

***P00***

Sabloane  
Adapter

# Cuprins

---

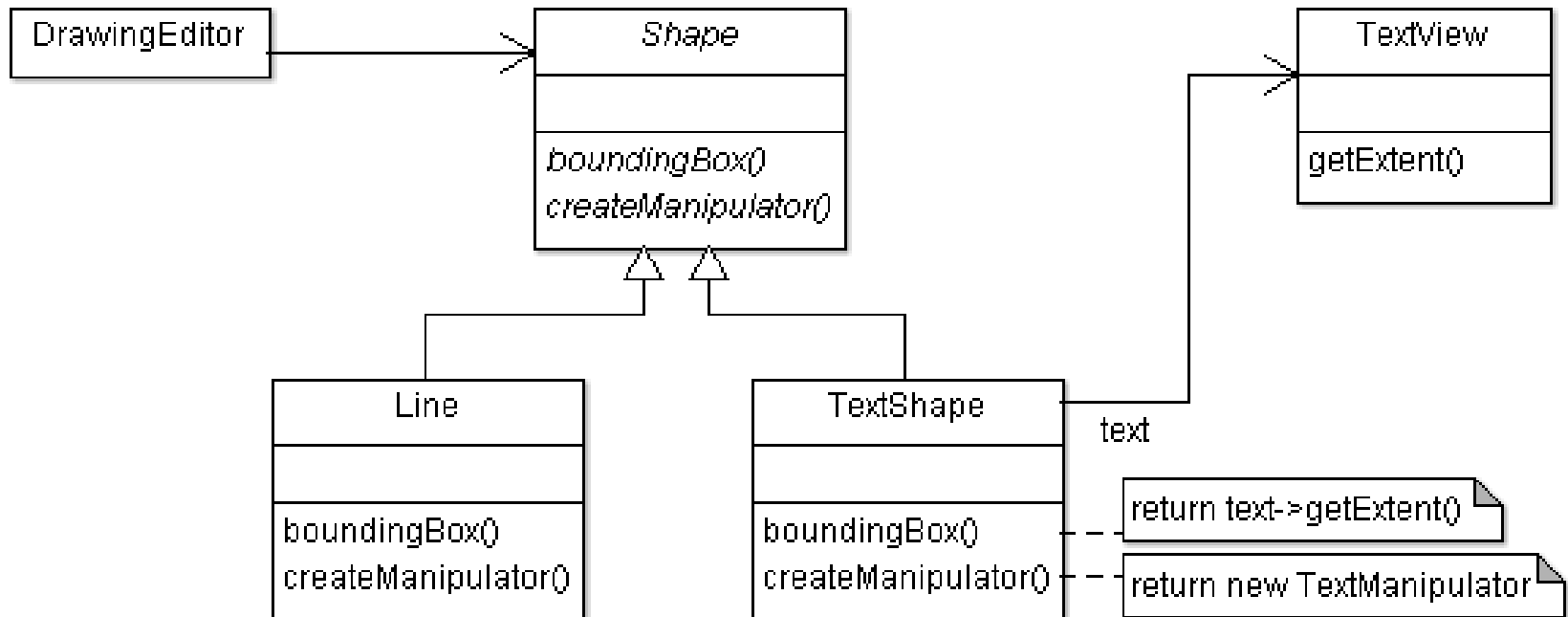
- sablonul Adapter (prezentare dupa GoF)
- aplicatie (not yet)

