

**Зохиомжийн загварууд 1**

# **Програм хангамжийн архитектур (Software Architecture)**

**2017**

# Агуулга

- Нэр томъёо
- Зохиомжийн загвар гэж юу вэ?
- Зохиомжийн загварын ангилал
- Үүсгэгч загварууд
- Бүтцийн загварууд
- Харьцааны загварууд
- Abstract factory pattern
- Builder pattern
- Singleton pattern
- Дүгнэлт

# Нэр томьёо

- Design pattern – Зохиомжийн загвар
- Creational pattern – Үүсгэгч загварууд
- Structural pattern – Бүтцийн загварууд
- Behavioral pattern – Харьцааны загварууд

# Зохиомжийн загвар гэж юу вэ?

- Загвар (Pattern) нь тодорхой кодын хэсэгт давтагддаг тодорхой хэлбэрийн хийсвэрлэл юм.

“Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.”

Each pattern is a three part rule, which expresses a relation between a certain context, a problem, and a solution.

*Christopher Alexander*

- ОХ загварт түгээмэл давтагддаг тодорхой асуудлын шилдэг шийдлийн хийсвэрлэлийг тодорхойлдог.

# Зохиомжийн загвар гэж юу вэ?

Зохиомжийн загвар дараах хэсгүүдээс тогтдог.

- Зохиомжийн загварын нэр
- Тухайн зохиомжийн загвараар шийдэгдэх ёстой асуудал
- Асуудлыг агуулж буй хам бичвэр (context) – Асуудал гарч буй газар
- Шийдэл – Батлагдсан шийдэл
- Эерэг сөрөг үр дагаварууд

# GoF Зохиомжийн загварын ангилал

Зорилго – хийдэг үйлдлээр нь

- Үүсгэгч загвар (creational pattern)
  - Объект үүсгэхтэй холбоотой
- Бүтцийн загвар (structural pattern)
  - Класс ба объектын нэгдэлтэй холбоотой
- Харьцааны загвар (behavioral pattern)
  - Класс ба объектын харьцаатай холбоотой

Хүрээ – хэрэглээгээр нь

- Класс загварууд
  - Класс ба тэдгээрийн дэд классуудын холбоо
  - Удамшил ба дахин хэрэглэгдэх боломж
- Объект загварууд
  - Объектуудын холбоо

# Үүсгэгч загварууд

- Abstract factory (хийсвэр үйлдвэр)
- Builder (бүтээгч)
- Factory method (үйлдвэрийн арга)
- Singleton
- Object pool
- ...

# Харьцааны загвар

- Command
- Command processor
- Composite
- Iterator
- Visitor
- Strategy
- Template method
- Observer
- ...



# Бүтцийн загварууд

- Adapter
- Bridge
- Decorator
- Facade
- Proxy
- Model View Controller
- Flyweight
- ...

