

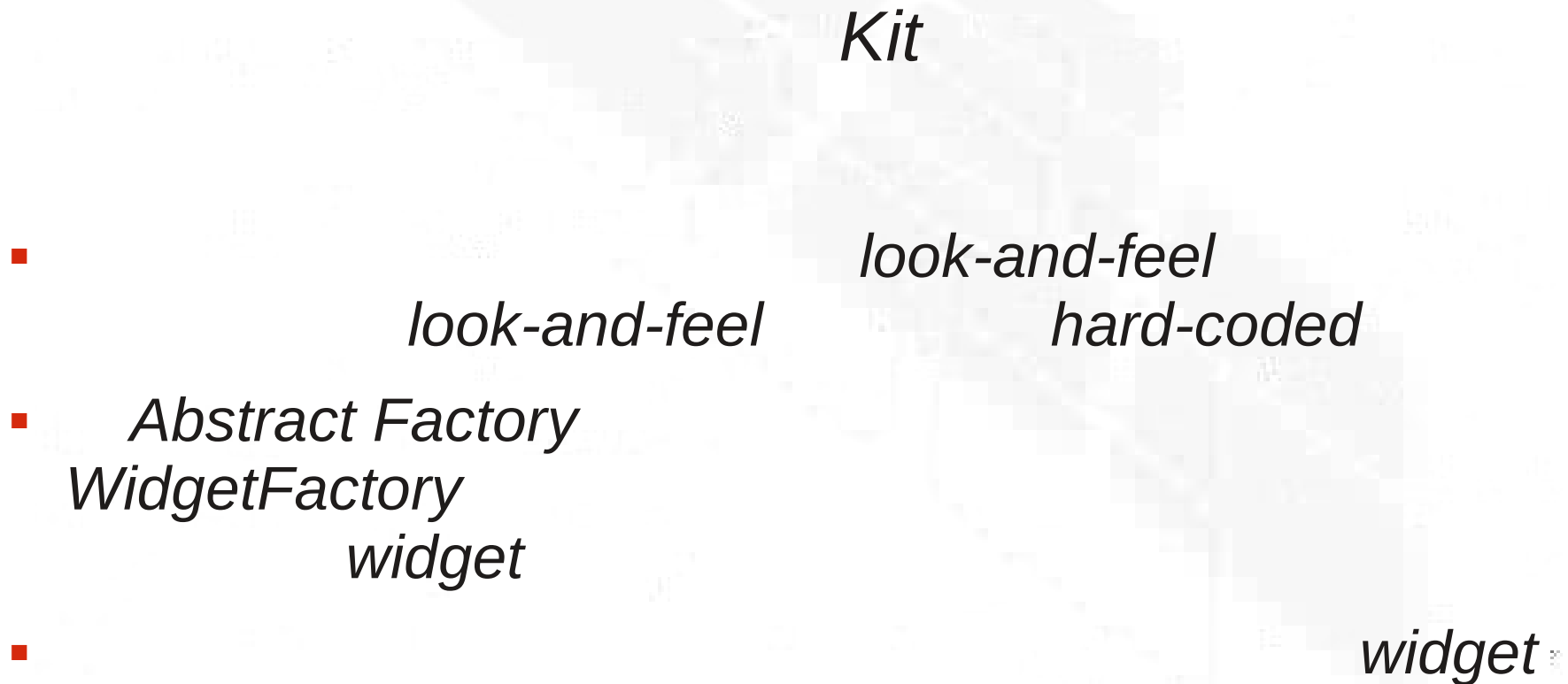
Abstract Factory

Sandro Santos Andrade

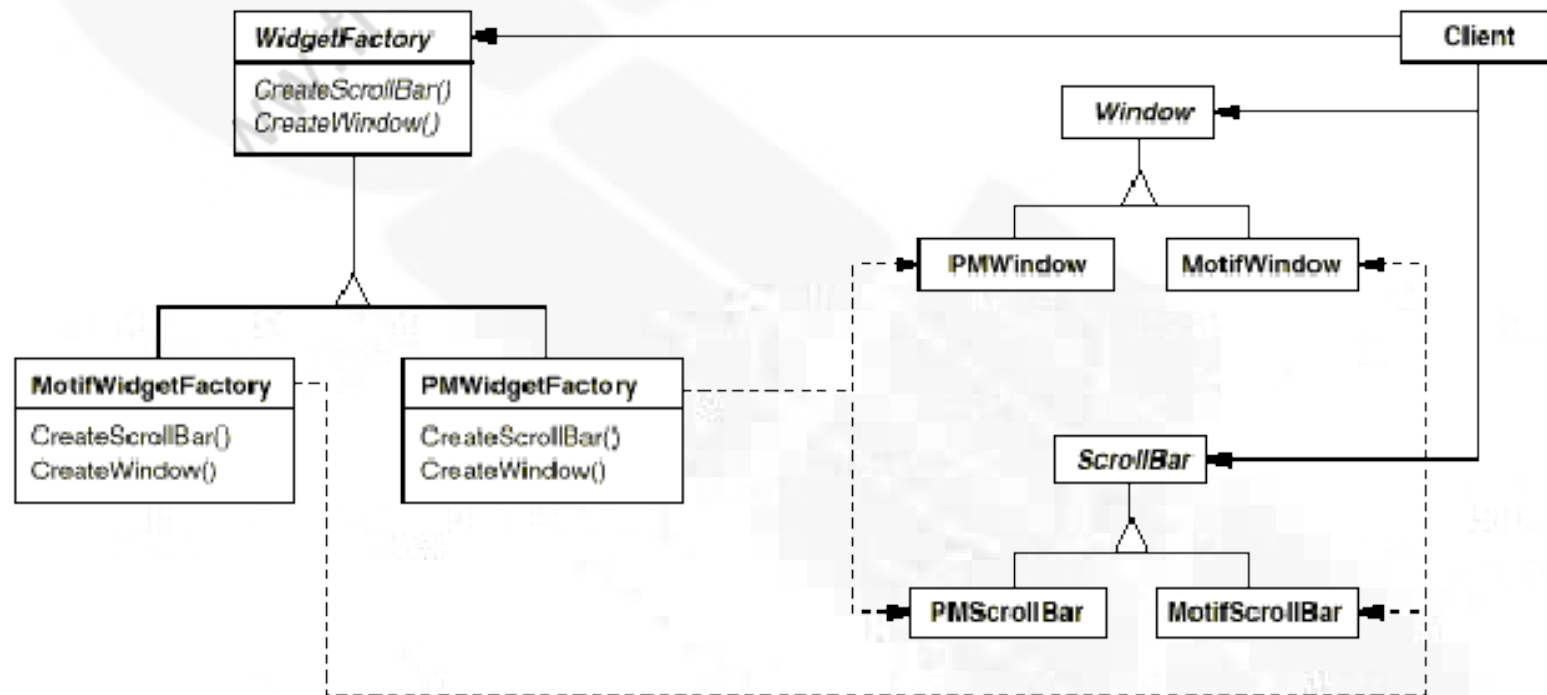
**Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Departamento de Tecnologia Eletro-Eletrônica
Graduação Tecnológica em Análise e Desenvolvimento de Sistemas**