# Intentie

---

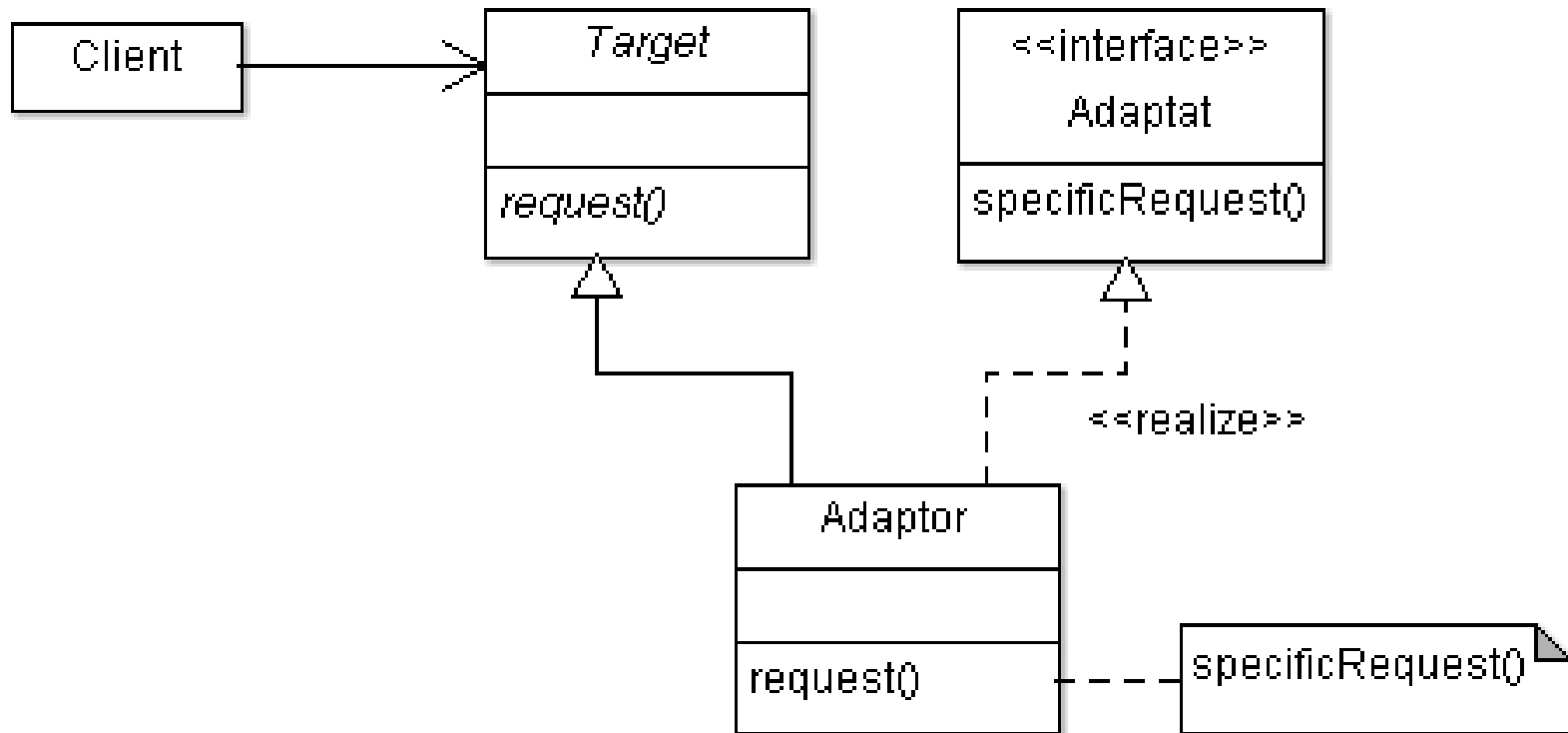
- convertește o interfata a unei clase într-o alta interfata
- cunoscut și sub numele de *Wrapper*

