

php design patterns II

<draft>

on command

the command design pattern

Interface
for the
Command
Pattern

```
interface Cmd ICommand {  
    public function execute();  
    // public function undo();  
};
```

the command design pattern

The Class
Macro that
run Commands

```
class Cmd_Macro
{
    private $_commands = array();

    public function add(Cmd_ICommand $c)
    {
        $this->_commands[] = $c;
    }

    public function run()
    {
        // iterator $it
        foreach ($this->_commands as $it) {
            $it->execute();
        }
    }
};
```

the command design pattern

Application Implementations

```
class Hello implements Cmd_ICommand
{
    public function execute()
    {
        echo "Hello ";
    }
};
```

```
class World implements Cmd_ICommand
{
    public function execute()
    {
        echo "World!\n";
    }
};
```

the command design pattern

```
main  
application.  
  
The Macro  
in  
action  
  
$macro = new Cmd_Macro();  
$macro->add(new Hello);  
$macro->add(new World);  
  
$macro->run();  
  
// Ausgabe  
Hello World!
```

the command design pattern

```
// command need to know history
$history = new History();

$cmd = new Command($history);

$cmd->add(new FileCopyCommand);
$cmd->run();
//-----
$cmd->undo();
$cmd->redo();
```

[full source code of the history class](#)

the command design pattern

credits:

"do, undo, redo pattern"

GoF[235],

Thinking In C++ Vol.2 [Chapter 10],

Stefan Schmidt PHP Design Patterns