

РЕАЛИЗАЦИЯ ИНТЕРФЕЙСОВ ДЛЯ ПРИКЛАДНЫХ ЗАДАЧ НА JAVA И PYTHON

Машков В.В., Филиппов К.Е.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Научный руководитель: Василькова А.Н. – ассистент кафедры ИПиЭ

Аннотация. Рассмотрено применение API и библиотек различных языков программирования, а также в эргономике. Приведены решения конкретных задач на таких языках программирования как Java, Python. Указаны преимущества использования библиотек и API.

Ключевые слова: API, библиотека, язык программирования, приложение, данные, доступ.

Введение. Интерфейсы прикладного программирования (API) – это наборы функций и процедур, которые позволяют разработчикам создавать приложения на основе веб-сервисов. Эти функции позволяют программистам получать доступ к данным безопасным и контролируемым способом. Библиотеки языков программирования позволяют программистам использовать заранее написанный код для выполнения определенных задач, что экономит их время и усилия. Библиотеки могут быть составлены из функций или классов, которые имеют возможность были вызваны программистом при выполнении конкретного дела. API и библиотеки различных языков программирования используются для создания онлайн-приложений.

Основная часть. Разработчики могут использовать API для доступа к данным с удаленного сервера, а библиотеки различных языков программирования используются для упрощения программирования. Эти инструменты необходимы для современных разработчиков веб-приложений [1]. Возможность использовать API и библиотеку на различных языках позволяет создателям быстро, легко получить доступ к данным, сервисам и другим ресурсам. Это может быть особенно полезно при работе над проектами, в которых используется несколько языков. Например, если для проекта требуется веб-приложение, написанное на языке Python, но при этом необходимо получить доступ к базе данных Oracle, разработчик может использовать Oracle API для Python для подключения к базе данных без необходимости изучать синтаксис языка. Аналогично, если для проекта требуется приложение для iOS, написанное на Objective-C, но при этом необходимо получить доступ к ресурсу Amazon Web Services (AWS), разработчик может использовать AWS SDK для Objective-C для доступа к сервису без необходимости изучать синтаксис языка.

API – это набор конечных точек, которые позволяют пользователям взаимодействовать с системой. Эти конечные точки предоставляют доступ к данным и функциональности системы и позволяют пользователям создавать, читать, обновлять и удалять данные. Доступ к конечным точкам может осуществляться различными методами, включая HTTP-запросы, веб-службы и интерфейс командной строки. Кроме того, API может предоставлять доступ к ряду инструментов и библиотек, которые можно использовать для разработки приложений, взаимодействующих с системой.

Использование API и библиотек на разных языках не только облегчает разработчикам доступ к данным и сервисам из разных источников, но и позволяет им создавать приложения, не зависящие от платформы. Это означает, что одно и то же приложение может использоваться на нескольких платформах без необходимости переписывать код.

API с точки зрения эргономики – это изучение и проектирование наилучшего возможного пользовательского опыта с помощью современных методов программирования. Он включает в себя разработку веб-сервисов и интерфейсов прикладного программирования

(API) для облегчения взаимодействия пользователей с системой. API позволяют пользователям быстро и легко получать доступ к данным и услугам без необходимости изучать язык программирования или ориентироваться в сложном пользовательском интерфейсе.

Суть заключается в том, чтобы позволить пользователям делать запросы к серверу и получать ответы в ответ. Эти запросы могут быть любыми - от простого GET-запроса до более сложного POST-запроса, содержащего данные. Затем сервер обрабатывает запрос и отправляет ответ с соответствующими данными. Этот ответ может быть в виде HTML-страницы, объекта JSON, XML-документа или любого другого формата, который сервер настроен на обработку.

Лучший способ реализовать API на любом языке программирования - использовать библиотечные функции, классы и объекты языка. В зависимости от языка программирования, библиотечные функции, классы и объекты могут быть использованы для создания необходимых объектов, методов и функций для API. Например, в Java можно использовать класс `URLConnection` для создания HTTP-запроса к API и класс `JSONObject` для разбора ответа от API. В Python можно использовать библиотеку `requests` для создания HTTP-запросов и библиотеку `json` для разбора ответа.

В Python для доступа к API обычно используется библиотека `requests`. Многие разработчики также используют популярный веб-фреймворк `Flask` для создания API. В JavaScript для работы с API разработчики могут использовать `Fetch API` или библиотеку `Axios`. Разработчики Java используют для доступа к API такие библиотеки, как `Apache HttpClient` или `Unirest`. Те, кто использует C#, имеют доступ к библиотеке `HttpClient`.

