**Рефакторинг (1)**

Пришло время немного порефакторить.  
Википедия говорит: «*Рефакторинг (англ. refactoring) или реорганизация кода — процесс изменения внутренней структуры программы, не затрагивающий её внешнего поведения и имеющий целью облегчить понимание её работы. В основе рефакторинга лежит последовательность небольших эквивалентных (то есть сохраняющих поведение) преобразований. Поскольку каждое преобразование маленькое, программисту легче проследить за его правильностью, и в то же время вся последовательность может привести к существенной перестройке программы и улучшению её согласованности и четкости.» Такими маленькими преобразованиями мы и будем заниматься. После каждого изменения следи за сохранностью работоспособности кода. Многие методы рефакторинга не однократно применены и проверены сообществом программистов. Получили свои названия. Каждое задание будет сопровождено названием рефакторинга. Ты всегда можешь прочитать более подробно о нем в книге Мартина Фаулера «Рефакторинг: Улучшение существующего кода*«.

Начнем. Внимательно ознакомься с кодом пакета human (человек).

Задания:  
1.1. Подъем поля. Подними поле children в базовый класс.  
1.2. Подъем метода. Подними сеттер и геттер для children в базовый класс.  
1.3. Инкапсуляция коллекции.  
1.3.1. Метод getChildren должен возвращать не модифицируемое представление списка children.  
1.3.2. Убери сеттер для children.  
1.3.3. Добавь методы addChild(Human) и removeChild(Human). Реализуй их логику.

### Рефакторинг (2)

2.1. Извлечение подкласса.  
2.1.1. Добавь класс Soldier в пакет human.  
2.1.2. Избавься от поля isSoldier.  
2.1.3. Перенеси в Soldier необходимые методы из **Human**.  
2.1.4. Обнови сигнатуру конструктора **Human**.  
2.2. Подъем тела конструктора.  
2.2.1. Перенеси инициализацию полей name и age в подходящее место, добавь необходимые параметры в конструктор **Human**.  
2.2.2. Добавь конструктор в класс Soldier.

**Рефакторинг (3)**

3.1. Спуск поля. Спусти поле course в соответствующий класс. Сделай его приватным.  
3.2. Спуск метода. Спусти геттер для поля course в соответствующий класс.  
3.3. Извлечение интерфейса.  
3.3.1. Создай интерфейс Alive (живой) в пакете human.  
3.3.2. Интерфейс должен содержать метод жить live().  
3.3.3. Добавь интерфейс нужному классу.  
3.4. Свертывание иерархии. Избавься от класса Professor.

Требования:  
1. Необходимо спустить поле course в нужный класс и сделать его приватным.  
2. Необходимо спустить геттер для поля course в нужный класс.  
3. Интерфейс Alive должен существовать в пакете human.  
4. Интерфейс Alive должен содержать объявление метода жить live().  
5. Класс Human должен поддерживать интерфейс Alive.  
6. Класс Professor нужно удалить.