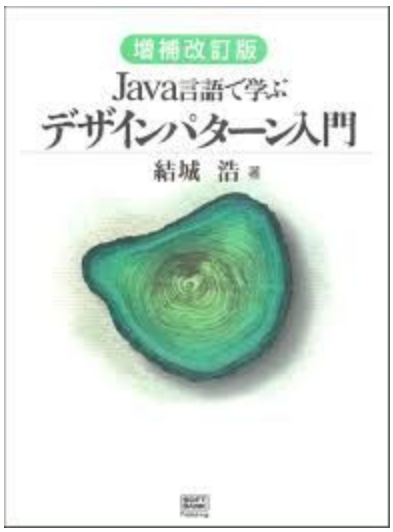
Java言語で学ぶデザインパターン入門

Adapter
一皮かぶせて再利用



Adapterパターン

「すでに提供されているもの」と「必要なもの」の間のずれを埋めるようなデザインパターン。AdapterパターンはWrapperパターンと呼ばれることもある。Adapterパターンには次の２種類がある。

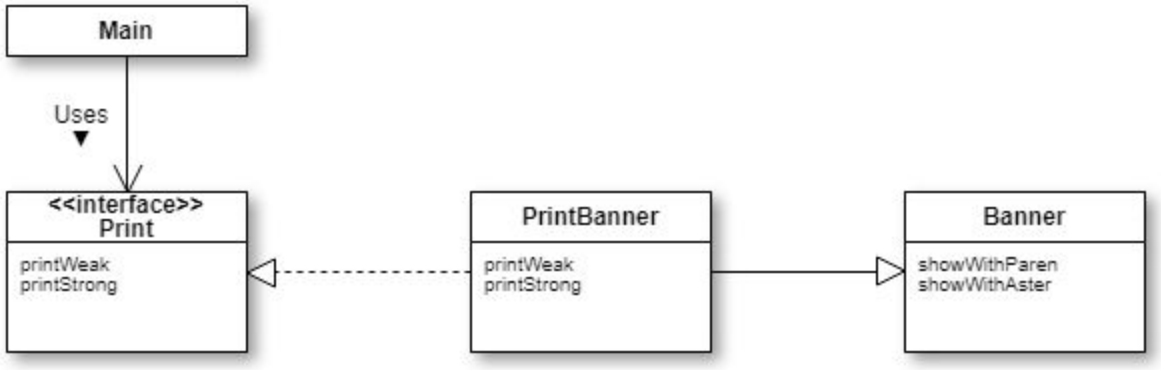
1. クラスによるAdapterパターン（継承をつかったもの）
2. インスタンスによるAdapterパターン（委譲をつかったもの）

サンプル１（継承を使ったもの）

与えられた文字列を（Hello）のように表示したり、*Hello*のように表示したりするサンプルプログラム。Bannerクラスには、文字列をカッコでくくって表示するshowWithParenメソッドと、文字列の前後に*印をつけて表示するshowWithAsterメソッドがある。このBannerクラスを「すでに提供されているもの」とする。一方、Printインタフェースでは、文字列を弱く（カッコつきで）表示するためのメソッドprintWeakと、文字列を強く（*ではさんで強調し

て）表示するためのメソッドprintStrongが宣言されている。このインタフェースが「必要なもの」とする。

クラス図



	サンプルプログラム
提供されているもの	Bannerクラス
変換装置	PrintBannerクラス
必要なもの	Printインタフェース

```
public class Banner {
	private String string;
	public Banner(String string) {
		this.string = string;
	}
	public void showWithParen() {
		System.out.println("(" + string + ")");
	}
	public void showWithAster() {
		System.out.println("*" + string + "*");
	}
}
```

Bannerクラス

```java
public interface Print {
	public abstract void printWeak();
	public abstract void printStrong();
}
```

Printインタフェース

```java
public class PrintBanner extends Banner implements Print{
	public PrintBanner(String string) {
		super(string);
	}
	public void printWeak() {
		showWithParen();
	}
	public void printStrong() {
		showWithAster();
	}
}
```

PrintBannerクラス

```java
public class Main {
	public static void main(String[] args) {
		Print print = new PrintBanner("Hello");
		print.printWeak();
		print.printStrong();
	}
}
```

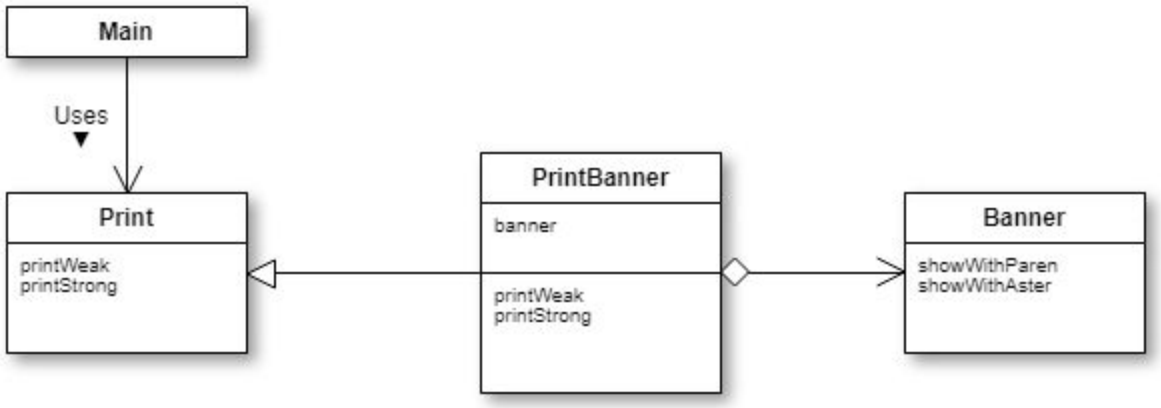
Mainクラス

PrintBannerクラスがどのように実現されているかをMainクラスは知らない。知らないということは、Mainクラスを全く変更せずにPrintBannerクラスの実装を変えられるということ。

サンプルプログラム２（委譲を使ったもの）

Javaでの委譲は、あるメソッドの実際の処理を他のインスタンスのメソッドに任せてしますことをいう。

クラス図




```java
public abstract class Print {
	public abstract void printWeak();
	public abstract void printStrong();
}
```

Printクラス

```java
public class PrintBanner extends Print{
	private Banner banner;
	public PrintBanner(String string) {
		this.banner = new Banner(string);
	}
	public void printWeak() {
		banner.showWithParen();
	}
	public void printStrong() {
		banner.showWithAster();
	}
}
```

PrintBannerクラス