GitHub: https://github.com/Doraemon012/Saturn

Saturn IDE

A code editor developed in JAVA

By: Neyati

Contents

- 1. Introduction
- 2. Tech Stack & Concepts used
- 3. Features
- 4. Code Samples

```
/home/neyati/Desktop/ide-test/python.py - Saturn IDE
File Edit Format Theme
                                                                                               ▶
  # Python Code
  rows = 5
  for i in range(rows):
       # Print stars
       for j in range(i + 1):
            print('*', end='')
       print()
Exited with error code : 0
                           Path: /home/neyati/Desktop/ide-test/python.py Lang: .py
```



Welcome to **Saturn IDE**, a versatile code editor designed to empower your coding journey with simplicity and efficiency.

Developed entirely in **Java**, Saturn IDE boasts a clean and minimalist user interface, making it the perfect tool for developers of all levels.

Whether you're coding in C++, C, Python, or Java, Saturn IDE provides a seamless and enjoyable coding experience, tailored to meet your needs.

Works on Windows, Linux

Tech Stack

Java

Java Swing

Java Awt

Concepts

File Handling

OOPs Concepts

Exception Handling

Features

Syntax Highlighting

Enhance code readability with vibrant syntax highlighting for languages like C++, C, Python, and Java.

Dark and Light Modes

Personalize your coding environment with both dark and light modes, ensuring comfort and visual appeal.

•Basic Editing Functions 哭 📋

Perform essential code manipulations effortlessly with cut, copy, paste, and select all functionalities.

Print Code with a Click

Streamline the process of printing your code with a single click, making documentation and reference easier.

Automatic Interpreter/Compiler Detection

Saturn IDE intelligently identifies the required interpreter or compiler for your code, removing the need for manual selection.

Syntax Highlighting

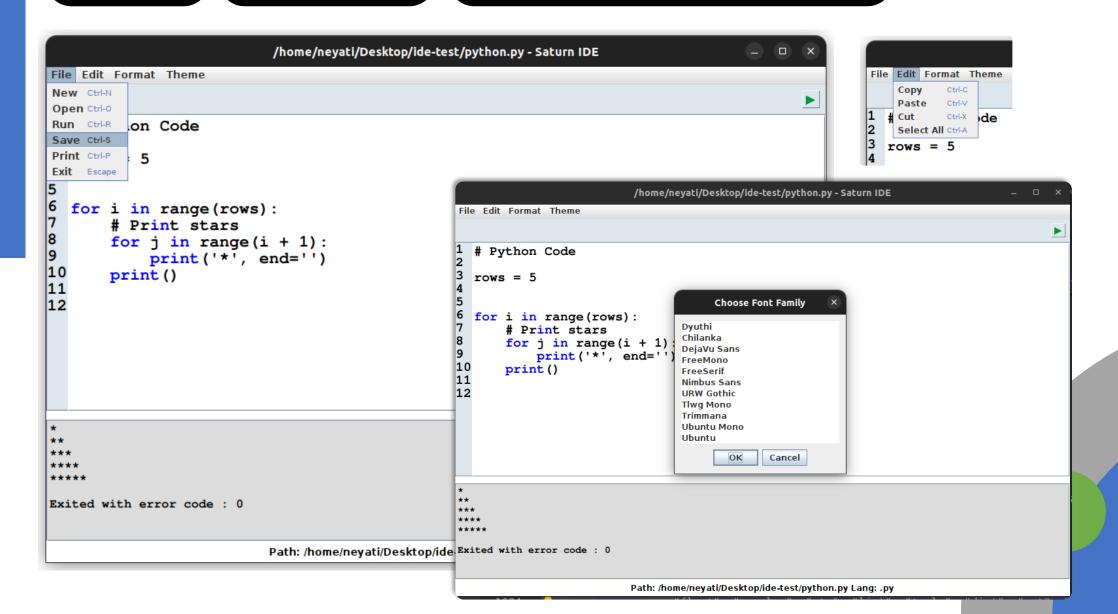
```
/home/neyati/Desktop/ide-test/python.py - Saturn IDE
File Edit Format Theme
                                                                                   /home/neyati/Desktop/ide-test/python.py - Saturn IDE
   # Python Code
                                                      File Edit Format Theme
2
3
4
5
6
   rows = 5
                                                         # Python Code
   for i in range(rows):
                                                         rows = 5
        # Print stars
        for j in range(i + 1):
            print('*', end='')
                                                         for i in range(rows):
10
11
12
        print()
                                                              # Print stars
                                                              for j in range(i + 1):
                                                                   print('*', end='')
                                                              print()
***
****
****
                                                      **
Exited with error code : 0
                                                       ***
                                                       ****
                                                      ****
                            Path: /home/neyati/Desktop/ide-te
```

