

# 17-Observer

Seeeeeee:D夏休み勉強会

デザインパターン

Katsuya Suzuki

# Observerパターンとは

- 複数あるオブジェクトのうちの1つに状態の変更があったときに、ほか全てのオブジェクトにその旨を通知するための方法(を実現するデザインパターン)

# プログラム例

- 鬼ごっこするプログラムを考える
- 誰かが鬼に捕まったら，自分が捕まったことをほか全員に知らせたい
- どうやって知らせる？

パターン適用前

# 鬼に捕まったら

- 通知相手(まだ捕まっていない人)を捕まっていない人が判断する必要がある
- (そもそも捕まってる人が知らせるのはどうなの...)

```
// Programs.cs
using System;

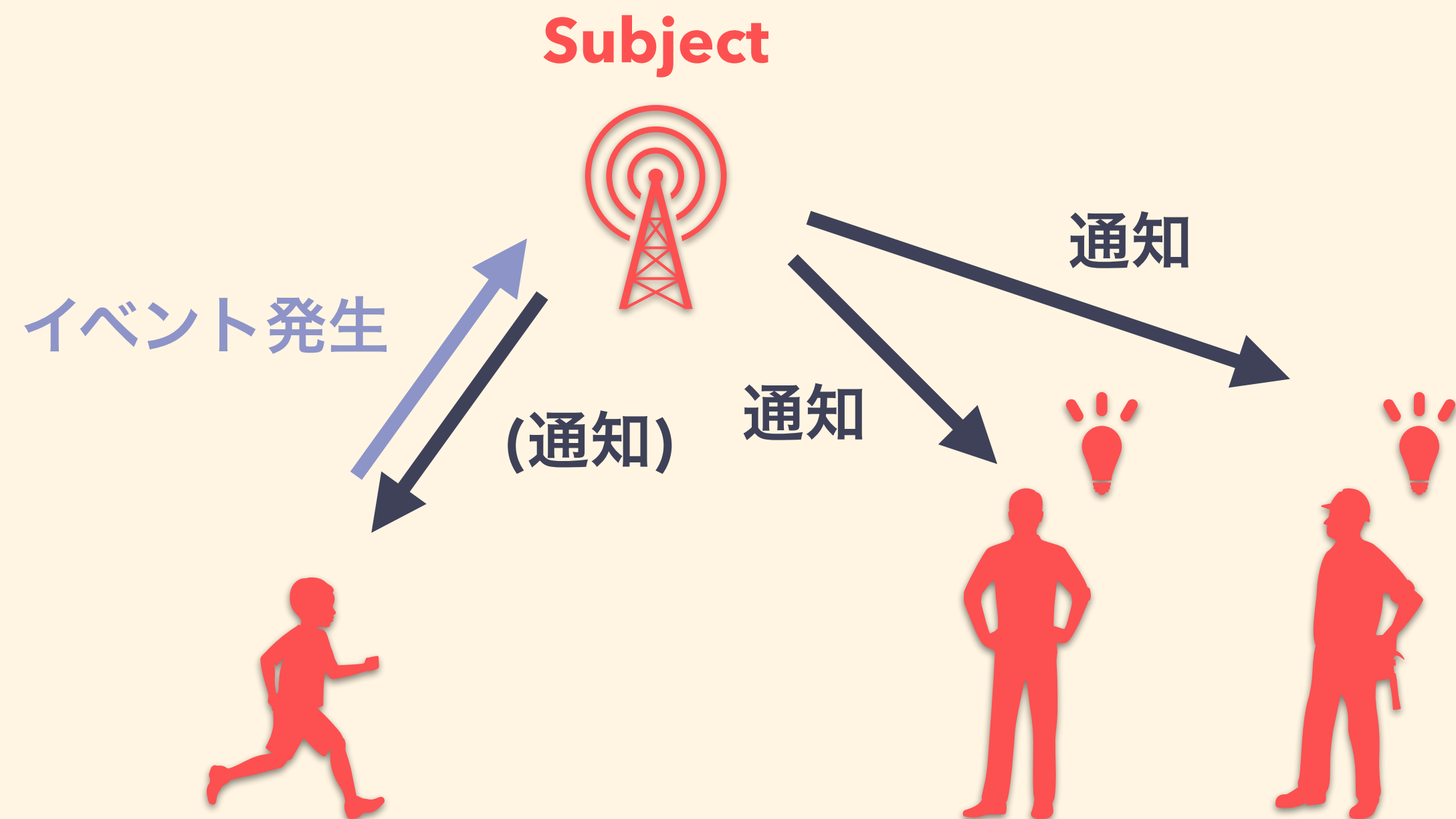
namespace Tosochu
{
    class Program
    {
        static void Main(string[] args)
        {
            Runner sato = new Runner("佐藤");
            Runner suzuki = new Runner("鈴木");
            Runner tanaka = new Runner("田中");

            // 田中さんが捕まったら...
            tanaka.NotifyTo(sato);
            tanaka.NotifyTo(suzuki);

            // 佐藤さんが捕まったら...
            sato.NotifyTo(suzuki);
        }
    }
}
```

# どうしたら上手く通知できるか

- 全体に通知するオブジェクト(Subject)を設ける
- 全てのインスタンスがSubjectにアクセスできるようにする
- Subjectを通して他のインスタンスにイベントの発生を通知することができる



# パターン適用後

- VSCodeで確認します

# まとめ

- 「観察」というより「通知」
- 関連するオブジェクト全部とアクセスできるオブジェクト (Subject)を通して通知する