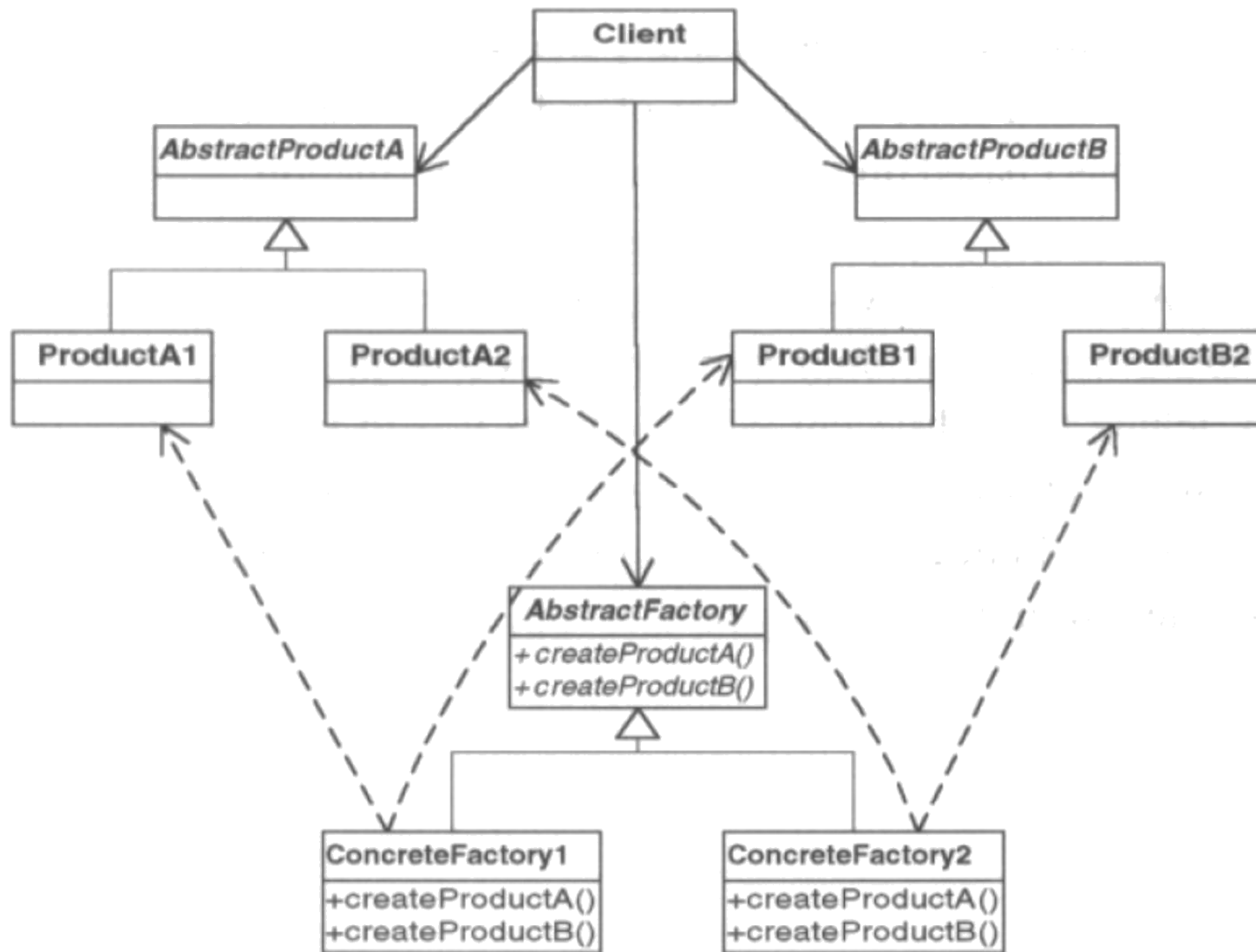
## Яагаад зохиомжийн загвар гэж?

- Асуудлыг шийдэх шаблом болж өгдөг.
- Зохиомжийн загвар нь дахин хэрэглэгдэх боломж, холболт хийх боломж олгоход нэн тустай.
- Шинжилгээ ба загвар гаргахад дээд төвшнөөс харах боломж олгоно.
- Багийн холбоог сайжруулж, хувийн сурах боломжийг нэмэгдүүлнэ.
- Кодын өөрчлөгдөх боломжийг сайжруулна.
- ОХП-н суурь зарчмуудын ойлголтыг нарийвчилж өгнө. (encapsulation, inheritance, polymorphism)
- Архитектурын зарчмуудыг хэрэгжүүлэх боломж олгоно.

# Abstract factory

- Конкрет классыг тодорхойлохгүйгээр хоорондоо холбоотой эсвэл хамааралтай объектуудын бүл үүсгэх зурвас бэлтгэнэ.
- Мөн Kit, Toolkit гэж ч нэрлэдэг. Хэзээ хэрэглэх вэ?
- Систем тодорхой объект яаж үүсгэхээс хамааралгүй байх ёстой бол
- Систем янз бүрийн „объектын бүл“ -тэй тохиргоо хийгдсэн бол
- Янз бүрийн объектыг үргэлж цугт нь үүсгэх ёстой бол
- Хоорондоо хамааралтай объектын бүл конкрет хэрэгжүүлэлтгүйгээр харин зөвхөн зурвасаар үүсгэх бол