Abstract Factory



Abstract Factory

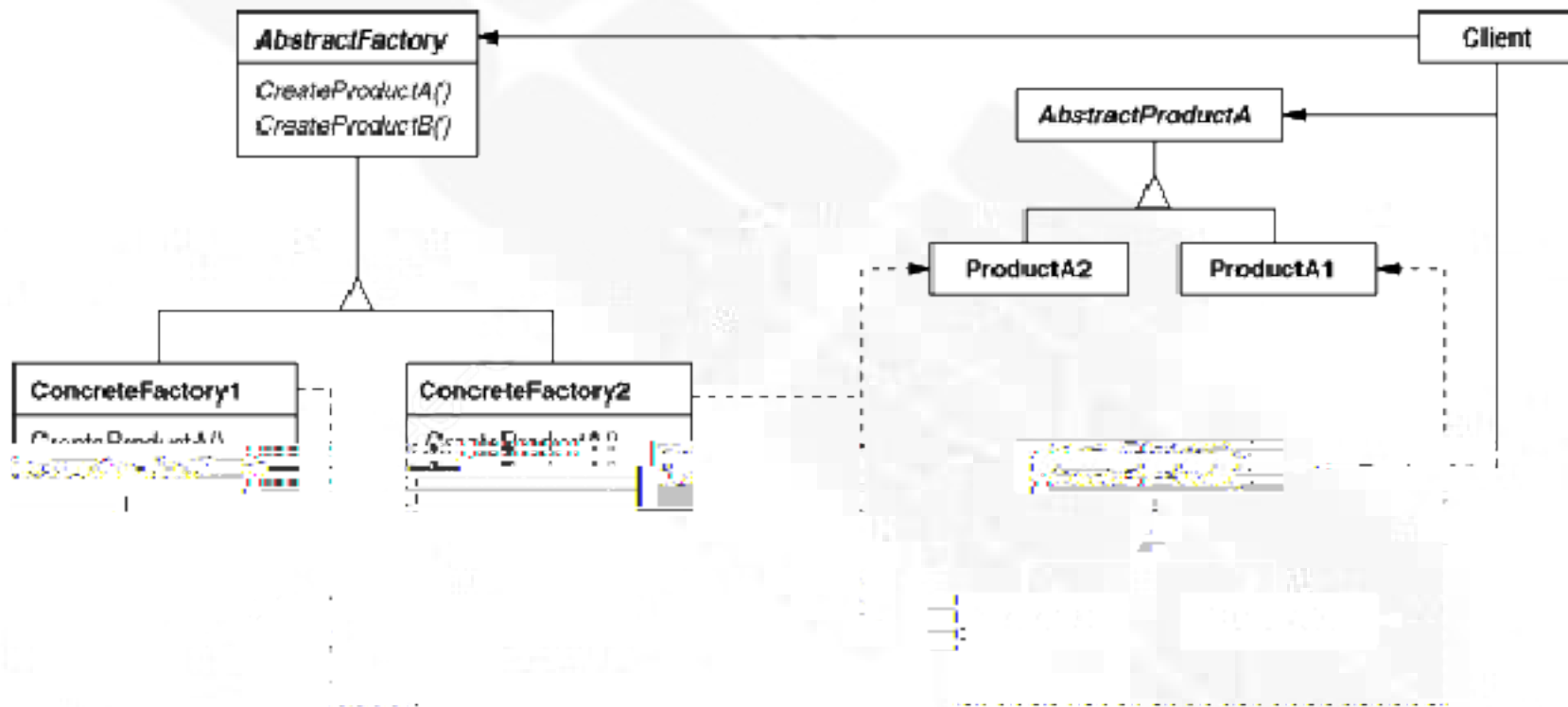


*look-and-feel
widgets*

Abstract Factory



Abstract Factory



Abstract Factory



- *AbstractFactory*
 - *ConcreteFactory*
 - *AbstractProduct*
 - *ConcreteProduct*
- AbstractProduct*

