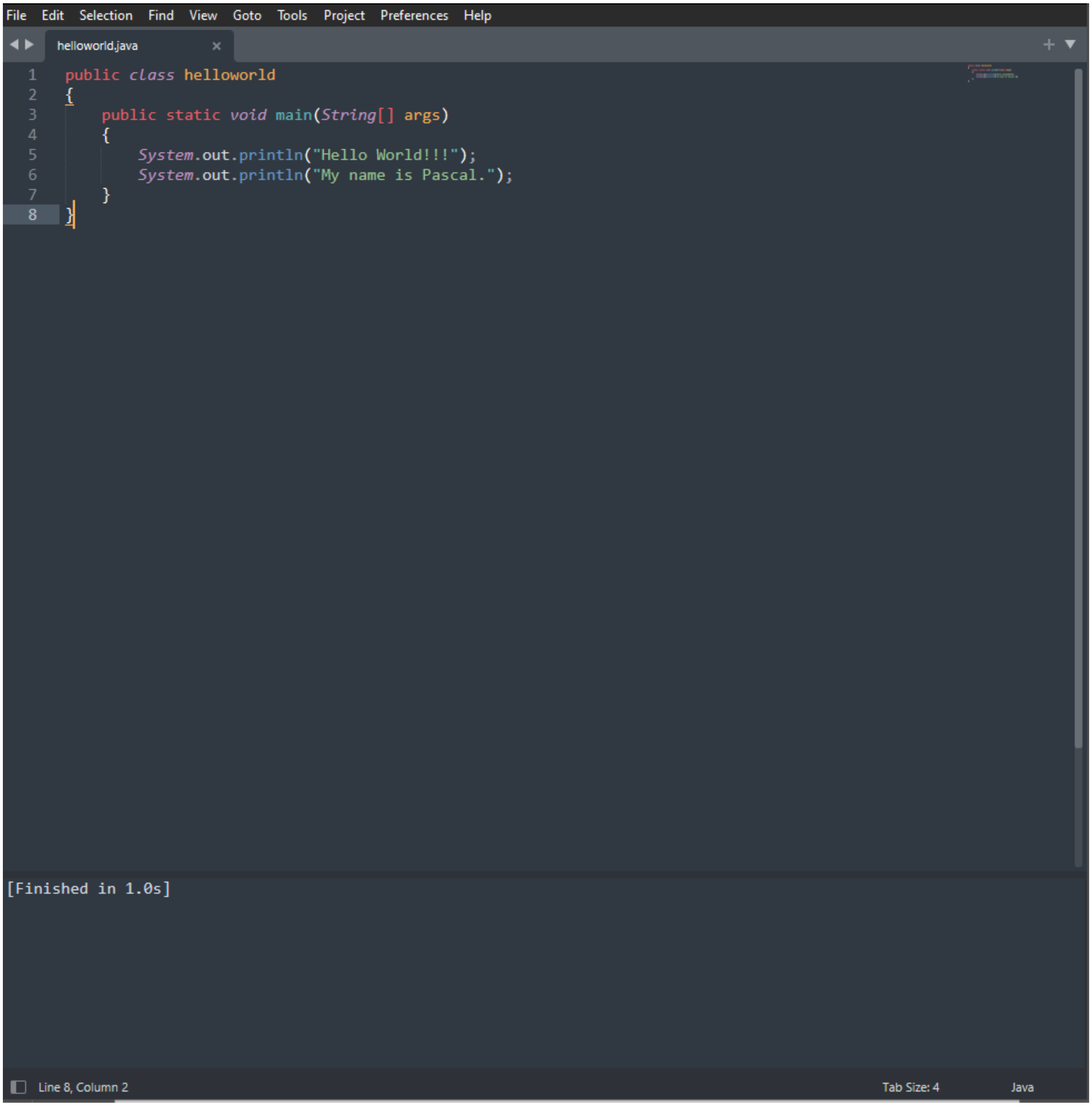
D. Description of Criteria

First, I will be assessing the user interface of a text editor: Sublime Text, and an IDE: Eclipse for Java development. It is important to inform the user of the UI because it is what the programmer will look at the entire time while using the application. Secondly, I will analyze the features of Sublime Text versus Eclipse. It is essential to know what the two can do, and what is the full potential that can be done by using either or. Lastly, I will assess the accessibility and performance of the text editor and IDE. The reader should consider what device they primarily use and if this application can be used on their operating system of choice. It is good to know how and where to download these devices and whether they cost money. Performance-wise, it is important to analyze if one is faster than the other, both in download and execution. I will also analyze the reliability of the two. It is important to consider what project one might be working on and whether the application can handle it, or if it will crash altogether.

