Preface:

* Please note that my original (5a) submission and my second (5b) submission are in two separate GitHub repositories. Due to the nature of having to restart completely, a significant amount of time was lost trying to troubleshoot and resolve the issues involved with adding Gradle and its subsequent programs (adding them made it so my IDE could not recognize any main).
* (5a) <https://github.com/JoshBenn/ser316-summer2023-C-jdbenn-assign5>
* (5b) <https://github.com/JoshBenn/ser316-summer2023-C-jdbenn-assign5b>
* The latest running version of the game will be in the “master” branch.

Design Patterns Used:

* Decorator Pattern:
  + Multiple character types were created under a parent Character class. Each character type can be considered a Decorator pattern. The three decorations are Cat, Demon, and Human. This same design pattern was used as well for the various Floors and Armor items. The floors implemented are Normal, Medium, and Boss. The Armor items implemented are Weapon, Helmet, Chest, Pants, and Boots. Each of these subsets are decorations of their respective supersets.
  + The requirements fulfilled through this pattern style are:
    - Having different classes:
      * Cat – Mage.
      * Demon – Warrior
      * Human – Either
    - Having Different Races (As stated above).
    - The passive skill that gets better for each are:
      * Cat – Magic Power.
      * Demon – Physical Power.
      * Human – Accuracy.
    - The floor encountered and the enemy on that floor:
      * Normal – Basic 1\*floor health enemy.
      * Medium – More difficult 2\*floor health enemy.
      * Boss – Most difficult 3\*floor health enemy.
    - 5 Unique equipped armor items are created:
      * Each with their own respective accuracy and penetration modifiers depending on the item type and the floor they were received.
      * These are automatically equipped upon retrieval as they will always be better than previous.
    - Potions, Scrolls, and Buffs are intended to follow this pattern but currently are implemented in a more static methodology due to time constraints.
      * Potions currently just heal health to max or remove (non-implemented) negative status effects.
      * Buffs and scrolls can be bought and will show their status on the overall GUI, but do not actually perform any function yet.
* Builder Pattern:
  + In order to manage building each of the items, whether direct generation or random generations, a builder pattern was created for each item, floor, or character superset.
    - At the start of the game, the player is asked to create a character and the character builder is called.
    - At each floor, a floor builder is called to build the floor and that floor builder calls the character builder to build an enemy.
    - At the start of the game, 5 unique items are created and equipped to the character. These items fill all the armor slots (weapon, helmet, chest, pants, boots).
    - After completing a Medium floor, a random non-weapon armor item is generated and equipped to the player.
    - After completing a Boss floor, a random weapon armor item is generated and equipped to the player.
* Strategy Pattern:
  + For most of the interactions from the player and/or choices made, strategy patterns were implemented to handle the options chosen and the directions those options lead to.
    - This is present in:
      * Character creation.
      * Floor encounters.
      * Shop interactions.
      * Item usage.
      * General interface interactions.
* GameState Machine:
  + The state of the game is tracked and passed along to each other system interaction as it dictates what each thing does.
    - Starting at and returning to the “main” floor (implemented as Home) sets all status back to original states. Permanent buffs are intended to remain permanent, however, this has not been implemented fully yet.
    - Events are not implemented as time did not allow for their implementation and thus cycles were not implemented either.
    - Stats of characters are:
      * Accuracy.
      * Health.
      * Penetration.
      * Potency.
    - Each character has an attack they can use.
      * This attack is either magic or physical based.
    - Each attack has a chance to miss based on the accuracy.
    - The player character gains experience after the completion of any floor.
      * The experience gained is dependent on the floor number and the floor value – medium and boss floors offer more experience.
    - No upgrade paths for skills or abilities are implemented yet.
* Mediator Pattern:
  + The combat itself is turn based and the player interactions are also turn based.
    - Items can be used and do not take a turn.
    - When the player dies, the game ends.
    - If the player survives a hit but that hit brings them to <15% health, they will return to the “main” floor.

Checkstyle Report:

* Main:
  + A picture containing text, software, line, screenshot

    Description automatically generated
* Test:
  + A screenshot of a computer

    Description automatically generated with medium confidence

Spotbugs Report:

* A screenshot of a computer

  Description automatically generated with medium confidence

JUnit Report:

* A screenshot of a computer

  Description automatically generated with medium confidence

Jacoco:

* A screenshot of a computer

  Description automatically generated with medium confidence
  + Most of the loc missed are within the main and shop files which handle player input and output (See the below image)
  + A screenshot of a computer

    Description automatically generated with medium confidence
  + ArmorBuilder, CharacterBuilder, and FloorBuilder are tested and all floor versions are tested through those.
  + Rough actual code coverage: ~80%.

SCREENCAST LINK!!!

* <https://youtu.be/jVKW8mHFY8Y>