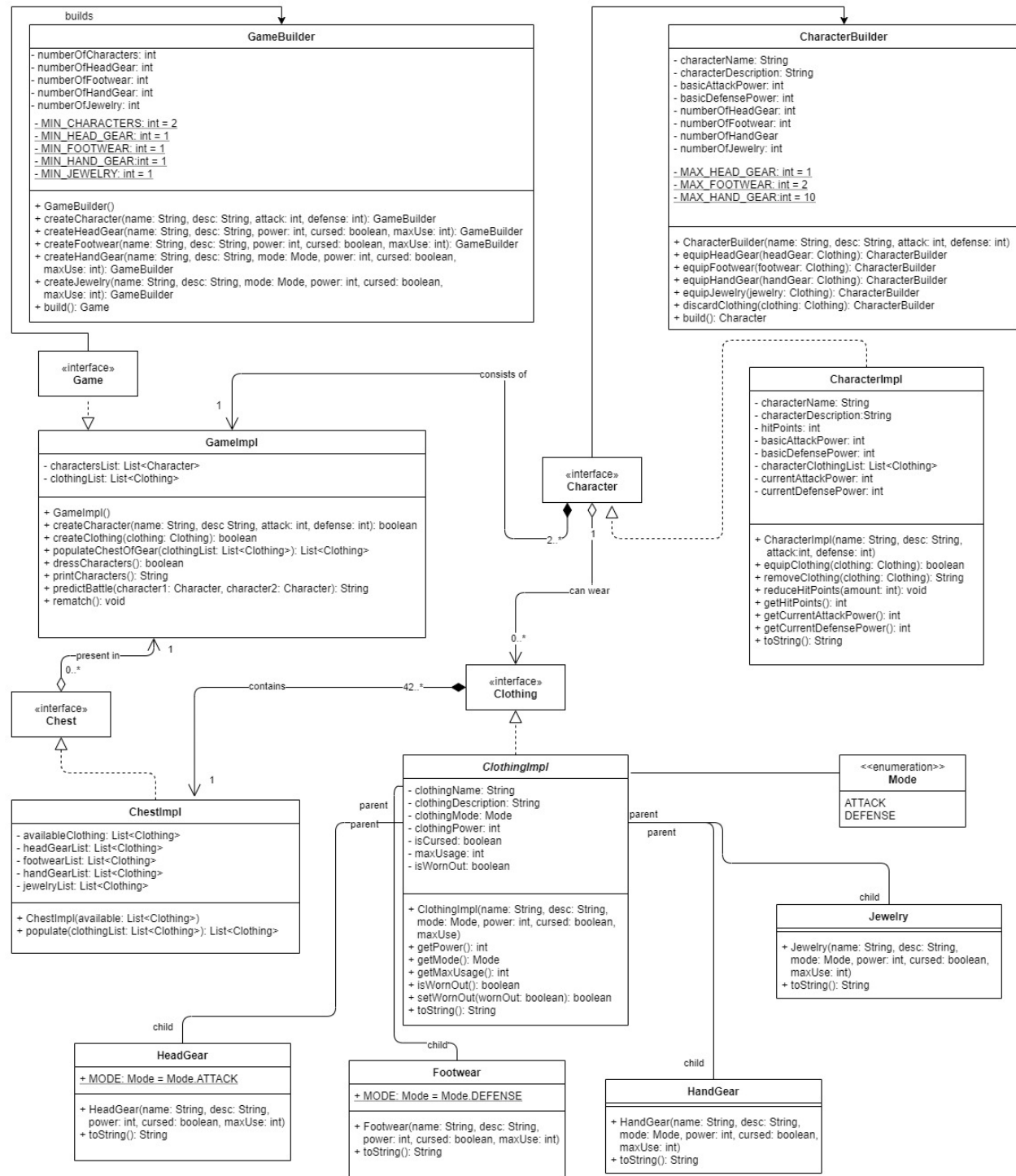


## Project 2 – Equipping Characters Design

### UML Class Diagram



## Testing Design for Equipping Characters

### Testing HeadGear class:

Test Case	Input Parameters	Expected Result
Constructor for HeadGear	Valid parameters	Instance of HeadGear
Exception for invalid name	Invalid name parameter (null or empty)	IllegalArgumentException
Exception for invalid description	Invalid description parameter (null or empty)	IllegalArgumentException
Exception for invalid power value	Invalid power parameter (negative value)	IllegalArgumentException
Exception for invalid maxUse parameter	Invalid maxUse parameter (0 or less than -1)	IllegalArgumentException
Test toString	Initialize HeadGear	String with details of the headgear
Test getPower	Initialize HeadGear with power 5	5
Test getMode	Initialize HeadGear with mode = Mode.DEFENSE	DEFENSE from Mode enum
Test getMaxUsage	Initialize HeadGear with maxUsage = 2	2

### Testing Footwear class:

Test Case	Input Parameters	Expected Result
Constructor for Footwear	Valid parameters	Instance of Footwear
Exception for invalid name	Invalid name parameter (null or empty)	IllegalArgumentException
Exception for invalid description	Invalid description parameter (null or empty)	IllegalArgumentException
Exception for invalid power value	Invalid power parameter (negative value)	IllegalArgumentException
Exception for invalid maxUse parameter	Invalid maxUse parameter (0 or less than -1)	IllegalArgumentException
Test toString	Initialize Footwear	String with details of the headgear
Test getPower	Initialize Footwear with power 5	5
Test getMode	Initialize Footwear with mode = Mode.ATTACK	ATTACK from Mode enum

Test getMaxUsage	Initialize Footwear with maxUsage = 2	2
------------------	--	---

#### Testing HandGear class:

Test Case	Input Parameters	Expected Result
Constructor for HandGear	Valid parameters	Instance of HandGear
Exception for invalid name	Invalid name parameter (null or empty)	IllegalArgumentException
Exception for invalid description	Invalid description parameter (null or empty)	IllegalArgumentException
Exception for invalid mode value	Invalid mode parameter (null or not in Mode.ATTACK or Mode.DEFENSE)	IllegalArgumentException
Exception for invalid power value	Invalid power parameter (negative value)	IllegalArgumentException
Exception for invalid maxUse parameter	Invalid maxUse parameter (0 or less than -1)	IllegalArgumentException
