

Task 3: RPG Characters

Brief

This Task is designed to test your modelling and class design skills. To successfully complete it you will have to demonstrate your understanding of inheritance, polymorphism, composition, and SOLID principles. You should make your code as extendable as possible – meaning it is easy to add new weapons, characters, abilities, and so on. You can feel free to include some design patterns if you see their usefulness, some you could look to include are: strategy (will help a lot), factory, and builder.

What you are required to do is model and demonstrate a simple RPG character system. No game logic is required aside from the aspects explicitly mentioned in this document. The RPG system is very basic by design. This is meant to be something simple and easy to create in the given time frame – however, there is enough openness to allow for extension.

NB: Instead of calling them Character and Characters, you can call them Hero and Heroes – this can avoid some ambiguity with built in Java objects.

HINT: When deciding on character and item designs, be aware of deep inheritance hierarchies, this is not something that is ideal. An inheritance hierarchy should be avoided if you could have the same outcome if composition were used. If an inheritance hierarchy is needed, it should not be deeper than 3 levels, if you need more, reconsider your design. For some more information on this read up about “composition over inheritance”. The best way to see where you can replace inheritance with composition is if the children are not extending the parents (adding any extra fields or behaviors) but simply having the same attributes/fields with different values.

The characters

There are several attributes that all characters will have: Health, Strength, Dexterity, and Intelligence.

Health

Health determines how much health a character has.

Strength

Strength boosts the character's damage done with melee weapons. 1 point of Strength results in 1.5 additional melee weapon damage.

Dexterity

Dexterity boosts the character's damage with ranged weapons. 1 point in Dexterity results in 2 additional ranged weapon damage.

Intelligence

Intelligence boosts the character's spell damage. 1 point in Intelligence results in 5 additional spell damage done (NB: Spells are not to be implemented).

In addition to these attributes, a character has a Level. All characters start at level 1 and required 100xp to get to level 2. The required experience to get to the next level is increased by 10% each time, rounded to the nearest whole number, i.e. 100 -> 110 -> 121, etc..

A snippet is shown below to illustrate the level up process.

```
Warrior gaining 500xp!  
  
Level up!  
Now level 2  
  
Level up!  
Now level 3  
  
Level up!  
Now level 4  
  
Level up!  
Now level 5  
10 more needed to level up
```

A character can interact with Items in various ways. Details of items are discussed later.

A character has 4 slots where they can equip items:

- Weapon
- Head
- Body
- Legs

A character can equip any kind of item, i.e. a Warrior can wear Cloth armor pieces and use a Bow as a weapon (due to items offering different stats, a Warrior may not want to do this).

IMPORTANT: There is no requirement to include spells or spell functionality at this point.

Character classes

Currently, there are 3 classes in the game: Warrior, Ranger, and Mage. You should design the classes so that more can be added in future.

Warrior

Thematically, Warriors are mighty brutes well versed in melee combat. For this reason, they have naturally higher Strength than other classes.

- Warriors begin at level 1 with: 150 HP, 10 Str, 3 Dex, 1 Int.
- On level up, a Warrior gains: 30 HP, 5 Str, 2 Dex, 1 Int.

Ranger

Thematically Rangers are masters of combat at a distance, and are proficient with missile (ranged) weapons, they have naturally higher Dexterity than other classes.

- Rangers begin at level 1 with: 120 HP, 5 Str, 10 Dex, 2 Int.
- On level up, a Ranger gains: 20 HP, 2 Str, 5 Dex, 1 Int.

Mage

Thematically Mages are masters of arcane and magic. They are natural spell casters and can cast more powerful spells than other classes, they have naturally higher intelligence than other classes.

- Mages begin at level 1 with: 100 HP, 2 Str, 3 Dex, 10 Int.
- On level up, a Mage gains: 15 HP, 1 Str, 2 Dex, 5 Int.

Example outputs

Shown below will be some example outputs for various classes created. Please note, this is just displaying the character stats and not including their equipment.

Warrior details:

HP: 390

Str: 50

Dex: 19

Int: 9

Lvl: 9

XP to next: 151

Mage details:

HP: 235

Str: 20

Dex: 30

Int: 100

Lvl: 10

XP to next: 84

Ranger details:

HP: 340

Str: 27

Dex: 65

Int: 13

Lvl: 12

XP to next: 121

Items

As mentioned previously, there are different types of items: **armor** and **weapons**. There is cloth armor, leather armor, plate armor, melee weapons, and ranged weapons. All items have a level associated with them, which is a required level to equip as well as scaling base damage. All items have a particular slot they fit into: Weapon, Head, Body, or Legs. All items have a name.

Weapons

Weapons have a **base damage** which is increased by Dexterity for Ranged Weapons or Strength for Melee Weapons when calculating damage dealt. Weapons have no other stats on them.

- All melee weapons have a base damage of 15, scaling by 2 every level. When a character attacks with a melee weapon, the damage dealt is the damage of the weapon (with level scaling included) + the character's effective (base + equipment) Strength value * 2.
 - Meaning the damage of a melee weapon is increased by $2 * \text{Str}$
- All ranged weapons have a base damage of 5, scaling by 3 every level. When a character attacks with a ranged weapon, the damage dealt is the damage of the weapon (with level scaling included) + the character's effective (base + equipment) Dexterity value * 1.5.
 - Meaning the damage of a ranged weapon is increased by $1.5 * \text{Dex}$, converted to int (rounded down).

