

Наследование, интерфейсы

(задания со * необязательные для тех, кто не хочет стать программистом)

0. // Для тех, кто еще не пользуется git.

Создайте удаленный репозиторий на bitbucket.org. На диске создайте папку, в нее склонируйте удаленный репозиторий. Напишите файл .gitignore и отправьте его в удаленный репозиторий. Опробуйте основные операции с git (status, add, rm, commit, pull, push и т.д.). Создайте в репозитории ветку develop. Создайте любой файл и отправьте его в ветку develop. Создайте ветку test-feature и переключитесь в нее. Отредактируйте файл и отправьте его в ветку test-feature. Приклейте (merge) ветку test-feature в ветку develop. Ветку test-feature удалите.

С этого момента и до конца учебного года предполагается, что все проекты оформляются в виде maven-проектов и хранятся в git-репозитории. Все основные методы должны быть снабжены unit-тестами.

1. Создайте maven-проект. Напишите набор следующих классов. Каждый класс должен содержать подходящие конструкторы, геттеры полей, методы equals и hashCode, toString. Организуйте эти классы в подходящие иерархии по наследованию. При необходимости добавьте свои классы / интерфейсы для получения подходящей иерархии классов. Напишите необходимые unit-тесты.
 - 1.1. Класс «Упаковка товара». Упаковка имеет название и массу в кг (собственная масса упаковки).
 - 1.2. Класс «Товар». Товар имеет название и описание (строки).
 - 1.3. Класс «Весовой товар». Весовой товар хранит название и описание.
 - 1.4. Класс «Штучный товар». Штучный товар хранит название, описание и вес одной штуки товара (в кг). Должен быть конструктор по всем трем полям.
 - 1.5. Класс «Упакованный весовой товар» содержит упаковку и некоторое количество весового товара (в кг). Методы: конструктор по товару, массе товара и упаковке, получить массу нетто (только товара) и массу брутто (упаковки и товара вместе).
 - 1.6. Класс «Упакованный штучный товар» содержит упаковку и несколько единиц штучного товара. Методы: получить количество штук товара, получить массу нетто (суммарный вес всех единиц товара) и массу брутто (упаковки и всех единиц товара вместе).
 - 1.7. *Класс «Упакованный набор товаров». Набор состоит из собственной упаковки и некоторого количества упаковок товаров разного вида (в том числе других наборов). Методы: конструктор по произвольной последовательности упаковок товаров, масса нетто (суммарная масса брутто всех элементов набора, но без массы упаковки самого набора), масса брутто (суммарная масса всего набора с упаковкой).

1.8. Класс «Партия товаров». Партия товаров имеет описание и содержит некоторое количество упакованных товаров, возможно, разных типов (в том числе наборов товаров). Методы: конструктор по произвольному набору упаковок товаров, получить массу всей партии (суммарная масса брутто всех товаров в партии).

2. Напишите следующие классы и интерфейс.

2.1. Интерфейс «Фильтр» с единственным методом `apply`. Метод получает на вход строку и возвращает логическое значение (строка удовлетворяет условию фильтра или нет).

2.2. Класс `BeginStringFilter`, реализующий этот интерфейс, со следующим поведением. При создании объекта класса конструктор получает на вход и сохраняет строку `pattern`. Метод `apply(str)` проверяет, что строка `str` начинается с подстроки `pattern`.

Т.е. должен работать следующий фрагмент кода:

```
String str = "Мама мыла раму";
BeginStringFilter filter1 = new BeginStringFilter("Мама");
BeginStringFilter filter2 = new BeginStringFilter("мыла");
boolean res1 = filter1.apply(str); // результат — true
boolean res2 = filter2.apply(str); // результат — false
```

2.3. Напишите еще две какие-нибудь реализации интерфейса «Фильтр».

3. Напишите класс «Сервис товаров» со следующими методами.

3.1. Метод `countByFilter`, который получает на вход партию товара и произвольный фильтр из п.2. Метод возвращает количество элементов партии, имена которых удовлетворяют фильтру. Для товаров-наборов анализируется только название самого набора.

3.2. *Метод `countByFilterDeep` — ведет себя аналогично методу `countByFilter`, но набор считается подходящим, если он содержит (возможно на некоторой глубине) хотя бы один товар с названием, удовлетворяющим фильтру.

3.3. *Метод `checkAllWeighted`, который для партии товаров проверит, что вся партия состоит только из весовых товаров.