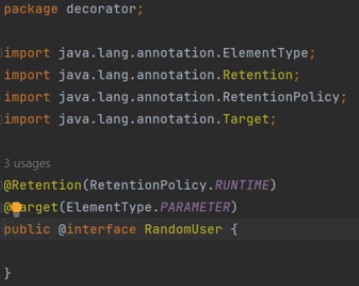
**Паттерн Observer**

**В**арианты использования:

**1.** Один класс зависит от другого.

Многократное заполнение полей сущностей разными параметрами, используя класс Random:

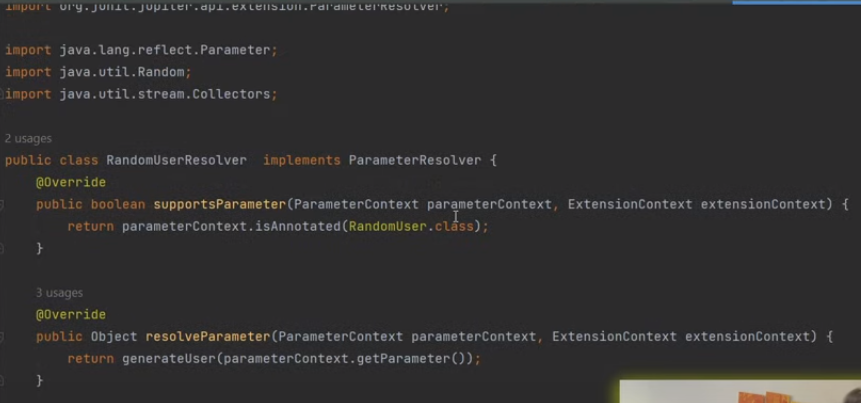
**I**. Создаем собственную аннотацию:



**II**. Реализуем класс, в котором:

- вызываем нашу аннотацию,

- прописываем действия, выполняемые при вызове нашей аннотации.





, где:

- **ParametrResolver** - определяет API для расширений, которые хотят динамически разрешать аргументы параметров во время выполнения. Если конструктор тестового класса или метод @Test, @BeforeEach, @AfterEach, @BeforeAll или @AfterAll объявляет параметр, аргумент для этого параметра должен быть разрешен во время выполнения с помощью ParameterResolver. Реализации должны предоставлять конструктор без аргументов.

- **Parametr** – класс с информацией о параметрах метода. Параметр предоставляет информацию о параметрах метода, включая его имя и модификаторы. Он также предоставляет альтернативные способы получения атрибутов параметра.

1. Имплементируем интерфейс **ParametrResolver.**

2. Переопределяем два метода интерфейса **ParametrResolver**, где:

- в методе ***supportsParameter*** мы привязываем к нашему классу созданную нами ранее аннотацию.

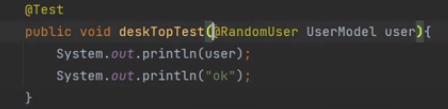
- в методе ***resolveParameter*** мы возвращаем 3-й метод с прописанной в нем логикой поведения

3. Заполняем логикой работы 3-й метод:

- проверяем введенный параметр на соответствие необходимому нам классу (тут “UserModel”)

- если проверка пройдена, то заполняем с помощью класса “Random” входящие параметры, каждый раз новыми данными.

**III**. В тесте, в параметрах метода, вначале прописав нашу аннотацию “@RandomUser”, мы запускаем механизм автоматического заполнения полей принимаемого на вход объекта класса “UserModel”:



**2.** Слушатель, который в отличие от Decarator-а, будет реализован не всегда, а только в случае выполнения каких то условий/действий *(класс Observable и интерфейс Observer deprecated, см.примеры https://javarush.com/groups/posts/3421-shablon-nabljudateljh-observer)*; **3.** ослабление связей между классами.

Класс Observable и интерфейс Observer из пакета java.util java Deprecation с java 9. Сейчас нужно использовать возможности пакета **java.beans**.

Для примера давайте попробует использовать PropertyChangeListener (чтобы не писать лишние классы, которые уже написаны) из пакета java.beans. **Давайте посмотрим как это будет:**

https://javarush.com/groups/posts/3421-shablon-nabljudateljh-observer

класс субъекта:

**public** **class** CatholicChurch {

**private** String news;

// используя support мы можем добавлять или удалять наших прихожан (слушателей)

**private** PropertyChangeSupport support;

**public** CatholicChurch() {

support = **new** PropertyChangeSupport(**this**);

}

**public** **void** addPropertyChangeListener(PropertyChangeListener pcl) {

support.addPropertyChangeListener(pcl);

}

**public** **void** removePropertyChangeListener(PropertyChangeListener pcl) {

support.removePropertyChangeListener(pcl);

}

**public** **void** setNews(String value) {

support.firePropertyChange("news", **this**.news, value);

**this**.news = value;

}

}

класс слушателя:

**public** **class** Parishioner **implements** PropertyChangeListener {

**private** String name;

**public** Parishioner(String name) {

**this**.name = name;

}

**public** **void** propertyChange(PropertyChangeEvent evt) {

**this**.setNews((String) evt.getNewValue());

}

**public** **void** setNews(String news) {

System.out.println(name + " узнал новость: " + news);

}

}

Если мы выполним следующий код:

**public** **static** **void** main(String[] args) {

CatholicChurch observable = **new** CatholicChurch();

observable.addPropertyChangeListener(**new** Parishioner("Мартин Лютер"));

observable.addPropertyChangeListener(**new** Parishioner("Жан Кальвин"));

observable.setNews("Дева Мария имеет непорочное зачатие... булла Ineffabilis Deus... 8 декабря 1854 года Папа Пий IX");

observable.setNews("Папа непогрешим... не всегда конечно, а только когда транслирует учение церкви ex cathedra... Первый Ватиканский собор 1869 год");

}

Получим следующий результат:

Мартин Лютер узнал новость: Дева Мария имеет непорочное зачатие... булла Ineffabilis Deus... 8 декабря 1854 года Папа Пий IX

Жан Кальвин узнал новость: Дева Мария имеет непорочное зачатие... булла Ineffabilis Deus... 8 декабря 1854 года Папа Пий IX

Мартин Лютер узнал новость: Папа непогрешим... не всегда конечно, а только когда транслирует учение церкви ex cathedra... Первый Ватиканский собор 1869 год

Жан Кальвин узнал новость: Папа непогрешим... не всегда конечно, а только когда транслирует учение церкви ex cathedra... Первый Ватиканский собор 1869 год