Jose A. Hernandez

Class Structure and Flowchart for Assignment 3

Panther ID: 5712864

28OCT20

public class App{

Make a new object of Game and call the play method

}

public class Game{

public method called play that expects no arguments{

Will hold a do/while loop that will run forever until the user picks up an item. The do/while loop will also hold the basic logic that allows the player to move around in the house and discover new items.

}

Private method called input that returns a string and expects 1 string argument{

Will create a JOptionPane asking the user for an input

}

Private method called output that expects 1 string argument{

Will create a JOptionPane showing the user the corresponding text for the action they just completed

}

Private method called input that returns an object and expects 1 string argument and 1 object array argument{

Will create a JOptionPane showing the user all their current options for the room and will return an object type as the input from the user

}

Private method called map that expects 1 string argument{

Will create a JOptionPane showing the user their current location on a map

}

}

public class Player{

private playerName ← String

private INV\_SIZE ← 4

private invContents ← String[ INV\_SIZE ]

private currentRoom ← Room

private house ← RoomActions

public constructor that expects 1 string argument{

playerName ← value of the string parameter

currentRoom ← “entrance”

invContents ← null

}

Public method called getLocation that returns a string{

Returns the player’s current location

}

Public method called getName that returns a string{

Returns the player’s name

}

Public method called connectedRoom that returns a string{

Returns the rooms that are connected to the current room

}

Public method called pickupItem that returns a string and expects 1 string argument{

Checks if the argument being passed is an item that exists in the room. Also checks to see if the player already has the item or not. If not, it will add the item to the backpack.

}

Public method called moveTo that returns a string and expects 1 string argument{

Checks to see if the string argument is a connected room to the player’s current room. If it is, then the player will be moved to that room.

}

Public method called inspectItem that returns a string and expects 1 string argument{

Checks if the argument being passed is an item that exists in the room. If it is, then it will return the item’s description.

}

}

public class Room{

private roomName ← String

private connectedRooms ← ArrayList of Strings

private itemsInRoom ← hashMap of String keys and String values

Create 3 constructors that take 0, 1, and 2 arguments, and each constructor will take a String and String array input to set roomName and connectedRooms to the arguments.

Public method called addItem that takes 2 string arguments{

Adds the parameters as a Key:Value set to itemsInRoom where the key is the name of the item and the value is the description for the item

}

Public method called getConnectedRooms that returns a list of all connected rooms as strings

Public method called getRoomName that returns roomName as a string

Public method called getItemNames that returns a string array of the keyset for itemsInRoom

Public method called getItemDesc that will take 1 string argument and return a string{

Will return the item description for the item that is passed in the parameter

}

}

public class RoomActions{

private rooms ← ArrayList of Room

private AMOUNT\_ROOMS ← 13

private counter ← int

create one constructor that will call generateRooms()

public method called generateRooms{

This method is used to create all the rooms as a Room object. All the connected rooms are passed in a string array for each room. Items are all added using the addItem() method from the Room class

}

}