HW2 Design
UML:

Character

- FOOT_WEAR_NUM: int = 2
- HAND_GEAR_NUM: int = 2
- HEAD_GEAR_NUM: int = 1
- name: String
- attackBase: int
- defenseBase: int
- handGear: List<Item>
- footWear: List<Item>
- headWear: Gear

public Character(String name,
int defenseBase,
int attackBase)

+ pickup(Gear g): void
+ getName: String
+ getTotalDefense(): int
+ getTotalAttack(): int
+ getHeadGear(): Gear
+ getHandGearList(): List<Gear>
+ getFootWearList(): List<Gear>
+ toString(): String

**<<Interface>>
CharacterInteface**

+ pickup(Gear g): void
+ getName: String
+ getTotalDefense(): int
+ getTotalAttack(): int
+ getHeadGear(): Gear
+ getHandGearList(): List<Gear>
+ getFootWearList(): List<Gear>

Battle

- character1: Character
- character2: Character
- gearList: List<Item>

public Battle(Character character1,
Character character2,
ArrayList<Gear> gearList)

+ getCharacter1(): Character
+ getCharacter2(): Character

+ check(Character c): boolean
+ oneTurn(): boolean
+ status(): String
+ winner(): String + toString(): String
+ addAfterCheck(Character c): boolean

**<<Interface>>
BattleInterface**

+ getCharacter1(): Character
+ getCharacter2(): Character
+ check(Character c): boolean
+ oneTurn(): boolean
+ status(): String
+ winner(): String
+ addAfterCheck(Character c): boolean

**<<Interface>>
Gear**

+ getDefense(): int
+ getAttack(): int
+ getType(): GearType
+ getAdj(): String
+ getNoun(): String
+ getTotalStrength(): int
+ combine(Gear other): Gear

**<<Abstract>>
AbstractGear**

- type: GearType
- attack: int
- defense: int
- adj: String
- noun: String

public AbstractGear(String adj,
          String noun,
          int attack,
          int defense,
          GearType type)
+ getDefense(): int
+ getAttack(): int
+ getType(): GearType
+ getAdj(): String
+ getNoun(): String
+ getTotalStrength(): int
+ toString(): String

GearType<Enum>

HEAD
HAND
FOOT

Extends

Extends

Extends

Head gear

public HeadGear(String adj,
String noun,
int defense)

+ combine(Gear g): Gear

Hand gear

public HandGear(String adj,
String noun.
int attack)

+ combine(Gear g): Gear

Foot wear

public FootWear(String adj,
String noun.
int attack,
int defense)

+ combine(Gear g): Gear
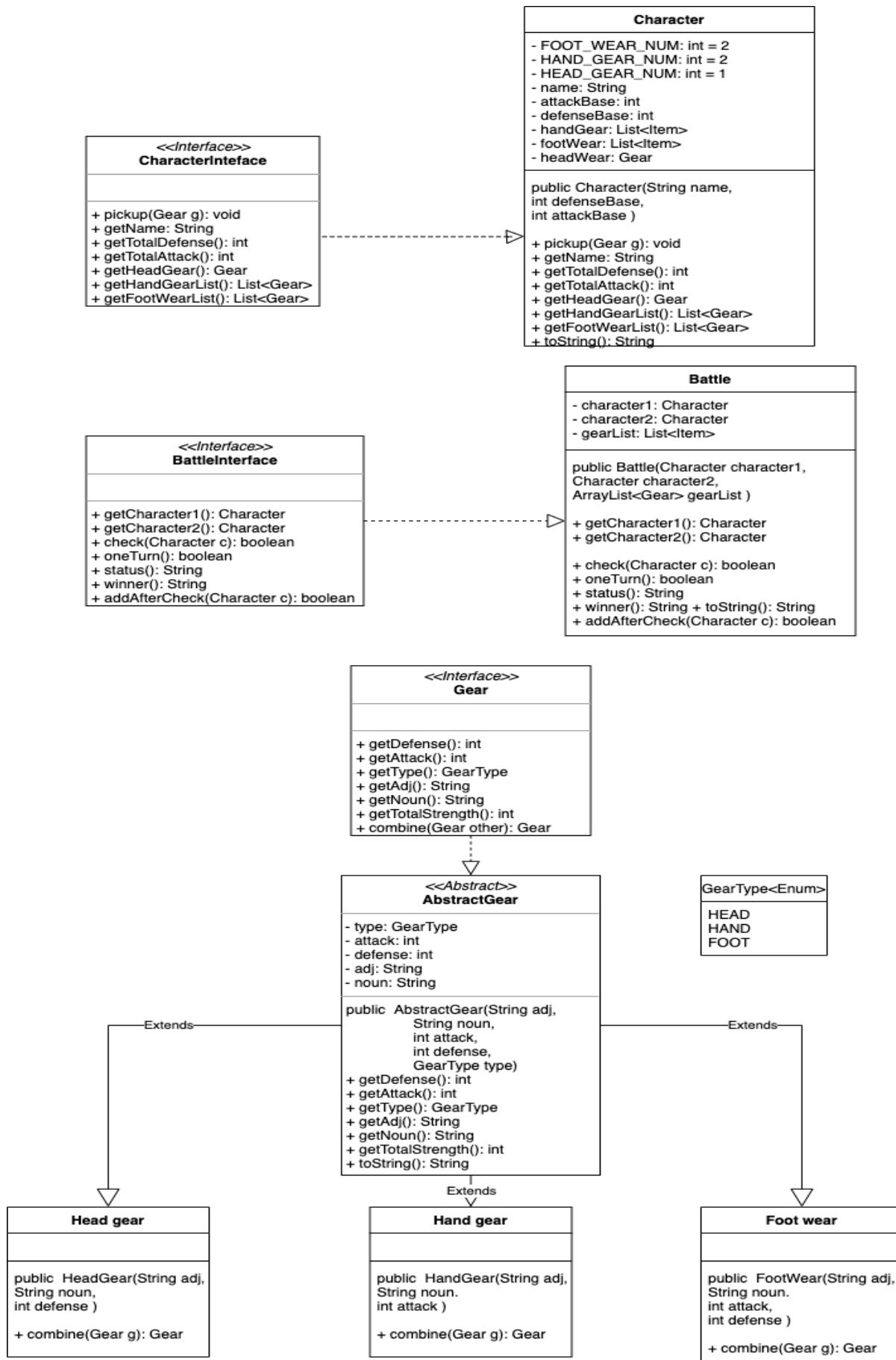
Test:

Character Test:
1. Check if the character has negative attack or defense power, throw illegalArgumentException
2. Test all get method for it.
3. Test the character has pick more than one piece of head gear, two pieces of footwear, and two pieces of hand gear
4. Test combine for head, hand, foot
5. Test combine item which should be same type.
   Character char1 = new Character (4,7);

GearTest:
1. Check if the item has null value name, throw illegalArgumentException
2. Check if the item has negative attack or defense power, throw illegalArgumentException
3. Test all get method for it.
   Item footWear = new footWear("Scurrying", "Sandals", 0, 1, "footWear");