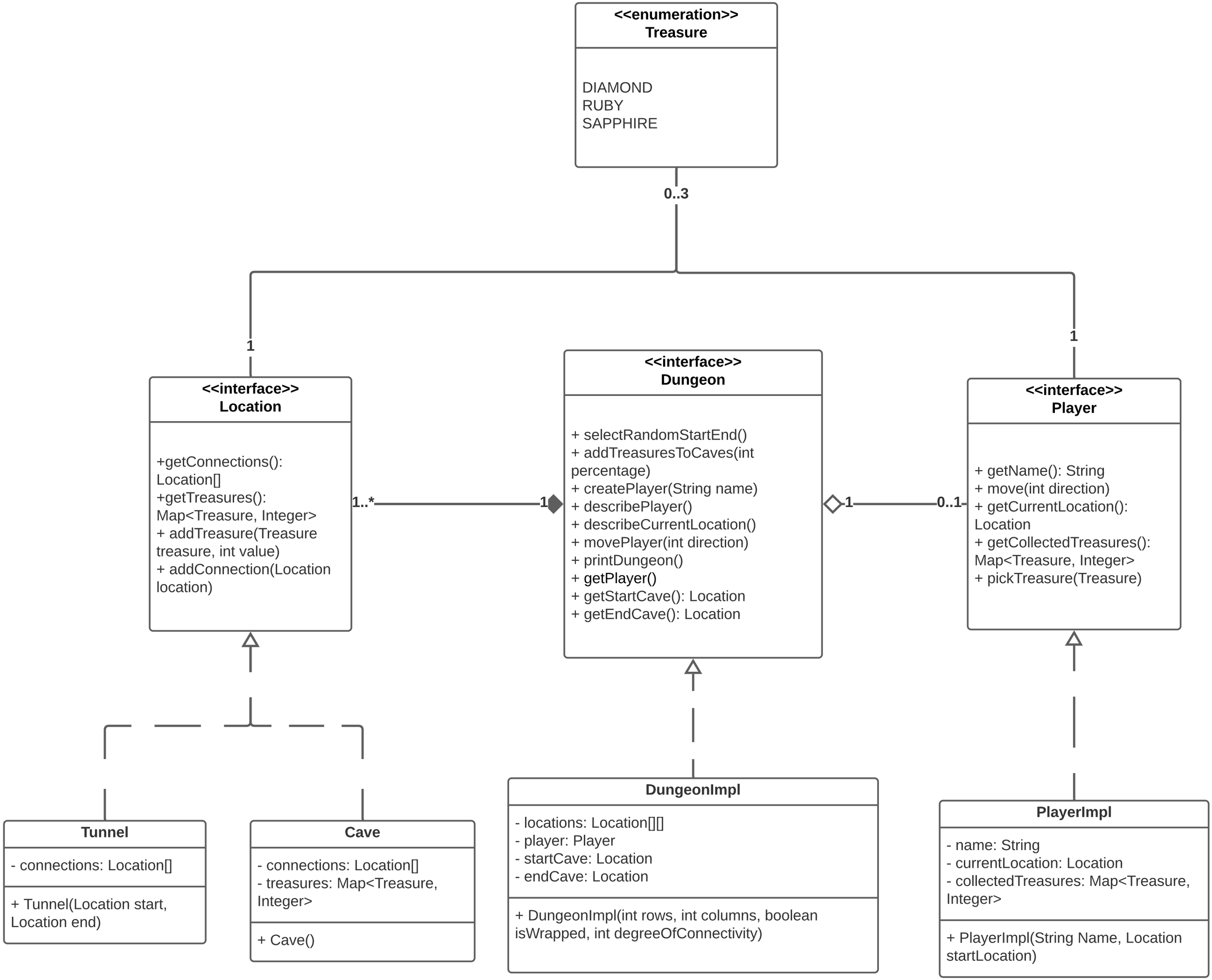
<<enumeration>>
Treasure
DIAMOND
RUBY
SAPPHIRE
0..3
1
1
<<interface>>
Location
+getConnections():
Location[]
+getTreasures():
Map<Treasure, Integer>
+ addTreasure(Treasure
treasure, int value)
+ addConnection(Location
location)
1..*
1
<<interface>>
Dungeon
+ selectRandomStartEnd()
+ addTreasuresToCaves(int
percentage)
+ createPlayer(String name)
+ describePlayer()
+ describeCurrentLocation()
+ movePlayer(int direction)
+ printDungeon()
+ getPlayer()
+ getStartCave(): Location
+ getEndCave(): Location
1
0..1
<<interface>>
Player
+ getName(): String
+ move(int direction)
+ getCurrentLocation():
Location
+ getCollectedTreasures():
Map<Treasure, Integer>
+ pickTreasure(Treasure)
Tunnel
- connections: Location[]
+ Tunnel(Location start,
Location end)
Cave
- connections: Location[]
- treasures: Map<Treasure,
Integer>
+ Cave()
DungeonImpl
- locations: Location[][]
- player: Player
- startCave: Location
- endCave: Location
+ DungeonImpl(int rows, int columns, boolean
isWrapped, int degreeOfConnectivity)
PlayerImpl
- name: String
- currentLocation: Location
- collectedTreasures: Map<Treasure,
Integer>
+ PlayerImpl(String Name, Location
startLocation)