II. ANALYSIS OF CRITERIA

A. User Interface

1). Sublime Text: User Interface, or UI, is the space where human and computer interaction occurs. A text editor, such as Sublime Text, has a very minimalistic user interface. A text editor is essentially a place to write text. Upon launching the application, there is a blank screen and a cursor blinker, prompting the user to begin typing. The creator of Sublime Text, Jon Skinner, created this text editor with the thought of ease and functionality. He states, “The UI toolkit sits on top of a cross-platform abstraction layer, which is more a union of platform functionality rather than [the] lowest common denominator.” [8]. Sublime Text (as with other text editors) was created to be user-friendly and easy to manage. For a beginner Java developer, programming in Sublime Text can begin from launch.

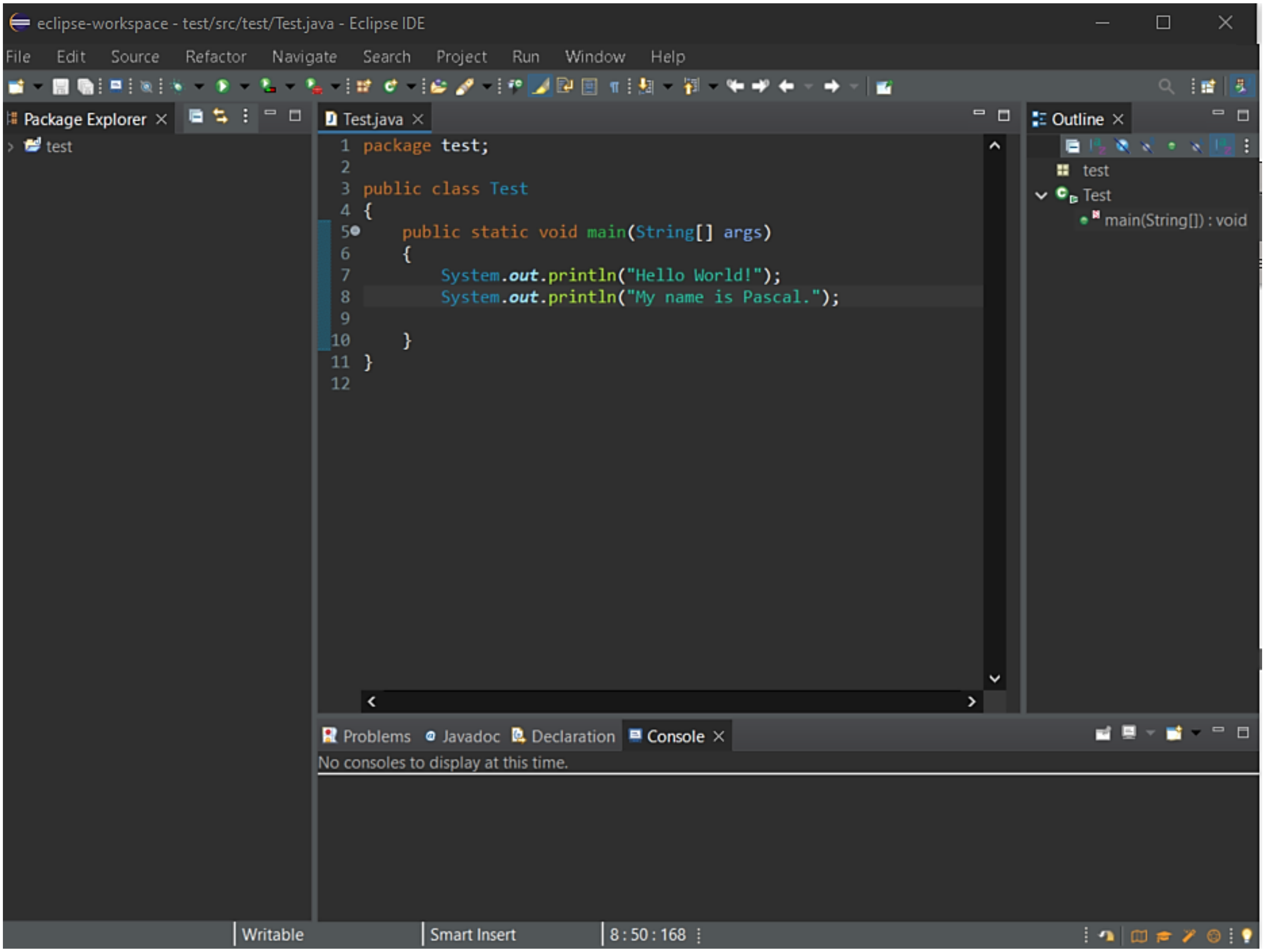


```java
public class helloworld
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!!!");
        System.out.println("My name is Pascal.");
    }
}
```

```
[Finished in 1.0s]
```

