В чем разница между параметрическим полиморфизмом и полиморфизмом подтипов? В первом случае реализуется общий алгоритм для контейнера (например массива), который содержит значение или значения типа T. Этот алгоритм не зависит от T и для любых T выполняется идентично. Во втором — алгоритм построен вокруг самого объекта и использует его методы. В полиморфизме подтипов полиморфная функция работает только с теми объектами, которые имеют необходимые для реализации алгоритма методы.

Термин “полиморфизм” обозначает семейство различных механизмов, позволяющих использовать один и тот же участок программы с различными типами в различных контекстах.

ЗАДАЧА

**FakeSubscription.js**

Реализуйте и экспортируйте по умолчанию класс *FakeSubscription*. В конструктор *FakeSubscription* принимает пользователя. Если пользователь администратор user.isAdmin(), то все доступы разрешены, если нет – то запрещены. Класс должен повторять интерфейс класса Subscription, то есть иметь те же методы, но со своей логикой.

**User.js**

Допишите конструктор пользователя, так, чтобы внутри устанавливалась реальная подписка если она передана снаружи и создавалась фейковая в ином случае.

import FakeSubscription from './FakeSubscription.js';

class User {

  constructor(email, currentSubscription = null) {

    this.email = email;

    // BEGIN (write your solution here)

    this.currentSubscription = currentSubscription ?? new FakeSubscription(this);

    // END

  }

  getCurrentSubscription() {

    return this.currentSubscription;

  }

  isAdmin() {

    return this.email === 'rakhim@hexlet.io';

  }

}

export default User;

test("stack's main flow", () => {

  // Добавляем два элемента в стек и затем извлекаем их

  const ex = 1

  const en = 1

  expect(ex).toEqual(en);

});