# Testing Plan

**Testing Plan for DungeonImpl:**

| | **Input** | **Output** |
|---|---|---|
| **Testing Constructor** | DungeonImpl(int rows, int columns, boolean isWrapped, int degreeOfConnectivity) | |
| rows < 0 | DungeonImpl(-5, 6, true, 8) | IllegalArgumentException |
| rows = 0 | DungeonImpl(0, 6, true, 8) | IllegalArgumentException |
| columns < 0 | DungeonImpl(5, -6, true, 8) | IllegalArgumentException |
| columns = 0 | DungeonImpl(5, 0, true, 8) | IllegalArgumentException |
| degreeOfConnectivity < 0 | DungeonImpl(5, 6, true, -8) | IllegalArgumentException |
| Normal Scenario | DungeonImpl(5, 6, true, 8) | printDungeon() to print a valid randomly generated dungeon |
| | | |
| **Testing: selectRandomStartEnd()** | DungeonImpl(5, 6, true, 8); selectRandomStartEnd(); | Distance between getStartCave() & getEndCave >=5 |
| | | |
| **Testing: addTreasuresToCaves()** | addTreasuresToCaves(int percentage) | |
| Normal Scenario | addTreasuresToCaves(20) | 20% of caves should have Treasure |
| percentage < 0 | addTreasuresToCaves(-5) | IllegalArgumentException |
| percentage > 100 | **addTreasuresToCaves(150)** | IllegalArgumentException |
| | | |
| **Testing: createPlayer** | createPlayer(String name) | |
| Normal Scenario | createPlayer(“Player1”) | getPlayer() should return the player with Name: Player1 and current location as getStartCave(); |
| name = null | createPlayer(null) | IllegalArgumentException |
| | | |
| **Testing: describePlayer** | createPlayer(“Player1”);<br>…<br>describePlayer(); | Name: Player1<br>Treasures:<br>Diamonds: 20<br>Rubies: 15<br>Sapphires: 0 |
| | | |

| | | |
|---|---|---|
| **Testing: describeCurrentLocation()** | describeCurrentLocation() | Current Location:<br>Treasures:<br>Diamonds: 0<br>Rubies: 5<br>Sapphires: 5<br>Possible directions:<br>NORTH<br>SOUTH<br>WEST |
| | | |
| **Testing: movePlayer()** | movePlayer(int direction) | |
| Normal Scenario | movePlayer(3) | Player moves depending on the input:<br>0 – North<br>1 – East<br>2 – South<br>3 – West |
| direction <= 0 | movePlayer(0) | IllegalArgumentException |
| direction > 4 | movePlayer(4) | IllegalArgumentException |
| | | |
| **Testing: printDungeon()** | printDungeon() | Print the dungeon |

**Testing Plan for PlayerImpl:**

| | **Input** | **Output** |
|---|---|---|
| **Testing Constructor** | PlayerImpl(String Name, Location startLocation) | |
| Normal Scenario | PlayerImpl (“Player1”, startLocation) | A player with name as Player1 and startLocation as the mentioned location |
| name = null | PlayerImpl(null, startLocation) | IllegalArgumentException |
| startLocation = null | PlayerImpl (“Player1”, null) | IllegalArgumentException |
| | | |
| **Testing: getName()** | PlayerImpl (“Player1”, startLocation);<br>getName(); | “Player1” |
| | | |
| **Testing: move ()** | move (int direction) | |
| Normal Scenario | move (3) | Player moves depending on the input:<br>0 – North<br>1 – East<br>2 – South |

| | | |
|---|---|---|
| | | 3 – West |
| direction <= 0 | move (0) | IllegalArgumentException |
| direction > 4 | move (4) | IllegalArgumentException |
| | | |
| **Testing: getCurrentLocation()** | getCurrentLocation() | Current Location of Player. |
| | | |
| **Testing: getCollectedTreasures()** | getCollectedTreasures() | Treasures:<br>Diamonds: 20<br>Rubies: 15<br>Sapphires: 0 |
| | | |
| **Testing: pickTreasure()** | pickTreasure(treasure) | |
| If CurrentLocation is a Cave and CurrentLocation has 5 Diamonds | pickTreasure(Treasure.DIAMOND) | Treasures:<br>Diamonds: 25<br>Rubies: 15<br>Sapphires: 0 |
| If CurrentLocation is a Tunnel | pickTreasure(Treasure.DIAMOND) | IllegalStateException |
| If currentLocation does not have diamonds | pickTreasure(Treasure.DIAMOND) | IllegalStateException |
| Same for RUBY, SAPPHIRE | | |

**Testing Plan for Locations:**

| | **Input** | **Output** |
|---|---|---|
| **Testing: constructor** | Cave(),Tunnel(Location start, Location end) | |
| Normal Scenario | Cave() | Creates a new cave with no connections |
| Normal Scenario | Tunnel(start, end) | Creates a tunnel connecting start and end |
| Start and end are not neighbors | Tunnel(start, end) | IllegalStateException |
| | | |
| **Testing: addConnection** | addConnection(Location location) | |
| In Cave | cave.addConnection(location) | Adds a new connection |
| In Tunnel | tunnel. addConnection(location) | InvalidOperationException |
| | | |
| **Testing: getTreasures()** | getTreasures() | |

| | | |
|---|---|---|
| In Cave | cave.getTreasures() | Treasures:<br>Diamonds: 0<br>Rubies: 5<br>Sapphires: 5 |
| In Tunnel | tunnel.getTreasures() | InvalidOperationException |
| | | |
| **Testing: addTreasure()** | addTreasure(Treasure treasure, int value) | |
| In Cave | cave.addTreasure(Treasure.DIAMOND, 10) | Treasures:<br>Diamonds: 10<br>Rubies: 5<br>Sapphires: 5 |
| In Tunnel | tunnel. addTreasure(Treasure.DIAMOND, 10) | InvalidOperationException |
| In cave, Treasure is null | cave.addTreasure(null, 10) | IllegalArgumentException |
| In cave, value<=0 | cave.addTreasure(Treasure.DIAMOND, -10) | IllegalArgumentException |