INTRODUCTION TO OOPS AND FILE HANDLING IN JAVA

Java internship by Chethana Kothepalle

INTRODUCTION

This presentation provides a comprehensive introduction to **Object-Oriented Programming** and **File Handling** in Java. It covers the basics of OOP concepts, including inheritance, polymorphism, and encapsulation, as well as file handling techniques such as reading and writing data to files. By the end of this presentation, you will have a solid understanding of how to use OOP and file handling in Java to create robust applications.

WHAT IS OBJECT-ORIENTED PROGRAMMING?

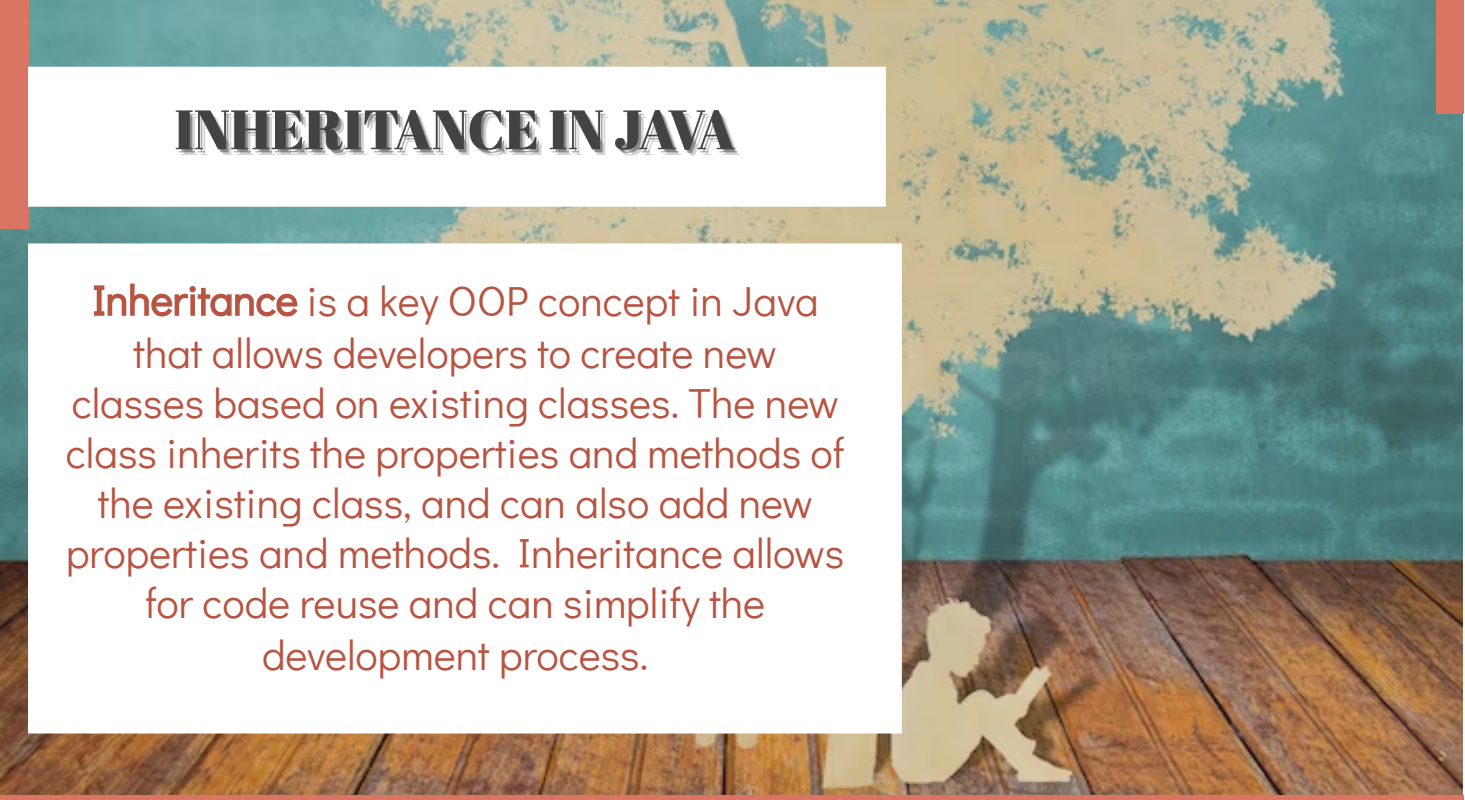
Object-Oriented Programming (OOP) is a programming paradigm that is based on the concept of objects, which can contain data and code to manipulate that data. OOP is based on four principles: **encapsulation**, **abstraction**, **inheritance**, and **polymorphism**. These principles allow developers to create reusable and maintainable code. Java is an OOP language that supports these principles.

