JP2 2019: Lab exam (Version A)

This lab exam is intended to be completed during the 9-11am Monday timeslot (Lab sections 1, 2, and 13). If you are taking the exam at a different time, you should locate the correct specification instead of this one.

Overview

Your task is to develop classes to model a video game where players can gather **resources** in the wild, and can also **craft** new resources from existing resources using a **recipe** that specifies the number and type of resources that are needed. The following are the main features that you will implement – the remainder of this lab sheet describes in more detail how the classes should be created.

Every **Player** of the video game has an inventory of the resources they are currently carrying. For example, a Player might be carrying the following resources:

- Five pieces of **Wood**
- Three **Diamonds**
- Four **Apples**
- Three pieces of **Gold**
- Three pieces of **Iron**

In addition, the player has access to a set of **Recipes** which specify how new resources can be crafted from the resources they are holding. For example, the following are all recipes:

- Three pieces of **Iron** plus one piece of **Wood** produces a **Sword**
- One piece of **Gold** plus an **Apple** produces a **Golden Apple**
- Five **Diamonds** produce a **Helmet**

If a player has the list of resources given above, they are able to craft a Sword and a Golden Apple, but cannot craft a Helmet because they do not have enough Diamonds. If the player crafts a Sword, their inventory would be the following afterwards:

- Four pieces of **Wood**
- Three **Diamonds**
- Four **Apples**
- Three pieces of **Gold**
- One **Sword**

Task 1: Resource (2 marks)

Note about implementation: all classes created in this exam should be put in the ***crafting*** *package.*

You must create an enumerated type **Resource** with the following values:

WOOD, STONE, IRON, GOLD, DIAMOND, APPLE, POTATO, CARROT, SWORD, SHIELD, HELMET

Task 2: Player (6 marks)

You must create a class **Player** representing a player including the following properties:

- name: (a String)
- inventory: (a Map<Resource, Integer> representing the resources that the player has)

The Player class should have a constructor with the following signature to initialise the fields:

public Player(String name, Map<Resource, Integer> inventory)

The **Player** class should also include the following:

- A complete set of **get** methods, but no **set** methods
- Appropriate implementations of **equals()**, **hashCode()**, and **toString()**

Task 3: Recipe (4 marks)

Create a class **Recipe** representing a recipe including the following properties:

- result: (a Resource)
- requirements: a Map<Resource, Integer> indicating the resources needed to craft the result

You should also include a constructor that sets the above two fields, as well as **get** methods to access both values. **You do not need to override equals(), hashCode(), or toString() for this class.**

For example, the **Sword** recipe from the previous page would be represented as follows:

- result: **Sword**
- requirements: {**Iron**=3, **Wood**=1}

Task 4: Crafting (7 marks)

In your **Player** class, implement one additional public method, as follows:

- **public boolean craft(Resource resource, Set<Recipe> recipes)**

This method should behave as follows:

- Search through the given set of Recipes to find one that produces the given resource. You can assume that there is at most one such Recipe in the list.
- If no suitable Recipe is found, return false
- If a suitable Recipe is found:
    - Check that the user has sufficient resources – if not, return false
    - Remove the necessary Resources from the player's inventory
    - Add the crafted Resource to the player's inventory

For example, with the inventory and the set of recipes given on the first page, here is how the method would work:

- **craft(Helmet)** would return false because the player does not have the resources – the inventory would be unchanged
- **craft(Sword)** would return true, and after the call the player's inventory would include:
    - Four pieces of **Wood**
    - Three **Diamonds**
    - Four **Apples**
    - Three pieces of **Gold**
    - One **Sword**

What to submit

On Moodle, go to **Lab Exam Submission – VERSION A** and upload the following three files:

- Resource.java
- Player.java
- Recipe.java

Be sure to submit to the correct submission link, and be sure to submit before the deadline.