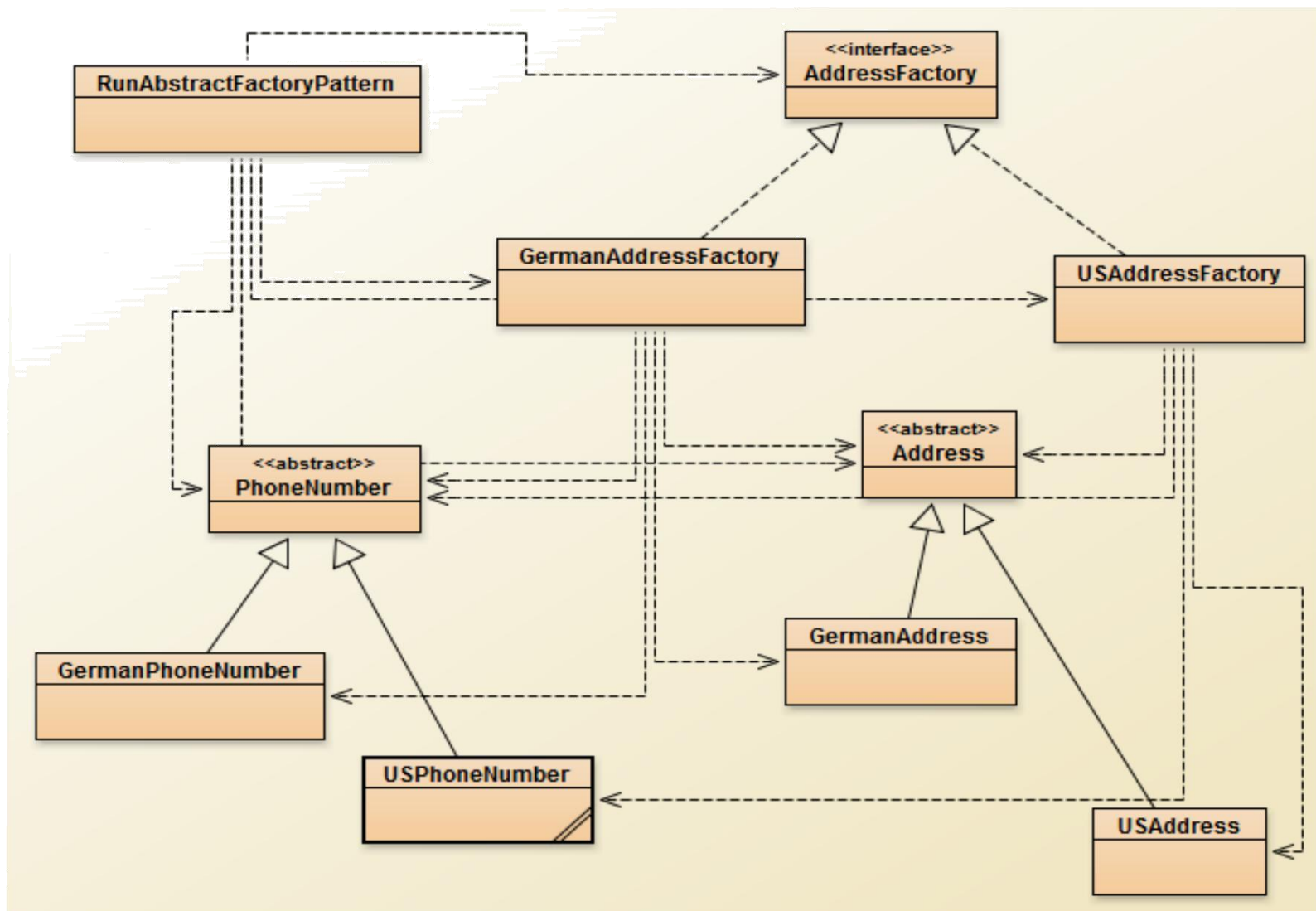
# Abstract factory



# Abstract factory

- Client 2 өөр сервер классыг хэрэглэж байна (AbstractProductA and AbstractProductB). Энэ нь хэрэгжүүлэлтийг далдалж, хялбарчилж системийг арчлахад хялбар болгож байна.
  - Client объект тухайн сервер объектын конкрет хэрэгжүүлэлтийг мэдэхгүй. Учир нь factory объект тэдгээрийг үүсгэх үүрэгтэй.
  - Client объект ямар тодорхой factory ашиглаж байгаагаа ч мэдэхгүй, зөвхөн Abstract Factory объект хэрэглэж байгаагаа л мэднэ. Түүнд ConcreteFactory1 эсвэл ConcreteFactory2 объект бий боловч аль нь ашиглагдаж буйг мэдэхгүй.