ENCAPSULATION AND ABSTRACTION IN JAVA

In Java, **encapsulation** is the process of hiding implementation details from the user and providing a simple interface to interact with the object. **Abstraction** is the process of hiding complexity and providing a simple interface to the user. Together, encapsulation and abstraction allow developers to create objects that are easy to use and maintain.

INHERITANCE IN JAVA

Inheritance is a key OOP concept in Java that allows developers to create new classes based on existing classes. The new class inherits the properties and methods of the existing class, and can also add new properties and methods. Inheritance allows for code reuse and can simplify the development process.

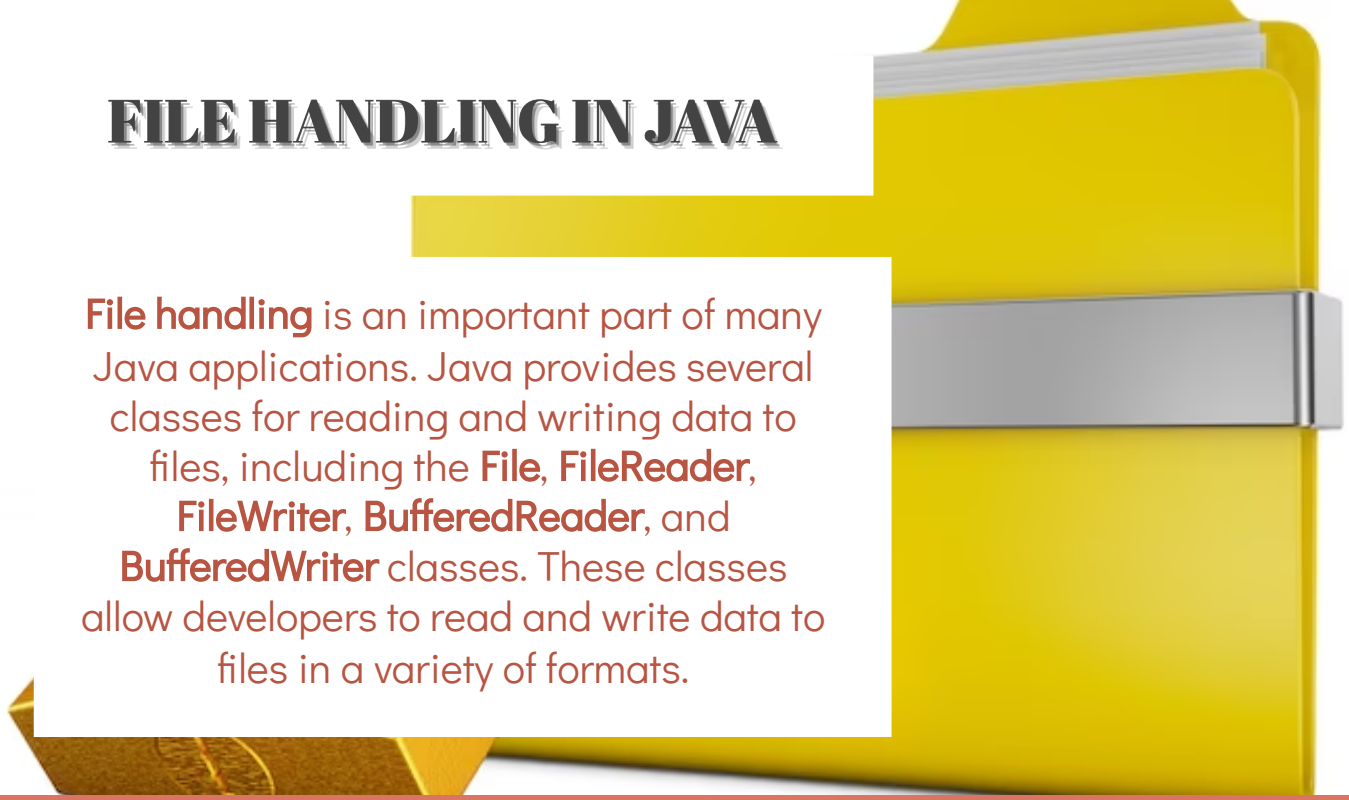


POLYMORPHISM IN JAVA

Polymorphism is the ability of an object to take on many forms. In Java, polymorphism allows developers to write code that can work with objects of different types, as long as they share a common interface. This allows for more flexible and reusable code.

FILE HANDLING IN JAVA

File handling is an important part of many Java applications. Java provides several classes for reading and writing data to files, including the **File**, **FileReader**, **FileWriter**, **BufferedReader**, and **BufferedWriter** classes. These classes allow developers to read and write data to files in a variety of formats.

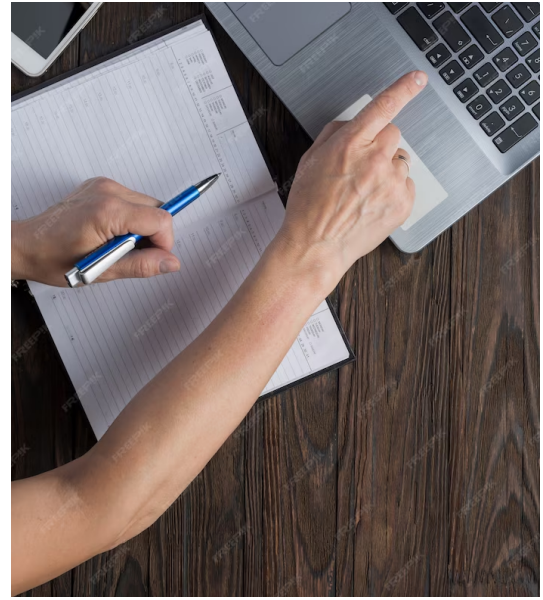


READING DATA FROM FILES IN JAVA

To read data from a file in Java, developers can use the **FileReader** and **BufferedReader** classes. The FileReader class reads data from a file one character at a time, while the BufferedReader class reads data from a file one line at a time. These classes make it easy to read data from files in a variety of formats.

WRITING DATA TO FILES IN JAVA

To write data to a file in Java, developers can use the **FileWriter** and **BufferedWriter** classes. The FileWriter class writes data to a file one character at a time, while the BufferedWriter class writes data to a file one line at a time. These classes make it easy to write data to files in a variety of formats.



BEST PRACTICES FOR OOP AND FILE HANDLING IN JAVA

When working with OOP and file handling in Java, it is important to follow best practices to ensure that your code is maintainable and scalable. Some best practices include using meaningful variable names, commenting your code, and organizing your code into logical classes and methods. By following these best practices, you can create robust and efficient Java applications.

CONCLUSION

In conclusion, Object-Oriented Programming and File Handling are essential concepts in Java programming. By understanding these concepts and following best practices, developers can create robust and efficient Java applications. We hope this presentation has provided you with a comprehensive introduction to OOP and file handling in Java.

Thanks!