

Submitted By,

Sharmika Das Banhi

Student ID: 210204

Computer Science & Engineering Discipline

Khulna University, Khulna

Submitted To,

Amit Kumar Mondal

Associate Professor

Computer Science & Engineering Discipline

Khulna University, Khulna

**An Assignment on Extract various Artifacts (such as source code, commit messages and so on) of the following open-source software systems:**

1. **Azure Java SDK**
2. **ChatGPT**

**Azure Java SDK**

**Source Code: GitHub Repository**

The source code for Azure Java SDK is the human-readable version of the software, written in Java. It comprises all the instructions that are executed by the computer to enable Java applications to interact with various Azure services programmatically.

The source code serves as the foundation for the Azure Java SDK, defining the logic and functionality needed for developers to seamlessly integrate Azure services into their Java applications. It allows developers to understand, modify, and contribute to the SDK.

The source code is developed by the Microsoft Azure team, with contributions from the broader community of developers. The development process involves writing Java code, designing libraries, and ensuring compatibility with different Azure services. Collaboration is facilitated through version control tools like Git, and the code is hosted on GitHub for community involvement.

**Core Executable:**

The core executable for Azure Java SDK is the compiled version of its source code. It typically includes Java Archive (JAR) files that can be integrated into Java applications.

? Developers include the core executable in their Java applications to programmatically interact with Azure services. It simplifies tasks such as making API calls, managing resources, and handling authentication, providing a convenient interface for working with Azure.

The core executable is created through the compilation process, converting the Java source code into machine-readable JAR files. Build tools like Apache Maven or Gradle are commonly used for this purpose.

**Codebase:**

The codebase of Azure Java SDK encompasses all the source code files, including Java classes, configuration files, and dependencies that collectively form the SDK.

The codebase defines the functionality and behavior of the Azure Java SDK. It serves as the foundation for developers to integrate Azure services into their Java applications seamlessly.

The codebase is developed and maintained by the Microsoft Azure team. Developers write Java code, design and organize classes, and ensure proper documentation. Contributions and collaborative efforts occur through version control systems like Git and collaborative platforms such as GitHub.

**Implemented Software Code:**

The implemented software code of Azure Java SDK refers to the specific instructions and logic written in Java that make up the SDK.

This code implements the features and functionalities of the Azure Java SDK, allowing developers to interact with Azure services in their Java applications.

The implemented code is created by the developers at Microsoft and the open-source community. It involves writing Java code that encapsulates the necessary functionalities for different Azure services.

**Programmers and Tools Write or Create It:**

Programmers: The Azure Java SDK is primarily written and maintained by developers employed by Microsoft. Contributions come from programmers who work on designing and implementing the SDK's functionalities.

Tools: Development tools include Java IDEs, build tools like Apache Maven or Gradle, and version control tools like Git and GitHub. These tools enhance collaboration, efficiency, and the overall development process.

**ChatGPT**

**Source Code: GitHub Repository**

The source code for ChatGPT is the human-readable version of the software, typically implemented in a machine learning framework like TensorFlow or PyTorch. It contains the instructions and model architecture that enable the ChatGPT model to understand and generate natural language responses.

The source code forms the foundation of ChatGPT, defining the neural network architecture and logic needed for natural language processing. It allows developers and researchers to comprehend, modify, and contribute to the ChatGPT project.

The source code is created by researchers, machine learning engineers, and contributors involved in the ChatGPT project. It involves writing code in machine learning frameworks, designing and training the neural network model, and implementing algorithms for language generation. Collaboration occurs through version control tools like Git, and the code is hosted on GitHub for community engagement.

**Core Executable:**

The core executable for ChatGPT could be a model file or a deployed service that allows end-users to interact with the ChatGPT model.

End-users utilize the core executable to engage in conversations with the ChatGPT model. It abstracts the complexities of the underlying machine learning model and provides a user-friendly interface for generating natural language responses.

The core executable may involve deploying a pre-trained model as a service. The creation process includes training the ChatGPT model on large datasets, converting it to a deployable format, and integrating it with an interface for user interaction. Tools like TensorFlow Serving or custom deployment scripts may be used in this context.

**Codebase:**

The codebase of ChatGPT comprises all the source code files, including scripts, configurations, and dependencies that constitute the ChatGPT project.

The codebase defines the functionality and behaviour of the ChatGPT model. It serves as the basis for developers to enhance the capabilities of the model, improve language understanding, and contribute to the project.

The codebase is typically developed and maintained by researchers and engineers working on the ChatGPT project. It involves writing code in machine learning frameworks, refining the model architecture, and incorporating improvements based on research findings. Contributions may come from the open-source community interested in advancing natural language processing.

**Implemented Software Code:**

The implemented software code of ChatGPT refers to the specific instructions and logic written in the chosen machine learning framework (e.g., TensorFlow, PyTorch) that make up the ChatGPT model.

This code implements the natural language processing capabilities of ChatGPT, allowing it to understand input and generate contextually relevant responses in a conversational manner.

The implemented code is created by researchers, machine learning engineers, and developers involved in the ChatGPT project. It involves writing code that defines the architecture of the neural network, training the model on diverse datasets, and fine-tuning it for language generation tasks.

**Programmers and Tools Write or Create It:**

Programmers: The ChatGPT codebase is primarily written and maintained by researchers, machine learning engineers, and developers. Contributions come from individuals and organizations interested in advancing natural language processing capabilities.

Tools: Development tools include machine learning frameworks (e.g., TensorFlow, PyTorch), version control tools like Git, and collaborative platforms like GitHub. These tools facilitate collaboration, experimentation, and the overall development process in the field of natural language processing.