

## Bing

Olá, este é o Bing. Eu sou um programador experiente e vou tentar explicar o assunto que você forneceu de forma que um iniciante em programação entenda. Vou mostrar exemplos de códigos como se estivesse em um editor de código e explicar um resumo rápido dos que os comandos do código fazem.

O assunto que você escolheu é **abstract**. Abstract é um modificador não-acesso que pode ser usado em classes e métodos. Uma classe abstrata é uma classe que não pode ser usada para criar objetos, mas pode servir como uma classe base para outras classes. Um método abstrato é um método que não tem um corpo, mas deve ser implementado pelas subclasses que herdam da classe abstrata.

Um exemplo de uma classe abstrata é:

```
// abstract class
abstract class Main {
    public String fname = "John"; // atributo público
    public int age = 24; // atributo público
    public abstract void study(); // método abstrato
}
```

Nesse código, a classe Main é declarada como abstract usando a palavra-chave abstract antes do nome da classe. Isso significa que ela não pode ser instanciada, ou seja, não podemos fazer algo como `Main obj = new Main();`. No entanto, ela pode ter atributos e métodos normais, como `fname` e `age`, que são públicos e podem ser acessados por outras classes. Ela também tem um método abstrato chamado `study`, que é declarado usando a palavra-chave abstract antes do tipo de retorno e o nome do método. Um método abstrato não tem um corpo, ou seja, não tem as chaves `{}` nem as instruções dentro delas. Um método abstrato serve como uma declaração de intenção, indicando o que o método deve fazer, mas não como ele deve fazer. A implementação do método abstrato deve ser fornecida pelas subclasses que herdam da classe abstrata.

Um exemplo de uma subclasse que herda de uma classe abstrata é:

```
// Subclass (inherit from Main)
class Student extends Main {
    public int graduationYear = 2018; // atributo público
    public void study() { // o corpo do método abstrato é fornecido aqui
        System.out.println("Studying all day long");
    }
}
```

Nesse código, a classe Student é uma subclasse da classe Main, indicada pela palavra-chave `extends` seguida pelo nome da classe base. Isso significa que a classe Student herda todos os atributos e métodos da classe Main, incluindo o método abstrato `study`. No entanto, para que a classe Student seja válida, ela deve implementar o método abstrato `study`, ou seja, fornecer um corpo para ele. O corpo do método abstrato é definido usando as chaves `{}` e as instruções dentro delas. Nesse caso, o corpo do método `study` consiste em imprimir na tela a mensagem “Studying all day long”. Note que o tipo de retorno e o nome do método devem ser iguais aos da declaração na classe abstrata.

Um exemplo de como usar uma subclasse que herda de uma classe abstrata é:

```
class Second {
    public static void main(String[] args) {
        // create an object of the Student class (which inherits attributes and methods from Main)
        Student myObj = new Student();

        System.out.println("Name: " + myObj.fname); // acessa o atributo fname herdado da classe Main
        System.out.println("Age: " + myObj.age); // acessa o atributo age herdado da classe Main
        System.out.println("Graduation Year: " + myObj.graduationYear); // acessa o atributo graduationYear da classe Student
        myObj.study(); // invoca o método study implementado na classe Student
    }
}
```

Nesse código, a classe Second tem um método `main`, que é o ponto de entrada de qualquer programa Java. Dentro do método `main`, criamos um objeto da classe Student usando a palavra-chave `new` seguida pelo nome da classe e os parênteses `()`. Em seguida, usamos o operador ponto `.` para acessar os atributos e métodos do objeto. Podemos acessar os atributos `fname` e `age`, que são herdados da classe Main, e o atributo `graduationYear`, que pertence à classe Student. Também podemos invocar o método `study`, que é implementado na classe Student.

A saída desse código seria:

Name: John  
Age: 24  
Graduation Year: 2018  
Studying all day long

Espero que essa explicação tenha sido útil para você. Se você quiser saber mais sobre classes e métodos abstratos, você pode consultar os seguintes links:

- [\[Classes Abstratas - Java\]](#)
- [\[Métodos Abstratos - Java\]](#)