

# VISITOR/パターン

Seeeeeee:D 夏休み勉強会  
Ishijima Nanaka

# Visitorパターンとは



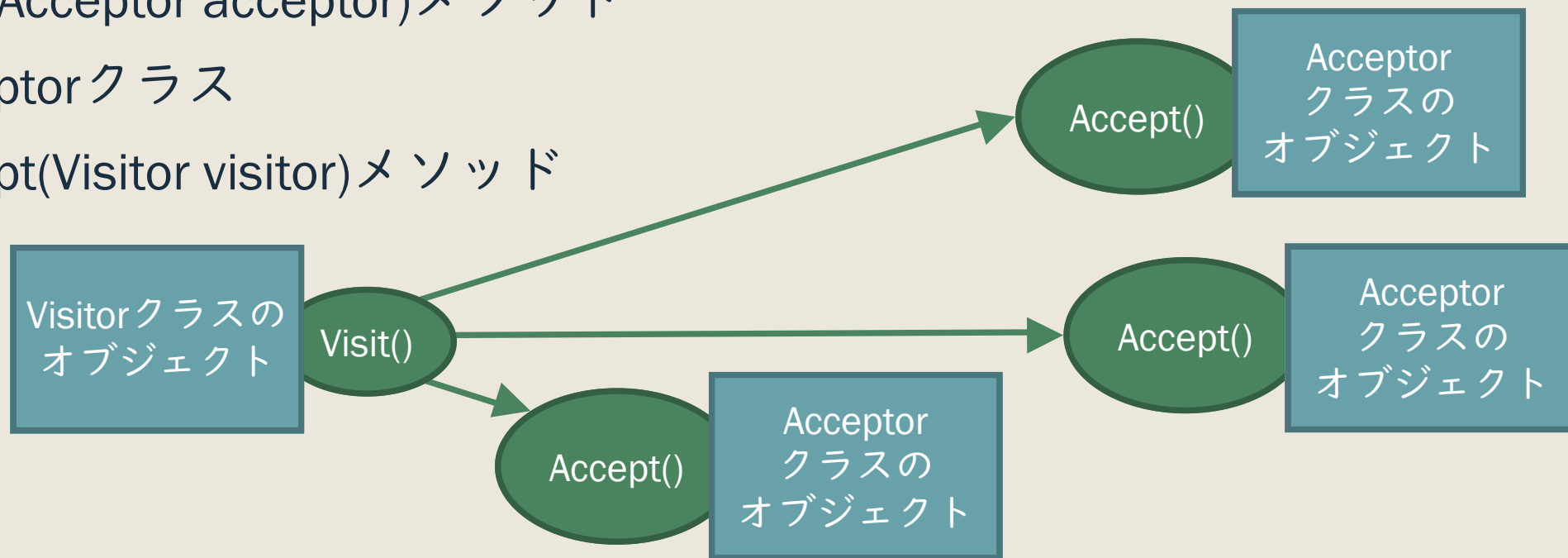
## Visitor = 訪問者



# Visitorパターンとは

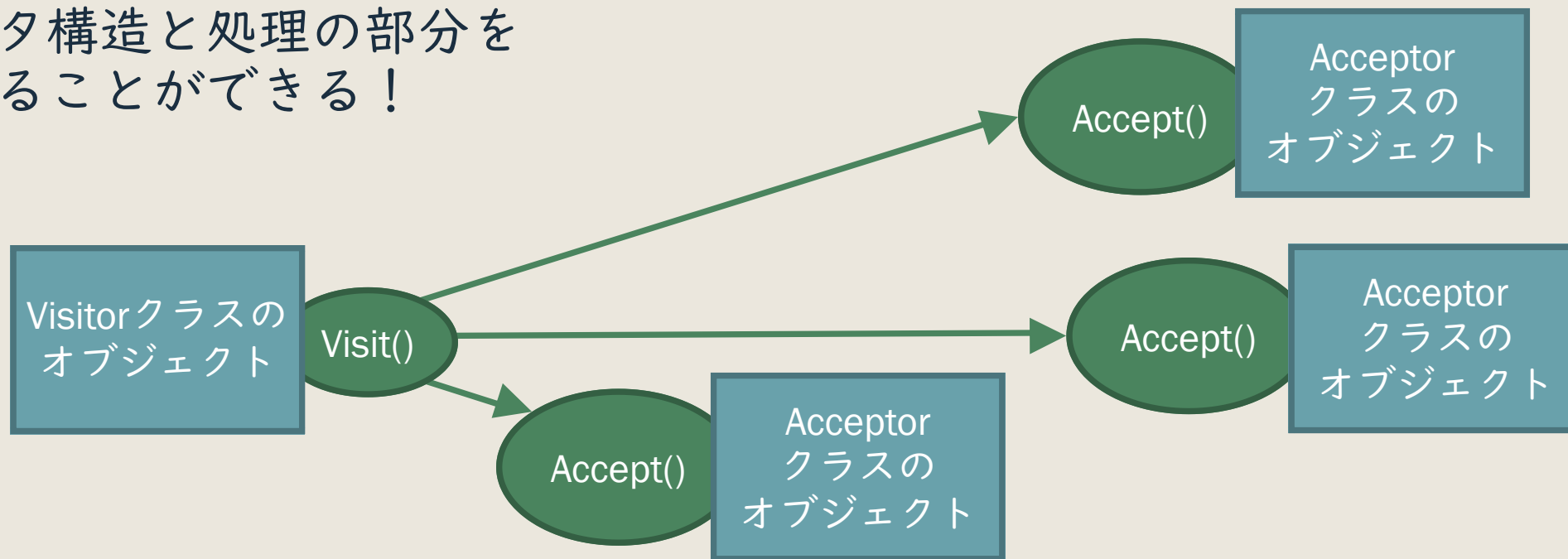
おもに以下の4つの要素で構成される

- Visitorクラス
- Visit(Acceptor acceptor)メソッド
- Acceptorクラス
- Accept(Visitor visitor)メソッド



# Visitorパターンとは

- Visitorクラスのオブジェクトがどれだけ増えても、受け入れ側はacceptメソッドさえあれば新しいオブジェクトを受け入れることができる
- データ構造と処理の部分を分けることができる！



# 実装方針

- 観光地に旅行に行く例
- 観光客は、シルバーツアーのお年寄りから、修学旅行の学生まで色々
- そこで、観光地を Acceptor クラス、観光客を Visitor クラスとして作っていく

# 実装方針

- インターフェース :
  - IVisitor
- Visitorクラス :
  - Silverクラス, Studentクラス
- 抽象クラス :
  - Siteクラス 抽象メソッドとしてAccept(Ivisitor visit)を持つ
- Acceptorクラス :
  - Tokyoクラス, Kyotoクラス, Hakataクラス, Hokkaidoクラス

(そういえばacceptorクラスの親として抽象クラスでSiteクラスを作ったけど、IAcceptableインターフェースのほうが良かった説)

# インターフェース：IVisitor

```
interface IVisitor
```

```
{
```

```
    void visit(Tokyo tokyo);
```

```
    void visit(Kyoto kyoto);
```

```
    void visit(Hakata hakata);
```

```
    void visit(Hokkaido hokkaido);
```

```
}
```

- Acceptorクラスの数だけ Visitメソッドを作る

# Visitorクラス：Silver, Student

```
class Silver : IVisitor
```

```
{  
    public void visit(Tokyo tokyo)  