Test toString	Initialize HandGear	String with details of the headgear
Test getPower	Initialize HandGear with power 5	5
Test getMode	Initialize HandGear with mode = Mode.ATTACK	ATTACK from Mode enum
Test getMaxUsage	Initialize HandGear with maxUsage = 2	2

#### Testing Jewelry class:

Test Case	Input Parameters	Expected Result
Constructor for Jewelry	Valid parameters	Instance of Jewelry
Exception for invalid name	Invalid name parameter (null or empty)	IllegalArgumentException
Exception for invalid description	Invalid description parameter (null or empty)	IllegalArgumentException
Exception for invalid mode value	Invalid mode parameter (null or not in Mode.ATTACK or Mode.DEFENSE)	IllegalArgumentException
Exception for invalid power value	Invalid power parameter (negative value)	IllegalArgumentException
Exception for invalid maxUse parameter	Invalid maxUse parameter (0 or less than -1)	IllegalArgumentException

Test toString	Initialize Jewelry	String with details of the headgear
Test getPower	Initialize Jewelry with power 5	5
Test getMode	Initialize Jewelry with mode = Mode.ATTACK	ATTACK from Mode enum
Test getMaxUsage	Initialize Jewelry with maxUsage = 2	2

#### Testing CharacterImpl class:

Test Case	Input Parameters	Expected Result
Constructor for CharacterImpl	Valid parameters	Instance of CharacterImpl
Exception for invalid name	Invalid name parameter (null or empty)	IllegalArgumentException
Exception for invalid description	Invalid description parameter (null or empty)	IllegalArgumentException
Exception for invalid attack power value	Invalid attack power parameter (negative value)	IllegalArgumentException
Exception for invalid defense power parameter	Invalid defense power parameter (negative value)	IllegalArgumentException
Test toString	Initialize CharacterImpl	String with details of the character
Test equipClothing	Valid clothing	Add to clothing list and modified hitpoints, current attack power, current defense power
Test removeClothing	equipClothing	Empty clothing list
Test reduceHitPoints	Initialize CharacterImpl with 50 attack, 50 defense reduceHitPoints(10)	getHitPoints = 40
Test getHitPoints	Initialize CharacterImpl with 50 attack, 50 defense	100
Test getCurrentAttackPower	Initialize CharacterImpl with 50 attack, 50 defense	50
Test getCurrentDefensePower	Initialize CharacterImpl with 50 attack, 50 defense	50