# Motivatie

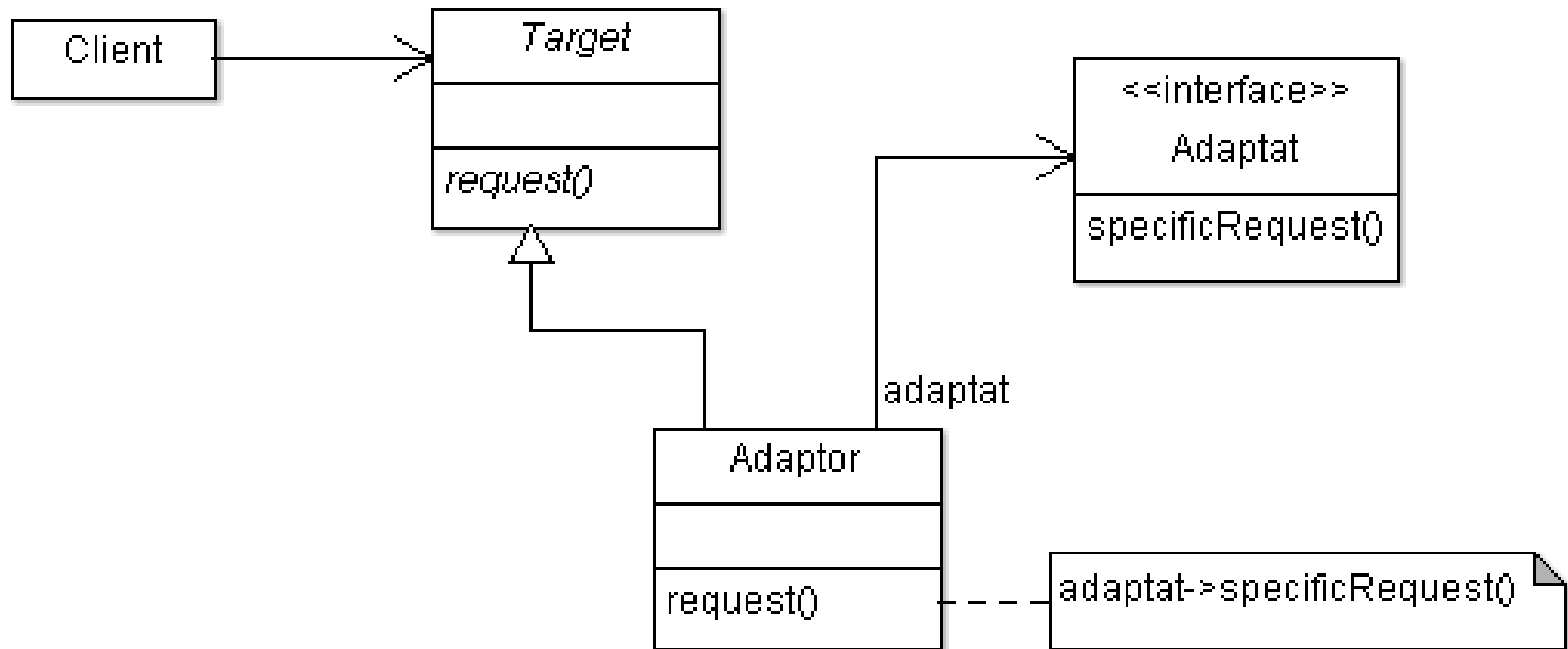


Cum poate exista o clasa ca *TextView* intr-o aplicatie in care clasele au interfețe diferite și incompatibile cu cea a lui *TextView*? Putem schimba *TextView* astfel incat sa fie conforma cu interfata clasei *Shape*?

# Structura – bazate pe mostenire multipla



# Structura – bazate pe compozitie slaba



# Implementare (interfetele)

---

```
class Shape {
public: Shape();
    virtual void boundingBox( Point& bottomLeft,
                           Point& topRight ) const;
    virtual manipulator* CreateManipulator() const;
};

class TextView {
public:
    TextView();
    void getOrigin(Coord& x, Coord& y) const;
    void getExtent(Coord& width, Coord& height)
                                                const;

    virtual bool isEmpty() const;
};
```

# Implementare – adaptor

- solutia bazate pe mostenire multipla

TextShape este subclasa (subtip) a clasei Shape

```
class TextShape : public Shape,  
                  private TextView {  
public:  
    TextShape() ;  
  
    virtual void boundingBox(  
        Point& bottomLeft, Point& topRight  
    ) const;  
    virtual bool isEmpty() const;  
    virtual manipulator* CreateManipulator()  
const;  
};
```

TextShape nu este subclasa (subtip) a clasei TextView



# Implementare

---

- operatia *boundingBox* convertește interfata clasei *TextView* launa conforma cu cea a clasei *Shape*

```
void TextShape::BoundingBox (
    Point& bottomLeft, Point& topRight
                                ) const {
    Coord bottom, left, width, height;

    GetOrigin(bottom, left);
    GetExtent(width, height);

    bottomLeft = Point(bottom, left);
    topRight = Point(bottom + height,
                      left + width);
}
```

# Implementare

---

- operatia *isEmpty* demonstreaza retrimitere directa a cererii la operatia mostenita

```
bool TextShape::isEmpty () const {  
    return TextView::isEmpty();  
}
```

- *createManipulator* nu este definit pentru Text, asa ca se defineste de la zero

```
Manipulator* TextShape::CreateManipulator ()  
    const {  
    return new TextManipulator(this);  
}
```