# Abstract factory - Жишээ

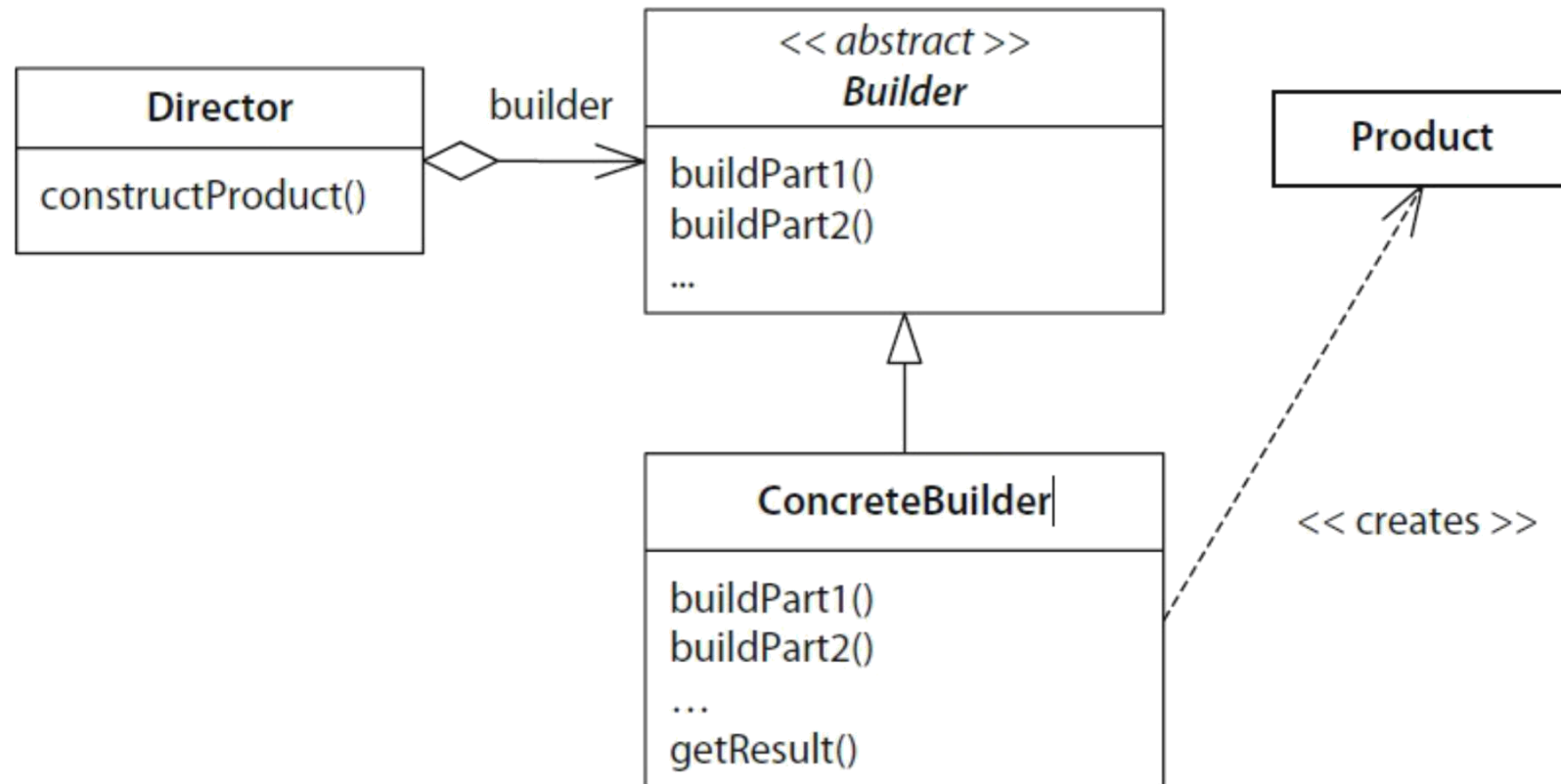


BlueJ дээр дээрх жишээг ажиллуулж харуулна.