#### Testing CharacterBuilder class:

Test Case	Input Parameters	Expected Result
-----------	------------------	-----------------

Constructor for CharacterBuilder	Valid parameters	Instance of CharacterBuilder
Exception for invalid name	Invalid name parameter (null or empty)	IllegalArgumentException
Exception for invalid description	Invalid description parameter (null or empty)	IllegalArgumentException
Exception for invalid attack power value	Invalid attack power parameter (negative value)	IllegalArgumentException
Exception for invalid defense power parameter	Invalid defense power parameter (negative value)	IllegalArgumentException
Test equipHeadGear	Valid HeadGear	Added HeadGear into clothing list of character
Test equipHeadGear with existing headgear	Valid HeadGear	IllegalStateException
Test equipFootwear	Valid Footwear	Added Footwear into clothing list of character
Test equipFootwear with existing 2 footwear	Valid Footwear	IllegalStateException
Test equipHandGear	Valid HandGear	Added HandGear into clothing list of character
Test equipHandGear with existing headgear	Valid HandGear	IllegalStateException
Test equipJewelry	Valid Jewelry	Added Jewelry into clothing list of character
Test equipJewelry with existing headgear	Valid Jewelry	IllegalStateException
Test discardClothing	Equip clothing	Remove from clothing list of character
Test discardClothing with no clothing		IllegalStateException
Test build	Valid input	Instance of Character
Test build with no headgear	Valid input except no equipHeadGear	Instance of Character
Test build failed with 2 headgear	Valid input and equipHeadGear equipHeadGear	IllegalStateException
Test build with 1 footwear	equipFootwear	Instance of Character
Test build with no footwear	Valid input except no equipFootwear	Instance of Character
Test build failed with 3 footwear	Valid input and equipFootwear equipFootwear equipFootwear	IllegalStateException

Test build with 1 handgear	equipHandGear	Instance of Character
Test build with no handgear	Valid input except no equipHandGear	Instance of Character
Test build failed with 11 handgear	Valid input and equipHandgear 11 times	IllegalStateException
Test build with 1 jewelry	equipJewelry	Instance of Character
Test build with no handgear	Valid input except no equipJewelry	Instance of Character
Test build with 11 footwear	equipJewelry 11 times	Instance of Character
Test build with no clothing and discard	discardClothing	IllegalStateException

#### Testing ChestImpl class:

Test Case	Input Parameters	Expected Result
Constructor for ChestImpl	Valid parameters	Instance of ChestImpl
Exception for empty list	Empty list	IllegalArgumentException
Test populate	Valid parameters	List of Clothing objects with minimum of 4 headgear, 8 footwear, 15 handgear and 15 jewelry

#### Testing GameBuilder class:

Test Case	Input Parameters	Expected Result
Constructor for GameBuilder		Instance of GameBuilder
Test createCharacter	Valid parameters	Instance of GameBuilder
Test createCharacter with invalid name	Invalid name	IllegalArgumentException
Test createCharacter with invalid description	Invalid description	IllegalArgumentException
Test createCharacter with invalid attack	Invalid attack (negative value)	IllegalArgumentException
Test createCharacter with invalid defense	Invalid defense (negative value)	IllegalArgumentException
Test createHeadGear	Valid parameters	Instance of GameBuilder
Test createHeadGear with invalid name	Invalid name	IllegalArgumentException
Test createHeadGear with invalid description	Invalid description	IllegalArgumentException
Test createHeadGear with invalid power	Invalid power (negative value)	IllegalArgumentException