Exited with error code : 0

Print Code

Editing Features

Auto Compiler/Interpreter Detection



Auto Compiler/Interpreter Detection

Code Example: Detecting C++ Compiler

```
public static String getCppInterpreterPath() {
       try {
           Process process = Runtime.getRuntime().exec("g++ --version");
           BufferedReader reader = new BufferedReader(new InputStreamReader(process.getInputStream()));
           int exitCode = process.waitFor();
           exit_code = "Exited with error code : " + exitCode;
           cpp_name = "q++";
           System.out.println("installed hai");
        } catch (IOException | InterruptedException e1) {
           showNotInstalledAlert("C++ (g++)");
       return cpp_name;
```

Executing Code

Code Example: Executing C++ Code

```
. .
                                                                           On other OS
else {
               Process process1 = Runtime.getRuntime().exec("/bin/bash -c cd " + cd_folder);
               Process process_compile = Runtime.getRuntime().exec(cppF + " " + file_ok + ".cpp " +
"-o " + file_ok);
               BufferedReader reader = new BufferedReader(new
InputStreamReader(process_compile.getInputStream()));
               String line1:
               String compileLine;
               while ((line1 = reader.readLine()) != null) {
                   out += line1 + "\n";
               int exitCode = process_compile.waitFor();
               Process process_run = Runtime.getRuntime().exec("/bin/bash -c ./" + file_ok);
               BufferedReader reader2 = new BufferedReader(new
InputStreamReader(process_run.getInputStream()));
               String line2;
               while ((line2 = reader2.readLine()) != null) {
                   out += line2 + "\n";
               int exitCode2 = process_compile.waitFor();
               exit_code = "Complied with error code : " + exitCode + "\n" + "Exited with error
code : " + exitCode2:
               System.out.println(out + " " + cpp_name);
```

On Windows

```
public static String runCppFileCommand(String cppF, String file path) {
        boolean isWindows = System.getProperty("os.name").toLowerCase().contains("win");
       String out = "":
        try {
           String cd_folder = extractContainingFolder(file_path);
           String file ok = extractFileName(file path);
           if (isWindows) {
                Process process1 = Runtime.getRuntime().exec("cmd.exe /c " + "cd " + cd_folder);
               Process process_compile = Runtime.getRuntime().exec(cppF + " " + file ok + ".cpp" +
"-o " + file_ok);
               Process process_run = Runtime.getRuntime().exec("./ " + file_ok);
                BufferedReader reader = new BufferedReader(new
InputStreamReader(process_compile.getInputStream()));
               String line1:
               while ((line1 = reader.readLine()) != null) {
                   out += line1 + "\n";
                int exitCode = process_compile.waitFor();
               BufferedReader reader2 = new BufferedReader(new
InputStreamReader(process_run.getInputStream()));
               String line2;
               while ((line2 = reader.readLine()) != null) {
                   out += line2 + "\n";
                int exitCode2 = process_compile.waitFor();
                exit code = "Complied with error code : " + exitCode + "\n" + "Exited with error
code : " + exitCode2;
        } catch (IOException | InterruptedException e) {
           e.printStackTrace();
        return out + "\n" + exit_code;
```

Supported languages

Java

Python

C++

Supported Operating Systems

Windows

Linux

Thank you!

By: Neyati

GitHub: https://github.com/Doraemon012/Saturn