







HELP UKRAINE STOP RUSSIA

☆ / Паттерны проектирования / Адаптер / Java



Адаптер — это структурный паттерн, который позволяет подружить несовместимые объекты.

Адаптер выступает прослойкой между двумя объектами, превращая вызовы одного в вызовы понятные другому.

Подробней о паттерне Адаптер >

Навигация

- **Ш** Интро
- 💷 Помещение квадратных колышков в круглые отверстия
- round
- RoundPeg
- □ square
- □ adapters
- Demo
- **OutputDemo**

Сложность: 🛊 🏠 🏠

Популярность: 🛊 🛊 🛊

14.02.2023, 11:57 Адаптер на Java

Применимость: Паттерн можно часто встретить в Java-коде, особенно там, где требуется конвертация разных типов данных или совместная работа классов с разными интерфейсами.

Примеры Адаптеров в стандартных библиотеках Java:

```
    java.util.Arrays#asList()
    java.util.Collections#list()
    java.util.Collections#enumeration()
    java.io.InputStreamReader(InputStream) (ВОЗВРАЩАЕТ ОБЪЕКТ Reader )
    java.io.OutputStreamWriter(OutputStream) (ВОЗВРАЩАЕТ ОБЪЕКТ Writer )
    javax.xml.bind.annotation.adapters.XmlAdapter#marshal() И #unmarshal()
```

Признаки применения паттерна: Адаптер получает конвертируемый объект в конструкторе или через параметры своих методов. Методы Адаптера обычно совместимы с интерфейсом одного объекта. Они делегируют вызовы вложенному объекту, превратив перед этим параметры вызова в формат, поддерживаемый вложенным объектом.

Помещение квадратных колышков в круглые отверстия

Этот простой пример показывает как с помощью паттерна Адаптер можно совмещать несовместимые вещи.

round

🖟 round/RoundHole.java: Круглое отверстие

```
package refactoring_guru.adapter.example.round;

/**
    * КруглоеОтверстие совместимо с КруглымиКолышками.
    */
public class RoundHole {
    private double radius;

    public RoundHole(double radius) {
        this.radius = radius;
    }
}
```

```
public double getRadius() {
    return radius;
}

public boolean fits(RoundPeg peg) {
    boolean result;
    result = (this.getRadius() >= peg.getRadius());
    return result;
}
```

🖟 round/RoundPeg.java: Круглый колышек

```
package refactoring_guru.adapter.example.round;

/**
    * КруглыеКолышки совместимы с КруглымиОтверстиями.
    */
public class RoundPeg {
    private double radius;

    public RoundPeg() {}

    public RoundPeg(double radius) {
        this.radius = radius;
    }

    public double getRadius() {
        return radius;
    }
}
```

≥ square

🖟 square/SquarePeg.java: Квадратный колышек

```
package refactoring_guru.adapter.example.square;

/**

* КвадратныеКолышки несовместимы с КруглымиОтверстиями (они остались в проекте

* после бывших разработчиков). Но мы должны как-то интегрировать их в нашу

* систему.

*/

public class SquarePeg {
    private double width;
```

```
public SquarePeg(double width) {
        this.width = width;
}

public double getWidth() {
    return width;
}

public double getSquare() {
    double result;
    result = Math.pow(this.width, 2);
    return result;
}
```

adapters

adapters/SquarePegAdapter.java: Адаптер квадратных колышков к круглым отверстиям

```
package refactoring_guru.adapter.example.adapters;
import refactoring_guru.adapter.example.round.RoundPeg;
import refactoring_guru.adapter.example.square.SquarePeg;
/**
* Адаптер позволяет использовать КвадратныеКолышки и КруглыеОтверстия вместе.
public class SquarePegAdapter extends RoundPeg {
    private SquarePeg peg;
    public SquarePegAdapter(SquarePeg peg) {
        this.peg = peg;
    }
    @Override
    public double getRadius() {
        double result;
        // Рассчитываем минимальный радиус, в который пролезет этот колышек.
        result = (Math.sqrt(Math.pow((peg.getWidth() / 2), 2) * 2));
        return result;
    }
}
```

🖟 Demo.java: Клиентский код

```
package refactoring_guru.adapter.example;
import refactoring_guru.adapter.example.adapters.SquarePegAdapter;
import refactoring guru.adapter.example.round.RoundHole;
import refactoring_guru.adapter.example.round.RoundPeg;
import refactoring_guru.adapter.example.square.SquarePeg;
/**
* Где-то в клиентском коде...
*/
public class Demo {
    public static void main(String[] args) {
        // Круглое к круглому — всё работает.
        RoundHole hole = new RoundHole(5);
        RoundPeg rpeg = new RoundPeg(5);
        if (hole.fits(rpeg)) {
            System.out.println("Round peg r5 fits round hole r5.");
        }
        SquarePeg smallSqPeg = new SquarePeg(2);
        SquarePeg largeSqPeg = new SquarePeg(20);
        // hole.fits(smallSqPeg); // Не скомпилируется.
        // Адаптер решит проблему.
        SquarePegAdapter smallSqPegAdapter = new SquarePegAdapter(smallSqPeg);
        SquarePegAdapter largeSqPegAdapter = new SquarePegAdapter(largeSqPeg);
        if (hole.fits(smallSqPegAdapter)) {
            System.out.println("Square peg w2 fits round hole r5.");
        if (!hole.fits(largeSqPegAdapter)) {
            System.out.println("Square peg w20 does not fit into round hole r5.");
        }
    }
}
```

🖹 OutputDemo.txt: Результат выполнения

```
Round peg r5 fits round hole r5.

Square peg w2 fits round hole r5.

Square peg w20 does not fit into round hole r5.
```

14.02.2023, 11:57 Адаптер на Java

Moct на Java →

ВЕРНУТЬСЯ НАЗАД

← Одиночка на Java

Главная	F
Рефакторинг	
Паттерны	0
Премиум контент	
Форум	

- © 2014-2023 Refactoring.Guru. Все права защищены.
- 🖪 Иллюстрации нарисовал Дмитрий Жарт
- Ш Хмельницкое шоссе 19 / 27, Каменец-Подольский, Украина, 32305
- ☑ Email: support@refactoring.guru

Условия использования

Связаться

Политика конфиденциальности

Использование контента