Figure 1: The User Interface of Sublime Text

2). Eclipse: Eclipse, as with many IDEs, was created to help the programmer. There are many features, that will be discussed later in this report, that Eclipse contains. Due to this, however, there are a plethora of buttons, menus, and columns to analyze. For a beginner programmer, this can easily be overwhelming. The Eclipse Foundation states, “For most developers, an introduction to the platform can be overwhelming...” [9]. The many buttons and places to look should come as no surprise as Eclipse is a development environment. There is a built-in terminal, allowing the programmer to compile and debug code quickly. Where in contrast, Sublime Text does not contain one.



```java
package test;

public class Test
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
        System.out.println("My name is Pascal.");

    }
}

```

No consoles to display at this time.

Figure 2: Eclipse IDE User Interface.

B. Features

1). Sublime Text: A text editor such as Sublime Text was created to markup, write code, and compile the code in a terminal. Sublime Text itself does have the capability to build and run code in a terminal, however, it requires the user to build the system using JSON format. For a beginner programmer, this is very difficult to do. Instead, the easier route would be to open the command prompt, or terminal for macOS, and compile the program there using 'javac'. By learning how to use the terminal, the programmer now has insight into the core architecture of a computer and the true power of the terminal. Fellow full-stack developer Samuel Martins states, "as a beginner, you want to be testing your programming and seeing the crash output. The console/terminal is your friend." [10]. The features of a text editor can be summarized into a list: syntax highlighting, code formatting, editing, file splitting, and support for cross platforms. [11]. These features allow for easy markup and code editing as there is no distraction in the user interface as described above.

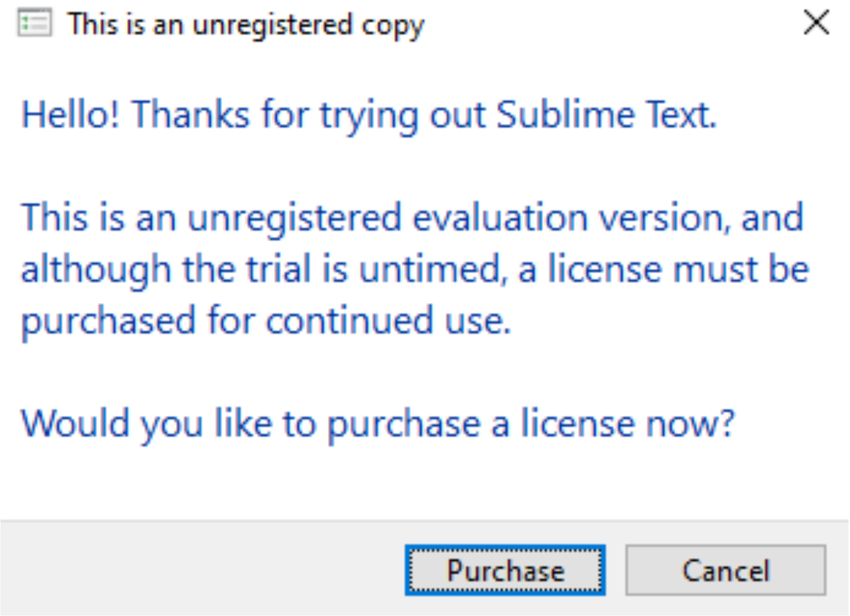
2). Eclipse: An IDE such as Eclipse was created for big projects. Its many features were implemented to do nothing but aid the programmer as they write their code. Essential Java IDE tools include debugging, syntax highlighting, build automation, code completion, and graphical user interface (GUI). With Eclipse, in particular, there is also an easy Git and Apache Maven integration, and a marketplace for installing plugins. [12]. All these features combined are perfect for the advanced programmer developing projects for a company, but for a beginner programmer just starting off, most of them will not be used. What will be used, and perhaps abused, is the auto code completion that many IDEs have. Code completion occurs as you write code, oftentimes giving suggestions as to what you are about to write, and then completing the statement for you, or even correcting syntax errors that may happen along the way. This feature is crucial for advanced programmers as it saves them time and lowers the risk for typos or bugs. For beginner programmers, this feature can have an impact on their comprehension of the language. Auto-correcting errors do not allow the coder to know what they are doing wrong. Martins claims “having help from an IDE will make you a lazy coder…once you get more advanced, using an IDE is fine” [10].

