22．インターフェース

22.1 例）　// Interfaceの定義

interface AnimalIntface {

public void eat();

public void sleep();

}

// CatクラスがAnimalIntface interfaceを実装する

class Cat implements AnimalIntface {

public void eat() {

System.out.println("Cat eats fish.");

}

public void sleep() {

System.out.println("Cat sleeps in a warm place.");

}

}

// DogクラスがAnimalIntface interfaceを実装する

class Dog implements AnimalIntface {

public void eat() {

System.out.println("Dog eats meat.");

}

public void sleep() {

System.out.println("Dog sleeps in a doghouse.");

}

}

// Mainクラス

public class Main {

public static void main(String[] args) {

// CatとDogのインスタンスを作成

Cat myCat = new Cat();

Dog myDog = new Dog();

// eat()とsleep()メソッドを呼び出す

myCat.eat();

myCat.sleep();

myDog.eat();

myDog.sleep();

}

}

上記の例では、AnimalIntface interfaceを定義し、CatクラスとDogクラスがそれを実装します。CatとDogは、それぞれeat()とsleep()メソッドを持っています。Mainクラスでは、CatとDogのインスタンスを作成して、それらのメソッドを呼び出します。このように、interfaceを使用することで、異なるクラスで同じメソッド名を定義することができます。

interfaceを使う理由はいくつかあります。

22.2interfaceを使う理由は

1. 多様性の実現

interfaceを使用すると、同じinterfaceを実装する複数のクラスを作成できます。これにより、同じinterfaceを使って、異なるクラスを操作することができます。たとえば、AnimalIntface interfaceを実装するCatクラスとDogクラスを作成した場合、AnimalIntface型の変数を使用して、CatクラスのインスタンスまたはDogクラスのインスタンスを操作できます。このように、interfaceを使用することで、コードの柔軟性が向上します。

1. ポリモーフィズムの実現

interfaceを使用することで、ポリモーフィズムを実現できます。ポリモーフィズムとは、異なるクラスのインスタンスを同じ型として扱うことができることを意味します。たとえば、AnimalIntface interfaceを実装するCatクラスとDogクラスを作成した場合、AnimalIntface型の変数を使用して、CatクラスのインスタンスまたはDogクラスのインスタンスを操作できます。

1. メソッドの強制

interfaceを使用することで、クラスに特定のメソッドを実装することを強制できます。たとえば、AnimalIntface interfaceにはeat()とsleep()メソッドが定義されています。CatクラスとDogクラスは、AnimalIntface interfaceを実装するため、それらのメソッドを実装する必要があります。これにより、プログラマが特定のメソッドを実装することを忘れることがなくなり、コードの品質が向上します。

以上のように、interfaceを使用することで、コードの柔軟性と品質を向上させることができます。

※カプセル化の意味合い

CapuslMeanのソースの**Person** クラスがカプセル化の概念を示しています。**name** と **age** のインスタンス変数は **private** で宣言されており、直接アクセスすることはできません。代わりに、**getName()**、**setName()**、**getAge()**、**setAge()** のようなパブリックなメソッドを通じてこれらのプロパティにアクセスします。これにより、**Person** クラスの内部実装の詳細を隠蔽し、外部のクライアントはメソッドを介してデータにアクセスします。