    {  
        Console.WriteLine("東京じゃあ～！浅  
草寺が趣深いのお～");  
    }  
    public void visit(Kyoto kyoto)  
    {  
        Console.WriteLine("京都じゃよ！八つ  
橋がのどに詰まるんじゃあ...");  
    }  
}
```

```
class Student : IVisitor
```

```
{  
    public void visit(Tokyo tokyo)  
    {  
        Console.WriteLine("東京だ！人がたくさ  
ん！");  
    }  
    public void visit(Kyoto kyoto)  
    {  
        Console.WriteLine("京都に来たよ！舞妓  
さんがいとをかし～");  
    }  
}
```



# 抽象クラス：Site

```
abstract class Site
{
    public abstract void Accept(IVisitor visitor);
}
```

- Acceptorクラスが必ず Acceptメソッドを持つようにする

# Acceptorクラス： Tokyo, Kyoto, Hokkaido, Hakata

```
class Tokyo : Site
{
    public override void Accept(IVisitor visitor)
    {
        visitor.visit(this);
        Console.WriteLine("僕はハチ公！東京によろこそだワン！");
    }
}
```

# Acceptorクラス： Tokyo, Kyoto, Hokkaido, Hakata

```
class Hakata : Site
{
    public override void Accept(IVisitor visitor)
    {
        visitor.visit(this);
        Console.WriteLine("博多ば、よおきんしゃったね！");
    }
}
```

# VISUAL STUDIO^

実行結果を見てみよ〜〜

# 感想とまとめ

- 今週忙しかったので、割りとイメージのつかみやすいパターンで助かった
- でも、分かりやすいんだけど、どんな時に有効なパターンかいまいち分かってないや
- 個人的にはインターフェースを使う練習としてすごく勉強になりました