Abstract Factory



run-time

Abstract Factory



Abstract Factory



Abstract Factory

Singletons

interface
Factory

Method

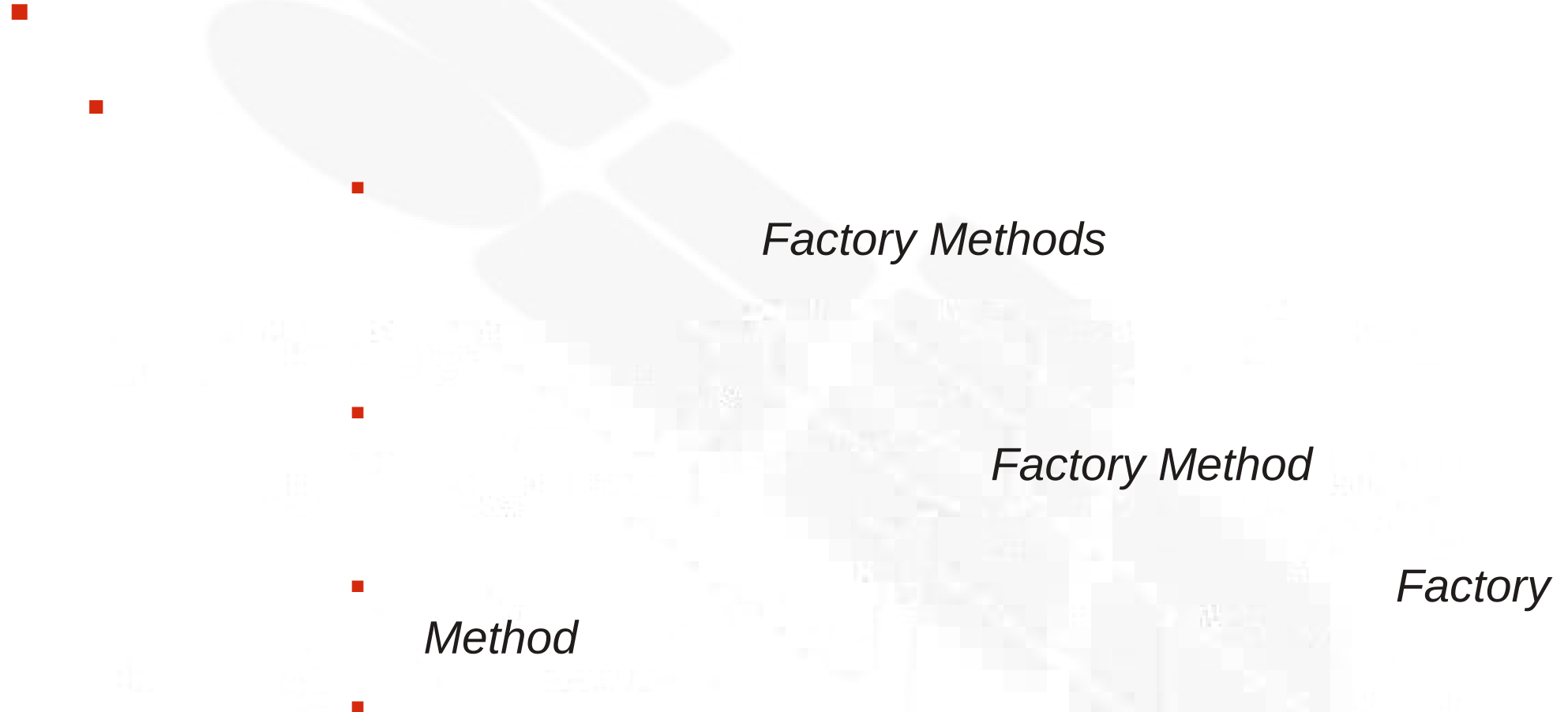
Factory

Methods

Abstract Factory

Prototype

Abstract Factory



Abstract Factory

```
class MazeFactory {
public:
    MazeFactory();

    virtual Maze* MakeMaze() const
        { return new Maze; }
    virtual Wall* MakeWall() const
        { return new Wall; }
    virtual Room* MakeRoom(int n) const
        { return new Room(n); }
    virtual Door* MakeDoor(Room* r1, Room* r2) const
        { return new Door(r1, r2); }
};
```

Abstract Factory

```
Maze* MazeGame::CreateMaze (MazeFactory& factory) {
    Maze* aMaze = factory.MakeMaze();
    Room* r1 = factory.MakeRoom(1);
    Room* r2 = factory.MakeRoom(2);
    Door* aDoor = factory.MakeDoor(r1, r2);

    aMaze->AddRoom(r1);
    aMaze->AddRoom(r2);
    r1->SetSide(North, factory.MakeWall());
    r1->SetSide(East, aDoor);
    r1->SetSide(South, factory.MakeWall());
    r1->SetSide(West, factory.MakeWall());

    r2->SetSide(North, factory.MakeWall());
    r2->SetSide(East, factory.MakeWall());
    r2->SetSide(South, factory.MakeWall());
    r2->SetSide(West, factory.MakeWall());

    return aMaze;
}
```

Abstract Factory

```
class EnchantedMazeFactory : public MazeFactory {
public:
    EnchantedMazeFactory();

    virtual Room* MakeRoom(int n) const
        { return new EnchantedRoom(n, CastSpell()); }

    virtual Door* MakeDoor(Room* r1, Room* r2) const
        { return new DoorNeedingSpell(r1, r2); }

protected:
    Spell* CastSpell() const;
};
```

Abstract Factory

```
Wall* BombedMazeFactory::MakeWall() const {  
    return new WallWithABomb();  
}  
  
Room* BombedMazeFactory::MakeRoom(int n) const {  
    return new RoomWithABomb(n);  
}
```

```
MazeGame game;  
BombedMazeFactory factory;  
  
game.CreateMaze(factory);
```


Abstract Factory



Methods

Factory



BombedMazeFactory

RoomWithABomb

BombedWall



RoomWithABomb

BombedWall

downcasting



downcast

Abstract Factory

- - *toolkit*

- *Abstract Factory*

Abstract Factory

- - *Abstract Factory*
Factory Methods

Prototype
Singleton

Abstract Factory

Sandro Santos Andrade

**Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Departamento de Tecnologia Eletro-Eletrônica
Graduação Tecnológica em Análise e Desenvolvimento de Sistemas**

