### Фабрики (1)

### Привет! Сейчас ты будешь разбираться с паттерном Factory Method. Возьмем пример с людьми. Будем определять, к какому типу относится человек в зависимости от его возраста.

### В этом задании подготовим данные, которые будем использовать в фабрике.

### 1. Для всех типов людей создай общий интерфейс Human.

### 2. Будем делать Factory для мужчин. Для этого создай пакет male, в котором создай классы KidBoy, TeenBoy, Man. Во всех этих классах реализуй интерфейс Human.

### 3. В каждом классе KidBoy, TeenBoy создай свою публичную константу MAX\_AGE с соответствующими значениями: 12, 19.

### 4. Сгенерируй автоматические методы toString для каждого класса KidBoy, TeenBoy, Man. Получится "KidBoy{}" и т.п.

### 5. Создай класс Solution с методом main.

### Фабрики (2)

### 1. Создай публичный класс MaleFactory в пакете male с публичным методом getPerson, который принимает один параметр intage (возраст) и возвращает нужный объект одного из классов, реализующих интерфейс Human.

### 2. Логика метода getPerson: используя константу MAX\_AGE определи, какой класс соответствует переданному возрасту. Например, age=4 соответствует мальчику (KidBoy), age=15 - подростку (TeenBoy), остальной возраст - взрослому мужчине.

### 3. В методе main класса Solution создай фабрику и вызови у нее метод getPerson три раза с такими параметрами: 99, 4, 15.

### Выведи результат в консоль. Вывод должен быть следующий: Man{} KidBoy{} TeenBoy{}

### Молодец, Фабрика готова! Это паттерн Factory Method.

### Требования: 1. Вывод на экран должен соответствовать условию задачи. 2. В методе main класса Solution у MaleFactory должен быть вызван трижды метод getPerson с параметрами 99, 4, 15. 3. Для возрастов больше 19 лет метод getPerson должен возвращать объект типа Man. 4. Для возрастов больше 12 лет, но меньше 20, метод getPerson должен возвращать объект типа TeenBoy. 5. Для возрастов меньше 13 лет, метод getPerson должен возвращать объект типа KidBoy.

### Фабрики (3)

### 1. Создай пакет female, в котором создай KidGirl, TeenGirl и Woman аналогично классам из пакета male.

### 2. Создай в пакете female фабрику FemaleFactory аналогичную MaleFactory с тем же принципом получения объектов.

### 3. Подумай, что общего у двух фабрик? Что можно сделать, чтобы легко можно было переключаться между ними?

### Ответ - в следующем задании.

### Требования: 1. Классы KidGirl, TeenGirl, Woman должны быть размещены в пакете female и поддерживать интерфейс Human. 2. В классах KidGirl, TeenGirl, Woman должен быть переопределен метод toString в соответствии с условием задачи. 3. В классе KidGirl константа MAX\_AGE должна быть равна 12. 4. В классе TeenGirl константа MAX\_AGE должна быть равна 19. 5. Для возрастов больше 19 лет метод getPerson фабрики FemaleFactory должен возвращать объект типа Woman. 6. Для возрастов больше 12 лет, но меньше 20, метод getPerson фабрики FemaleFactory должен возвращать объект типа TeenGirl. 7. Для возрастов меньше 13 лет, метод getPerson фабрики FemaleFactory должен возвращать объект типа KidGirl.

### Фабрики (4)

### 1. В корне задачи создай интерфейс AbstractFactory, в который вынеси общий в фабриках метод.

### 2. Реализуй интерфейс AbstractFactory в обеих фабриках.

### 3. В корне задачи создай класс FactoryProducer, в котором создай публичный статический энум HumanFactoryType со значениями MALE, FEMALE.

### 4. В FactoryProducer создай публичный статический метод getFactory с аргументом HumanFactoryType. Этот метод должен возвращать одну из фабрик: для ключа MALE - MaleFactory, для FEMALE - FemaleFactory.

### Молодец, Фабрика фабрик готова! Это паттерн Abstract Factory. В коде тебе не важно, какая фабрика используется.

### Твои успехи: 1. Освоен паттерн Factory Method. 2. Освоен паттерн Abstract Factory. 3. +2 паттерна для прохождения собеседования.

### Работа джава программистом стала еще ближе, скоро достигнешь цели!

### Требования: 1. Класс MaleFactory должен поддерживать интерфейс AbstractFactory. 2. Класс FemaleFactory должен поддерживать интерфейс AbstractFactory. 3. В классе FactoryProducer должен быть создан enum HumanFactoryType со значениями MALE, FEMALE. 4. Метод getFactory в классе FactoryProducer должен принимать один параметр типа HumanFactoryType. 5. Метод getFactory в классе FactoryProducer должен возвращать одну из фабрик в зависимости от принятого параметра.