Если вы видите что-то необычное, просто сообщите мне.

Адаптер (Adapter)

Паттерн Adapter относится к структурным паттернам уровня класса.

Часто в новом проекте разработчики хотят повторно использовать уже существующий код. Например, имеющиеся классы могут обладать нужной функциональностью и иметь при этом несовместимые интерфейсы. В таких случаях следует использовать паттерн Adapter.

Смысл работы этого паттерна в том, что если у вас есть класс и его интерфейс не совместим с кодом вашей системы, то что бы разрешить этот конфликт, мы не изменяем код этого класса, а пишем для него адаптер. Другими словами Adapter адаптирует существующий код к требуемому интерфейсу (является переходником).

Требуется для реализации:

1. Интерфейс Target, описывающий целевой интерфейс (тот интерфейс с которым наша система хотела бы работать);
2. Класс Adaptee, который наша система должна адаптировать под себя;
3. Класс Adapter, адаптер реализующий целевой интерфейс.

[!] В описании паттерна применяются общие понятия, такие как Класс, Объект, Абстрактный класс. Применимо к языку Go, это Пользовательский Тип, Значение этого Типа и Интерфейс. Также в языке Go за место общепринятого наследования используется агрегирование и встраивание.

```
//adapter.go
// Package adapter is an example of the Adapter Pattern.
package adapter

// Target provides an interface with which the system should work.
type Target interface {
	Request() string
}
```

```
// Adaptee implements system to be adapted.
type Adaptee struct {
}

// NewAdapter is the Adapter constructor.
func NewAdapter(adaptee *Adaptee) Target {
	return &Adapter{adaptee}
}

// SpecificRequest implementation.
func (a *Adaptee) SpecificRequest() string {
	return "Request"
}

// Adapter implements Target interface and is an adapter.
type Adapter struct {
	*Adaptee
}

// Request is an adaptive method.
func (a *Adapter) Request() string {
	return a.SpecificRequest()
}
```

```
//adapter_test.go
package adapter

import (
	"testing"
)

func TestAdapter(t *testing.T) {

	adapter := NewAdapter(&Adaptee{})

	req := adapter.Request()

	if req != "Request" {
		t.Errorf("Expect volume to %s, but %s", "Request", req)
```

```
}
}
```

Revision #1

Created 2022-07-03 10:27:53 UTC by gasick

Updated 2022-07-03 10:29:02 UTC by gasick