Эти технологии позволяют разработчикам создавать приложения и сервисы, которые взаимодействуют с другим программным обеспечением или веб-сервисами, не требуя прямого взаимодействия с пользователем, также можно использовать для доступа к данным или для выполнения конкретных задач, таких как осуществление платежей или отправка сообщений. API используются для доступа к сторонним сервисам, таким как переводческие услуги или информация о погоде. Вызовы API обычно осуществляются с помощью языка программирования, такого как JavaScript, Python или Java. Если смотреть современнее, то часто используют архитектуру REST (Representational State Transfer) и формат данных JSON (JavaScript Object Notation). Аутентификация API используется для обеспечения доступа к API только авторизованных пользователей и безопасности данных.

Для реализации интерфейсов на Java используются: `Collection` – определяет общие методы для всех классов, реализующих коллекции объектов. `Map` – определяет общие методы для всех классов, реализующих отображения объектов. `ActionListener` – определяет метод `actionPerformed()`, который вызывается при событии. `MouseListener` – определяет методы `mouseClicked()`, `mouseEntered()`, `mouseExited()`, `mousePressed()` и `mouseReleased()`, которые вызываются при соответствующих действиях мыши. `Runnable` – определяет метод `run()`, который вызывается при старте потока. `Comparable` – определяет метод `compareTo()`, который используется для сравнения объектов. `Serializable` – используется для сериализации объектов.

Для реализации интерфейсов на Python используются: `Generator` – определяет функцию-генератор, которая возвращает последовательность значений. `Context Manager` – определяет методы `enter()` и `exit()`, которые используются для управления контекстом выполнения программы. `Decorator` – определяет функцию-декоратор, которая добавляет дополнительный функционал к другой функции. `Abstract Base Class` – определяет абстрактный класс, который содержит абстрактные методы и свойства, которые должны быть реализованы в дочерних классах. `Callable` – определяет метод `call()`, который позволяет объекту быть вызываемым. `Iterable` – определяет метод `iter()`, который возвращает итератор. `Iterator` – определяет методы `next()` и `iter()`, которые используются для итерации по объекту.

`Py4J` – это мост между Python и Java, который позволяет программам Python взаимодействовать с объектами и методами Java. Он позволяет Python вызывать код Java и

использовать библиотеки Java так же, как и вызывать код Python. Py4J является программным обеспечением с открытым исходным кодом и доступен по лицензии Apache License 2.0. Он был создан для того, чтобы разработчики Python могли использовать возможности Java и при этом использовать язык Python. Py4J широко используется в проектах в области науки о данных, машинного обучения и больших данных, где Python является предпочтительным языком для анализа данных, а Java – предпочтительным языком для распределенных систем и обработки больших данных. Py4J предоставляет механизм для связи между Java и Python, позволяя вызывать методы Java из Python и наоборот. Это позволяет создавать гибкие системы, в которых можно использовать лучшее из обоих языков.

В Java интерфейсы играют важную роль в объектно-ориентированном программировании. Они позволяют объединять классы с общими методами и свойствами, не привязываясь к конкретной реализации. Реализация интерфейсов в Java происходит с помощью ключевого слова `implements`, которое позволяет классу наследовать все методы и свойства интерфейса.

Заключение. Языки программирования интересны тем, что позволяют решать сложные задачи, создавать новые приложения и исследовать новые способы выполнения задач. Они обеспечивают мощный способ выражения идей и воплощения их в жизнь. Они также позволяют развивать навыки, которые могут быть использованы в других областях, таких как математика, инженерия и даже искусство. Кроме того, языки программирования постоянно развиваются и предлагают постоянно растущий набор функций и возможностей. Большинство современных языков программирования поддерживают API и облегчают разработчикам доступ к ним. Такие популярные языки, как Python, JavaScript, Java и C#, имеют библиотеки, облегчающие работу с API.

В целом, использование API и библиотек на разных языках позволяет ускорить и повысить эффективность разработки. Предоставляя разработчикам доступ к данным и сервисам на разных языках без необходимости изучать синтаксис каждого языка, они могут сосредоточиться на создании приложений, которые не зависят от платформы и могут использоваться на разных платформах.

Список литературы

1. Rachel Emmer, *how to use API in your Application* [Электронный ресурс]. – Режим доступа: <https://rachelaemmer.medium.com/how-to-use-the-google-books-api-in-your-s-application-17a0ed7fa857>. – Дата доступа: 24.02.2021
2. Swagger, *Best Practices in API* [Электронный ресурс]. – Режим доступа: <https://swagger.io/resources/articles/best-practices-in-api-design/Rachel%20Emmer,%20How%20to%20use%20API%20in%20your%20Application>. – Дата доступа 11.04.2020

UDC 004.451.84+004.432

IMPLEMENTATION OF INTERFACES FOR APPLIED TASKS FOR JAVA AND PYTHON

V.V. Mashkov, K.E. Filippov

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Vasilkova A.N. – assistant of the Department of EPE

Annotation. Application of API and libraries of different programming languages and also in ergonomics is considered. Solutions of specific tasks in such programming languages as Java, Python are given. The advantages of using libraries and APIs are indicated.

Keywords: API, library, programming language, application, data, access.