C. Accessibility and Performance

1). Sublime Text: Sublime Text is available to download via their website and it is comparable with Windows, Mac OS, and Linux. The application itself is very lightweight, costing only 22MB of space for installation [13]. Sublime Text comes in many different themes, and it is most known for its colorful presentation. There is a dark mode and a light mode. For me, when I first launched the application, it came in the blue-tinted mode as seen in Figure 1. It was very easy on the eyes. The downfall in its accessibility is its constant prompt to purchase the application (Figure 3). This prompt will occur after several saves. The cost of Sublime Text is $80 USD. This is discouraging since it can be irritating to the coder after it occurs many times. Performance-wise, Sublime Text is fast. Jessica Cregg with ITPro states, “It loads pretty much instantly, is incredibly fast, and legendarily stable.” [14]. I agree with this claim as download and launch time were almost instantaneous. This speed comes from the fact that it is a text editor, its simplicity compared to IDEs allows for less memory and faster loading times. [15].

2). Eclipse: Like Sublime Text, Eclipse is available for Windows, Mac OS, and Linux and can be downloaded from their website. Unlike Sublime Text, Eclipse is completely free to use and does not have a prompt to purchase. They do, however, display a prompt once upon downloading to donate. Being that it is an IDE with many features, it is to no surprise that it downloads slower and takes more memory than Sublime Text. Erin Schaffer reveals that although Eclipse is one of the best IDEs to use for Java development, it has a “reputation for running slowly, uses a lot of CPU, a heavy tool with frequent crashes, and a complicated plugin environment” [12]. With its many features and capabilities to aid the programmer, the programmer can be negatively affected by them as well.

Figure 3: Sublime Text Purchase Prompt.


This is an unregistered copy

Hello! Thanks for trying out Sublime Text.

This is an unregistered evaluation version, and although the trial is untimed, a license must be purchased for continued use.

Would you like to purchase a license now?

III. RECOMMENDATION

For beginner programmers just starting their long journey into the world of technology, it is important to learn the core concepts of how to code. Knowing the syntax of the language you are using and how to compile and debug using a terminal does nothing other than benefit you long term. Although IDEs are very beneficial in speed and catching errors, if the programmer does not know how to catch the errors themselves, then they are, just as Martins claimed, “lazy coders.” As stated previously, many employers are more interested in what a programmer’s coding capabilities are. Even if they may not have the education as another programmer, if they can write better code than them, they have a higher chance of getting the job. It is important to note that this report is focused primarily on beginner programmers. For big, interactive projects, an IDE was designed to handle it. For beginner programmers, utilizing a text editor is better due to the is speed and lightweight. After all, a proficient coder is one that knows how to code.

This recommendation was created based on two key factors: performance, and user interface. When you first are learning how to code, you are excited to start! If it takes a while to download and launch the application, you might feel discouraged. If the application consistently crashed while trying to run your code, you would get more and more frustrated. Although IDEs are essential for big projects, when you are a beginner the projects you work on are relatively small. As for user interface, and IDE does have many buttons and they are intimidating to beginners. You fear clicking the wrong button will erase everything you have worked on, or worse. Thus, the simplicity of a text editor is essential for a beginner. There are not many buttons to be seen, just a blinking cursor. Learning how to truly start from nothing, and work and build something is very rewarding! It is important to note that Professors who are teaching a programming fundamentals course should emphasize the use of text editors before introducing IDEs. Many professors do not have a preference, so long as the program is written. This mindset does not benefit the students at all. Knowing the difference between the two makes you a better programmer. And once you begin using an IDE, you know for a fact that you are no longer a beginner.