Armor

Armor can apply bonuses to Health, Strength, Dexterity, and/or Intelligence. Changing depending on the type of armor. The slot effects how the armor stats are scaled after calculations.

- A Cloth armor piece has a base bonus of 10 HP, 3 Int, 1 Dex.
 - Scaled with 5 HP, 2 Int and 1 Dex per level.
- A Leather armor piece has a base bonus of 20 HP, 3 Dex, 1 Str.
 - Scaled with 8 HP, 2 Dex and 1 Str per level.
- A Plate armor piece has a base bonus of 30 HP, 3 Str, 1 Dex.
 - Scaled with 12 HP, 2 Str and 1 Dex per level.

Finally, all armor scales depending on what slot it is in.

- Body is 100%
- Head is 80%,
- Legs are 60%.

This means that after the bonuses have been calculated (base bonus + bonus from levels), all the stats are scaled according to slot – rounded down to the nearest whole number (converted to int).

Example outputs

Shown below will be some example outputs for various items created. Note: The names of the items are typed in manually, not generated, in these screen shots.

```
Item stats for: Great Axe of the Exiled  
Weapon Type: Melee  
Weapon level: 5  
Damage: 25
```

```
Item stats for: Long Bow of the Lone Wolf  
Weapon Type: Ranged  
Weapon level: 10  
Damage: 35
```

```
Item stats for: Cloth Leggings of the Magi  
Armor Type: Cloth  
Slot: Legs  
Armor level: 10  
Bonus HP: 36  
Bonus Dex: 6  
Bonus Int: 13
```

```
Item stats for: Plate Chest of the Juggernaut  
Armor Type: Plate  
Slot: Body  
Armor level: 15  
Bonus HP: 210  
Bonus Str: 33  
Bonus Dex: 16
```

When a character equips an item, their stats should update accordingly, a way of approaching this is having base stats, and stats added by gear stored separately – this could avoid issues when changing gear around and leveling up. When a new piece of equipment is added when there is an existing piece in that slot (i.e. adding new boots when you already have boots) it should just replace the existing equipment.

Characters with equipped items

As mentioned, a character has its attributes altered by equipped gear. This means that when showing the characters attributes, the following should occur, using Strength as the example:

- $\text{Effective Str} = \text{Character Str} + \text{Str from gear}$.

The Warrior from above is used to demonstrate this. They are levelled up to match the above example, then equipped with the created weapon and armor.

Warrior details:

HP: 390

Str: 50

Dex: 19

Int: 9

Lvl: 9

XP to next: 151

Item stats for: Great Axe of the Exiled

Weapon Type: Melee

Weapon level: 5

Damage: 25


```
Item stats for: Plate Chest of the Juggernaut
Armor Type: Plate
Slot: Body
Armor level: 5
Bonus HP: 90
Bonus Str: 13
Bonus Dex: 6
```

This results in the Warrior now having the following stats:

```
Warrior details:
HP: 480
Str: 63
Dex: 25
Int: 9
Lvl: 9
XP to next: 151
```

Attack damage

When a character attacks, their weapon will deal its base damage + any bonus from stats. If a character has no weapon equipped, this should result in no damage dealt.

Keep in mind, different weapons scale with different stats. This is detailed earlier in the document.

Warrior details:

HP: 480

Str: 63

Dex: 25

Int: 9

Lvl: 9

XP to next: 151

Attacking for 119

Demonstration

The requirements for demonstration are straight forward, at a minimum they are:

- Generate some characters and demonstrate the level up by giving them xp – that their attributes increase at higher levels.
- Create some items; demonstrate the creation of a melee weapon, ranged weapon, a helmet, body, and legs. The armor can be a mixture of cloth, leather, and plate.
- Equip some items to characters; this demonstrates that the characters effective stats are altered by new equipment.
- Change the equipment of characters; this demonstrates that changing equipment works properly – meaning that attributes wont infinitely increase and that equipping an item with 50 less strength results in a character's strength being 50 less. Hint: having base stats and bonus stats separate helps a lot here.
- Showcase the characters attacking, this should just display the amount of damage dealt.
 - This does not deal damage to another character. It should simply output the amount of damage dealt.

Feel free to demonstrate this further or showcase any additions made to the requirements, i.e. actually damaging another character, changing their health accordingly.

Submission

Provide a GIT repo link as submission. This repo should contain a well formatted (its written in markdown, so look how to make headings, #Heading1, ...) README explaining what your project does. Some extra inspirations could be provided if you want.

Remember, you can use your tasks as a portfolio, so try and make it as attractive as possible.

Comments

Provide comments to explain functionality. More comments are better and will result in a better mark. Be aware not to over comment just for the sake of placing a comment on every line. Comments should serve the purpose of explaining – I should be able to come into the application and understand exactly what it does and how it works from reading comments, variable names, and method names.