Test createHeadGear with invalid maxUse	Invalid maxUse (0 or less than -1)	IllegalArgumentException
Test createFootwear	Valid parameters	Instance of GameBuilder
Test createFootwear with invalid name	Invalid name	IllegalArgumentException
Test createFootwear with invalid description	Invalid description	IllegalArgumentException
Test createFootwear with invalid power	Invalid power (negative value)	IllegalArgumentException
Test createFootwear with invalid maxUse	Invalid maxUse (0 or less than -1)	IllegalArgumentException
Test createHandGear	Valid parameters	Instance of GameBuilder
Test createHandGear with invalid name	Invalid name	IllegalArgumentException
Test createHandGear with invalid description	Invalid description	IllegalArgumentException
Test createHandGear with invalid mode	Invalid mode value (null or other than Mode.ATTACK or Mode.DEFENSE)	IllegalArgumentException
Test createHandGear with invalid power	Invalid power (negative value)	IllegalArgumentException
Test createHandGear with invalid maxUse	Invalid maxUse (0 or less than -1)	IllegalArgumentException
Test createJewelry	Valid parameters	Instance of GameBuilder
Test createJewelry with invalid name	Invalid name	IllegalArgumentException
Test createJewelry with invalid description	Invalid description	IllegalArgumentException
Test createJewelry with invalid mode	Invalid mode value (null or other than Mode.ATTACK or Mode.DEFENSE)	IllegalArgumentException
Test createJewelry with invalid power	Invalid power (negative value)	IllegalArgumentException
Test createJewelry with invalid maxUse	Invalid maxUse (0 or less than -1)	IllegalArgumentException
Test build	Valid input	Instance of Game
Test build with no characters	Valid input with no characters	IllegalStateException
Test build with 1 character	Valid input with 1 character	IllegalStateException
Test build with no headgear	Valid input with no headgear	IllegalStateException
Test build with no footwear	Valid input with no footwear	IllegalStateException
Test build with no handgear	Valid input with no handgear	IllegalStateException
Test build with no jewelry	Valid input with no jewelry	IllegalStateException

**Testing GameImpl class:**

Test Case	Input Parameters	Expected Result
Constructor for GameImpl	Valid input	Instance of GameImpl
Test createCharacter	Valid parameters	True
Test createCharacter with invalid name	Invalid name	IllegalArgumentException
Test createCharacter with invalid description	Invalid description	IllegalArgumentException
Test createCharacter with invalid attack	Invalid attack (negative value)	IllegalArgumentException
Test createCharacter with invalid defense	Invalid defense (negative value)	IllegalArgumentException
Test createClothing	Valid parameters	True
Test createClothing with invalid name	Invalid name	IllegalArgumentException
Test createClothing with invalid description	Invalid description	IllegalArgumentException
Test createClothing with invalid mode	Invalid mode value (null or other than Mode.ATTACK or Mode.DEFENSE)	IllegalArgumentException
Test createClothing with invalid power	Invalid power (negative value)	IllegalArgumentException
Test createClothing with invalid maxUse	Invalid maxUse (0 or less than -1)	IllegalArgumentException
Test populateChestOfGear	Valid clothing list	List of Clothing objects with atleast 4 headgear, 8 footwear, 15 handgear and 15 jewelry
Test populateChestOfGear with invalid input	Empty list	IllegalArgumentException
Test dressCharacters		True (dresses characters with 20 random items)
Test printCharacters		String with details of characters with clothing details and attack and defense power
Test predictBattle	Valid input with 2 dressed characters	String output of battle prediction
Test predictBattle with character 1 winning	Input such that character 1 has more hitpoints	String output of battle prediction with character 1 winning



Test predictBattle with character 2 winning	Input such that character 2 has more hitpoints	String output of battle prediction with character 2 winning
Test predictBattle with characters using cursed items	Input such that characters use cursed items	String output of battle prediction
Test predictBattle with characters using worn out items	Input such that characters use items worn out and see reduction in hitpoints after every round	String output of battle prediction
Test predictBattle with one character having more attack power	Input such that character 1 has more attack power	String output of battle prediction
Test predictBattle with one character having more defense power	Input such that character 1 has more defense power	String output of battle prediction
Test predictBattle with characters using no cursed items	Input such that characters do not equip cursed items	String output of battle prediction
Test predictBattle with characters using no wearing out items	Input such that characters do not equip items that can be worn out	String output of battle prediction