References

[1] V. Anjos, “Are developers getting smarter or dumber?”, *linkedin.com,* May 30, 2018. [Online]. Available: https://www.linkedin.com/pulse/developers-getting-smarter-dumber-victor-anjos/. [Accessed on Feb. 15, 2022].

[2] Elye, “The Evolution of Coding,” *betterprogramming.pub*, Sept. 7, 2020. [Online]. Available: https://betterprogramming.pub/the-evolution-of-coding-profession-45eab9397466. [Accessed on Feb 15, 2022].

[3] A. Augustyn et al., “Java computer programming language,” *britannica.com*, Dec. 17, 2008. [Online]. Available: https://www.britannica.com/technology/Java-computer-programming-language/additional-info#history. [Accessed on Feb 15, 2022].

[4] I. Tkach et al., “How Many Software Developers Are in the US and the World?”, *daxx.com,* Sept 23, 2021. [Online]. Available: https://www.daxx.com/blog/development-trends/number-software-developers-world. [Accessed on Feb 16, 2022].

[5] “TIOBE Index for February 2022”, *tiobe.com*, Feb. 2022. [Online]. Available: https://www.tiobe.com/tiobe-index/. [Accessed on Feb 18, 2022].

[6] B. King, “The History of Modern Text Editors,” *blog.grio.com,* Jan. 20, 2020. [Online]. Available: https://blog.grio.com/2020/07/the-history-of-modern-text-editors.html. [Accessed on Feb 17, 2022].

[7] “Know your history – IDEs are a developer’s best friend”, *jaxenter.com,* Nov. 14, 2018. [Online]. Available: https://jaxenter.com/know-your-history-ides-151761.html#authors-block. [Accessed on Feb 17, 2022].

[8] J. Skinner, “Hacker News”, *news.ycombinator.com*, July 29, 2011. [Online]. Available: https://news.ycombinator.com/item?id=2822114. [Accessed on Feb 16, 2022].

[9] J. Li et al., “User Interface Guidelines”, *wiki.eclipse.org*, Nov. 2007. [Online]. Available: https://wiki.eclipse.org/User_Interface_Guidelines. [Accessed on Feb 17, 2022].

[10] S. Martins, “Beginner Coders, Here’s 5 Reasons Why You Should Stop Using IDEs,” *javascript.plainenglish.io*, Nov. 30, 2020. [Online]. Available: https://javascript.plainenglish.io/beginner-coders-heres-5-reasons-why-you-should-stop-using-ides-8bfbaf1d4220. [Accessed on Feb 16, 2022].

[11] S. Kumar, “Text Editor vs IDE: Which is Best for Beginner Programmers?”, *hackernoon.com*, Jan. 20, 2021. [Online]. Available: https://hackernoon.com/text-editor-vs-ide-which-is-best-for-beginner-programmers-7mae315u. [Accessed on Feb 18, 2022].

[12] E. Schaffer, “Top 10 best Java IDEs for 2022,” *educative.io*, Jan 03, 2022. [Online]. Available: https://www.educative.io/blog/best-java-ides-2021. [Accessed on Feb 18, 2022].

[13] N. Schäferhoff, “The Sublime Text Code Editor - An In-Depth Review,” elegantthemes.com, Dec. 22, 2016. [Online]. Available: https://www.elegantthemes.com/blog/resources/the-sublime-text-code-editor-an-in-depth-review. [Accessed on Feb 18, 2022].

[14] J. Cregg, “Best IDEs: The perfect code editors for beginners and professionals,” itpro.com, Jun. 28, 2021. [Online]. Available: https://www.itpro.com/software/360026/best-ides-the-perfect-code-editors-for-beginners-and-professionals#:~:text=It%20doesn't%20take%20long,use%20on%20a%20really%20bulky. [Accessed on Feb 17, 2022].

[15] K. Matthews, “Which are better for developers, IDEs or text editors?”, jaxenter.com, Sept. 4, 2018. [Online]. Available: https://jaxenter.com/ides-vs-text-editor-148936.html. [Accessed on Feb 19, 2022].