# Builder

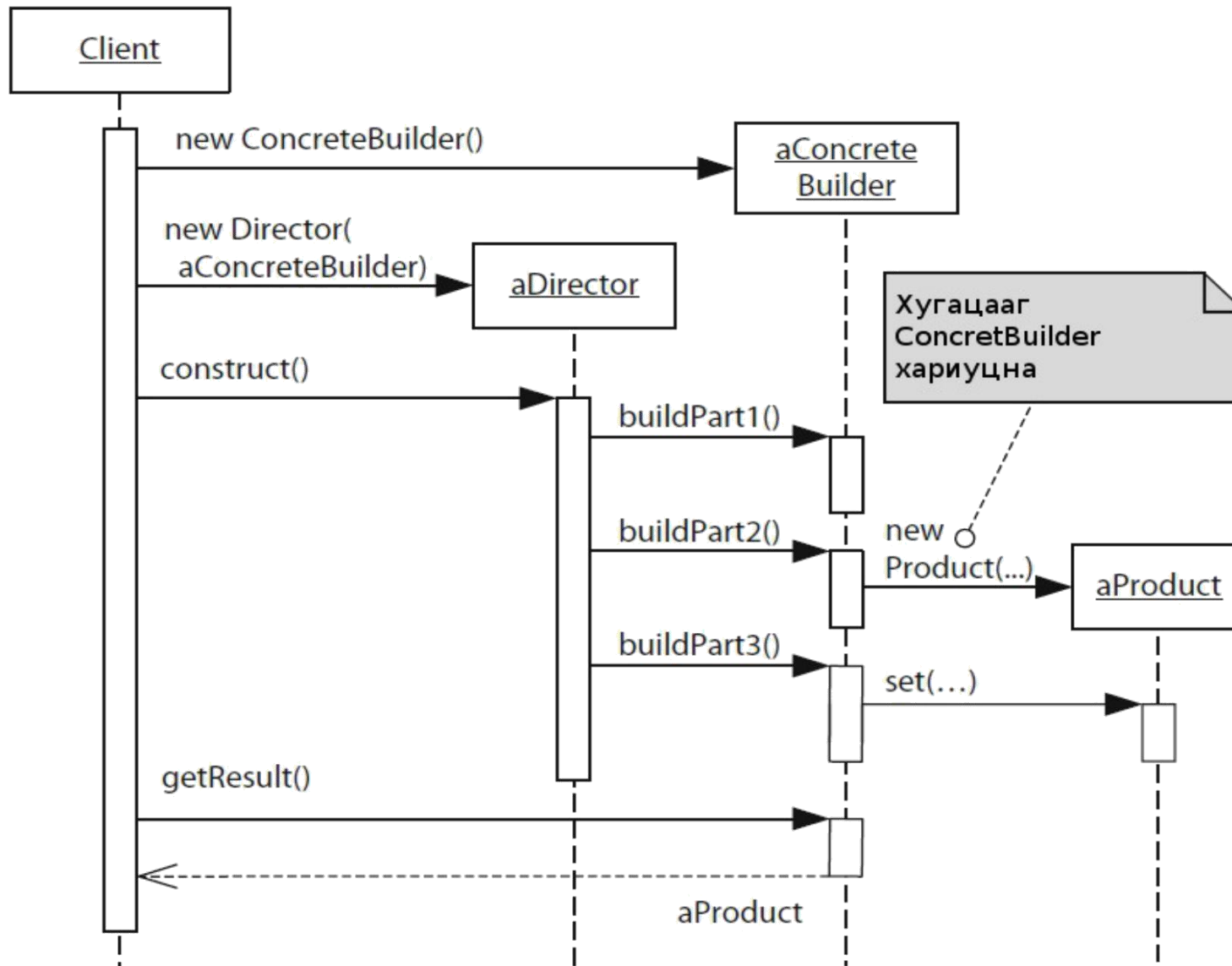
- Угсрах процессыг тусгай класст байршуулж төвөгтэй объект үүсгэхийг хялбарчлах.  
Хэзээ хэрэглэх вэ?
- Дамжлага бүхий угсралттай төвөгтэй объект үүсгэх бол
- Угсрах процесс янз бүрийн илэрхийлэлтэй байх хэрэгтэй бол
- BlueJ дээр жишээ авч ажиллуулж үзүүлье.

