

# Նախագծման Ձևանմուշներ: Facade

Հրաչյա Թանդիլյան

2020

# Facade

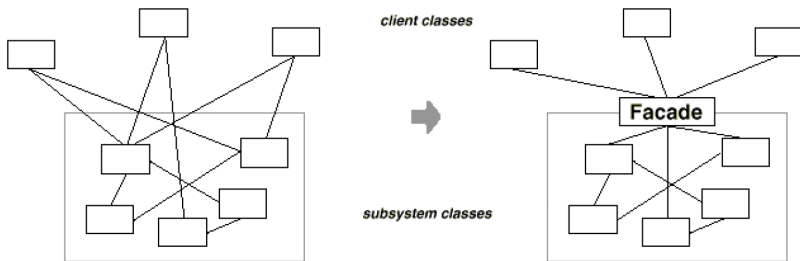
## Նպատակը

Ապահովել ունիֆիկացված ինտերֆեյս  
ենթահամակարգի ինտերֆեյսների բազմության  
համար: Սահմանում է բարձր մակարդակի  
ինտերֆեյս՝ ենթահամակարգը ավելի հեշտ  
օգտագործելու համար:

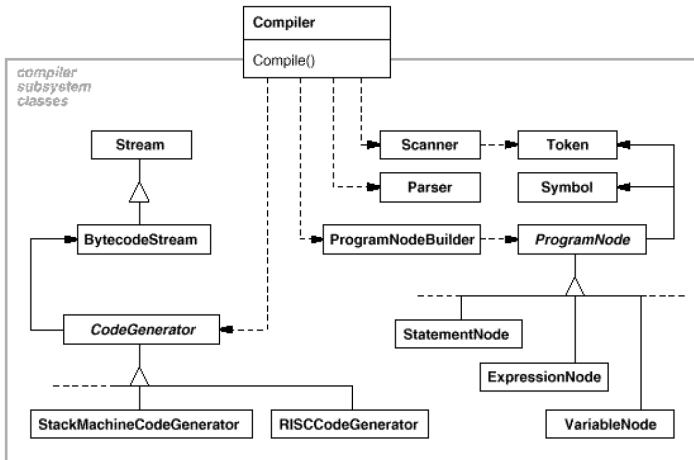
Նաև հայտնի է որպես

- Այլ լայնորեն կիրառվող անուններ չկան:

# Մոտիվացիան



# Մոտիվացիան



# Կիրառելիությունը

Այս Ն.Ձ. պետք է օգտագործել երբ.

- Ցանկանում եք տրամադրել պարզ ինտերֆեյս կոմպլեքս ենթահամակարգի հետ աշխատելու համար:

# Կիրառելիությունը

Այս Ն.Ձ. պետք է օգտագործել երբ.

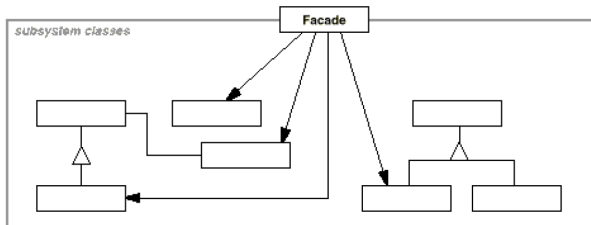
- Ա** Ցանկանում եք տրամադրել պարզ ինտերֆեյս կոմպլեքս ենթահամակարգի հետ աշխատելու համար:
- Բ** Ցանկանում եք նվազեցնել ենթահամակարգի կախվածությունը օգտագործողից և այլ ենթահամակարգերից:

# Կիրառելիությունը

Այս Ն.Ձ. պետք է օգտագործել երբ.

- Ա** Ցանկանում եք տրամադրել պարզ ինտերֆեյս կոմպլեքս ենթահամակարգի հետ աշխատելու համար:
- Բ** Ցանկանում եք նվազեցնել ենթահամակարգի կախվածությունը օգտագործողից և այլ ենթահամակարգերից:
- Գ** Ցանկանում եք շերտավորել ձեր ենթահամակարգերը:

# Կառուցվածքը





# Հետևանքները

Այս Ն.Ձ. ունի հետևյալ առավելություններն ու թերությունները.

- Նվազեցնում է այն օբյեկտների քանակը, որոնց հետ առնչվում է օգտագործողը, դարձնելով ենթահամակարգը հեշտ օգտագործելի:

# Հետևանքները

Այս Ն.Ձ. ունի հետևյալ առավելություններն ու թերությունները.

- Ա** Նվազեցնում է այն օբյեկտների քանակը, որոնց հետ առնչվում է օգտագործողը, դարձնելով ենթահամակարգը հեշտ օգտագործելի:
- Բ** Նվազեցնում է ենթահամակարգի կախվածությունը օգտագործողից և այլ ենթահամակարգերից:

# Հետևանքները

Այս Ն.Ձ. ունի հետևյալ առավելություններն ու թերությունները.

- Ա** Նվազեցնում է այն օբյեկտների քանակը, որոնց հետ առնչվում է օգտագործողը, դարձնելով ենթահամակարգը հեշտ օգտագործելի:
- Բ** Նվազեցնում է ենթահամակարգի կախվածությունը օգտագործողից և այլ ենթահամակարգերից:
- Գ** Չի խոչընդոտում օգտագործողի կողմից ենթահամակարգի դասերի օգտագործմանը (եթե դրա կարիքը կա): Այսինքն օգտագործողը կարող է որոշել ինչն է իրեն անհրաժեշտ ընդհանրությունը, թե պարզությունը:

# Իրականացումը

- Ա Հաճախորդներ-Ենթահամակարգ կապերի կրճատում:
- Բ public ենթահամակարգային դասերը ընդդեմ private ենթահամակարգային դասերի:

# Օրինակ

```
Scanner scanner(input);  
  
ProgramNodeBuilder builder;  
  
Parser parser;  
  
parser.Parse(scanner, builder);  
  
RISCCodeGenerator generator(output);  
  
ProgramNode* parseTree = builder.GetRootNode();  
  
parseTree->Traverse(generator);
```

# Օրինակ

```
class Compiler {  
  
    public:  
        Compiler();  
  
        virtual void Compile(istream& input, BytecodeStream& output) {  
  
            Scanner scanner(input);  
            ProgramNodeBuilder builder;  
            Parser parser;  
            parser.Parse(scanner, builder);  
            RISCCodeGenerator generator(output);  
            ProgramNode* parseTree = builder.GetRootNode();  
            parseTree->Traverse(generator);  
        }  
};
```

# Առնչվող Նախագծման Ձևանմուշները

- Abstract Factory

- Mediator

- Singleton