# Builder



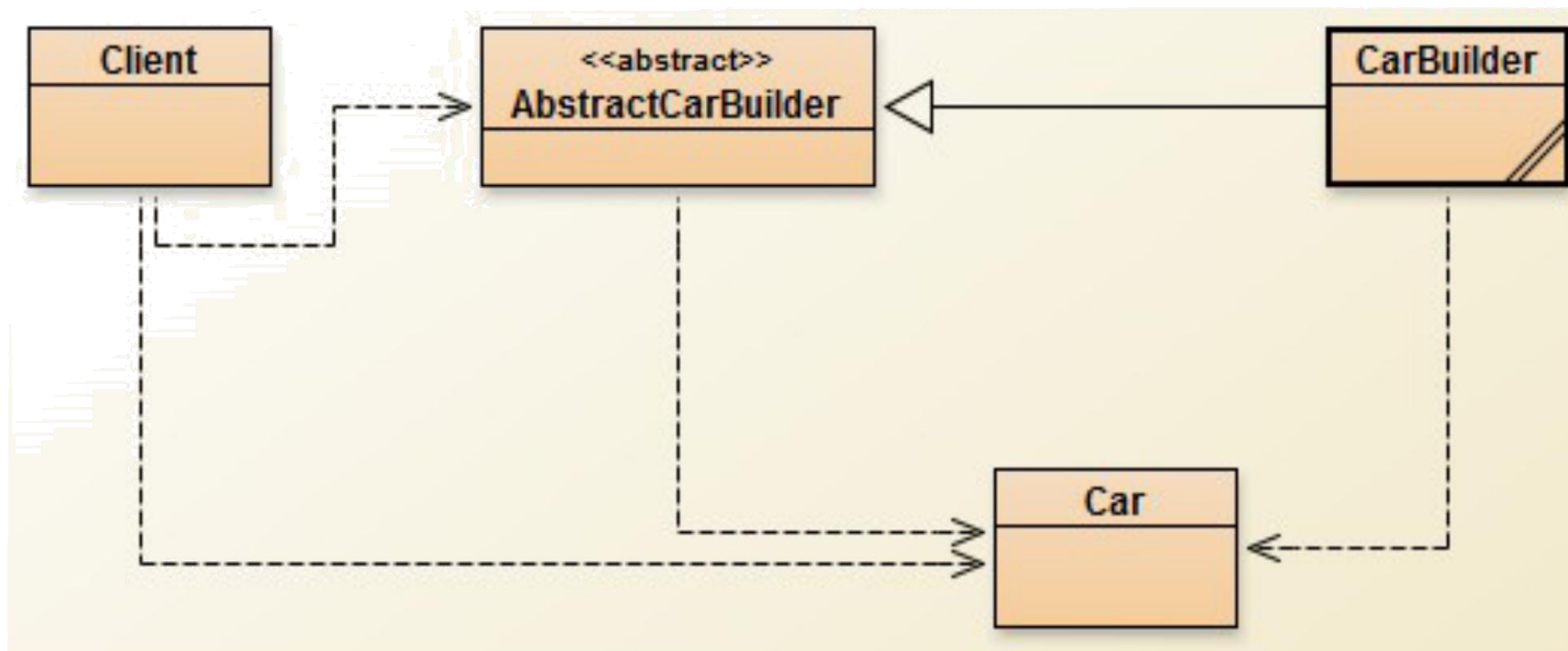


# Builder



# Builder

- Дараах жишээг хамтран хийж, ажиллуулж үзүүлэв.

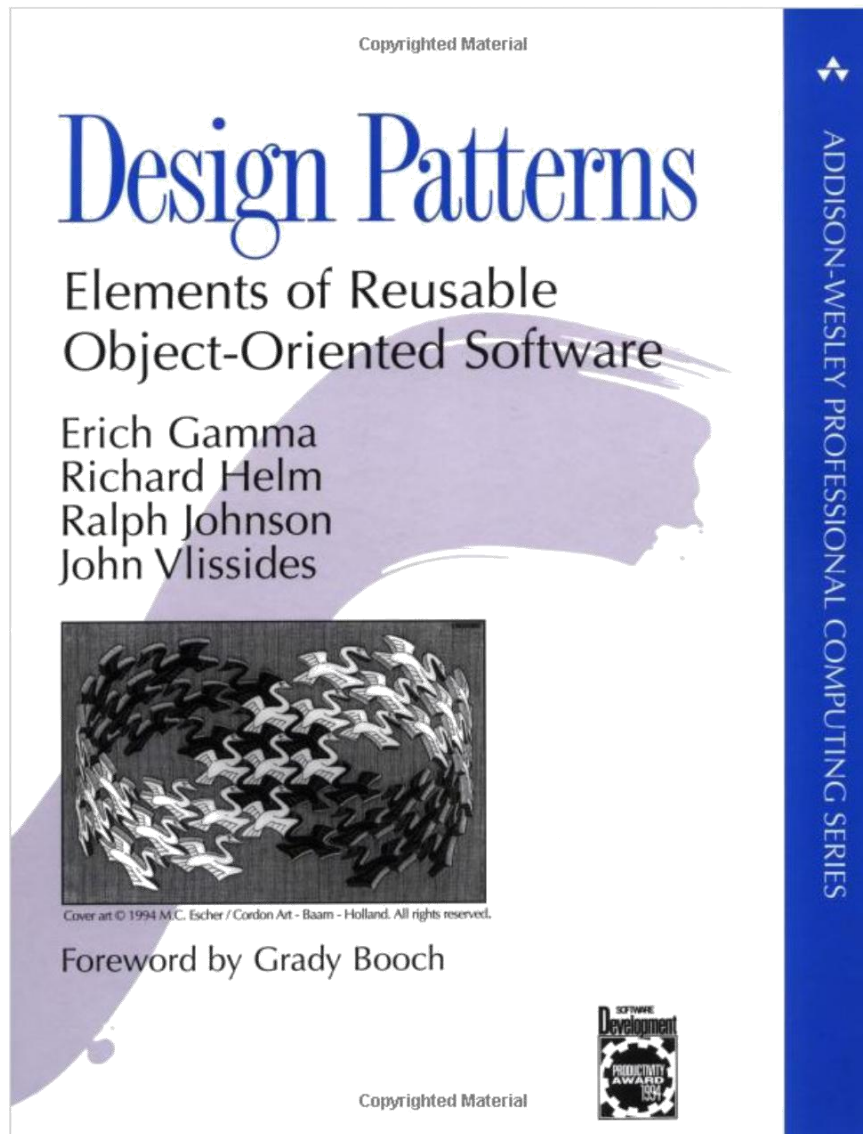


# Singleton

- Классаас зөвхөн нэг инстанц үүсгэх

```
public final class Singleton {  
    private static Singleton theInstance =  
        null; private Singleton() {};  
    public static Singleton getInstance() {  
        if (theInstance == null) theInstance = new  
            Singleton(); return theInstance;  
    }  
    //Prevent clone method  
}
```

# Судлах материал



## Design Patterns: Elements of Reusable Object-Oriented Software

**Erich Gamma,  
Richard Helm,  
Ralph Johnson,  
John Vlissides**

## Дүгнэлт

- Зохиомжийн загвар гэж юу болох
- Хэрхэн ангилагддаг
- Яагаад судлах хэрэгтэй гэж
- Үүсгэгч загварууд – Abstract factory pattern, builder pattern, singleton pattern