|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test | Description | Inputs | Expected Outputs | Actual Outputs | Test Creation Date | Build/Version | Pass/Fail | Outcome Description |
| 1 | Creation of a property object with the description, price and property level | Description: Robbie’s Home  Price: 250  Rents: 20, 40, 60, 80, 100, 120 | Getter methods to return correct description, price and correct rent for each property level | The expected outputs |  |  | PASS | Property objects are properly created |
| 2 | Should receive an error when property is created with invalid parameters | Description: null  Price: -250  Rents: null | Illegal argument exception | Illegal argument exception |  |  | PASS | Error correctly given for improper parameters |
| 3 | Creation of a station with a description and price | Description: Matt’s Rail  Price: 200 | Getter methods to return correct description and price for the station | The expected outputs |  |  | PASS | Stations are properly created |
| 4 | Should receive an error when station is created with invalid parameters | Description: null  Price: -200 | Illegal argument exception | Illegal argument exception |  |  | PASS | Error correctly given for improper parameters |
| 5 | Creation of a utility with a description and price | Description: Ollie’s Electrics | Getter methods to return correct description and price for the utility | The expected outputs |  |  | PASS | Utility was properly created |
| 6 | Should receive error when utility is created with invalid parameters | Description: null  Price: -200 | Illegal argument exception | Illegal argument exception |  |  | PASS | Error correctly given for improper parameters |
| 7 | Property objects that are bought should be owned by the correct player | Player object: buyer=player(jim,coral)  Call buy(buyer) from Property class | getOwner() in property should return buyer object and isOwned() should be true | Buyer object is returned and isOwned() is true |  |  | PASS | Correct player is given ownership of the property |
| 8 | Should receive an error when a null buyer object tries to buy a property | Player object: buyer=null  Call buy(buyer) from Property class | getOwner() in Property should return an error and instance isOwned() returns null | The expected results |  |  | PASS | Error correctly given for improper parameters |
| 9 | Downgrade() should return the cash amount that should be credited for and a property’s level should go down | cash=instance.downgrade();  instance.getImprovementCost();  getLevel(0); | Cash should equal getImprovementCost()  Instance.getLevel() should be 0 | The expected results |  |  | PASS | The correct amount is returned and the property level is decremented unless the min level |
| 10 | House properties should be part of a group | instance = Property.create("T's Gym", 500, new int[]{1,1,1,1,1,1});  Instance should be assigned to a new group ‘group’ using property.group.create method. | Instance.isGrouped() to return true and group equal instance.getGroup() | The expected results |  |  | PASS | The property was assigned to the correct group |
| 11 | Should be able to get cost of upgrading or downgrading a property | improve = instance.getImprovementCost();  where instance is a created property and it assigned to a group | Improve should be equal to the groups getImprovementCost method | The expected results |  |  | PASS | The correct amounts are returned |
| 12 | Should be able to get amount you would receive for mortgaging a property | mortgaged = instance.getMortgagedPrice() where instance is a property | Mortgaged should be equal to half the amount returned from instance.getPrice() | The expected results |  |  | PASS | Mortgaging works as it should |
| 13 | Should be able to get the rent price of a property | Property called instance, which is assigned to a group ‘group’, and then bought by a player and later upgraded | instance.getRentPrice(group.getLevels().getLevel(0) should return 1 and after instance.upgrade() is called, 5 | The expected result |  |  | PASS | Rent price returned works when a house is put on the property |
| 14 | Should be able to get the rent price of a station | Property called instance which is assigned to a group ‘group’ and then bought by a player. After the same player buys another station. | station1.getRentPrice(group.getLevels().getLevel(0) should return 25 and station2 should return 50 | The expected result |  |  | PASS | Works |
| 15 | Should be able to get rent price of utility, depending on dice throw and number of utilities owned by player | Two property objects utility1 and utility2, where they’re both assigned to a group and owned by the same player | Rent price should be 24 when one utility is owned and the dice throw equals 6 (4 times dice throw).  Rent price should be 70 when two utilities are owned and dice throw equals 10 (10 times dice throw). | The expected result |  |  | PASS | Works |
| 16 | A player should be able to sell a property it bought for the original amount | Property ‘instance’ where it’s owned by Player ‘buyer’ and then sold  sellPrice equals instance.sell() | sellPrice should equal the getPrice method and the property shouldn’t be owned anymore | The expected result |  |  | PASS | The correct amount is returned |
| 17 | Should be able to upgrade a property level of a property | Property ‘instance’ where its assigned to a group, bought by Player ‘buyer’ and upgraded | Instance.getLevel() should equal 1 | The expected results |  |  | PASS | Level increased from 0 to 1 |
| 18 | Should be able to set a property to a group | Property ‘instance’ which is assign to Group ‘group’ | Instance.getGroup() should equal group | The expected results |  |  | PASS | Getter returns the correct group |
| 19 | Should be able to get the current property level of a property or utility | Property ‘instance’ and ‘utility’ where they’re assigned to different groups | getLevel() call for both instance and utility should both equal 0 | The expected results |  |  | PASS | Works |
| 20 | Should be able to get the owner of a property | Property ‘instance’ which calls its buy method on a Player ‘buyer’ | Buyer should be equal to instance.getOwner() and isOwened() should be true | The expected results |  |  | PASS | Property is owned by the specified buyer |
| 21 | Should be able to get the price of a property | Property ‘instance’ which is initialised with a cost of 50 | The getPrice() method should return 50 and isMortgaged() should be false | The expected results |  |  | PASS | Returned 50 |
| 22 | Should be able to check whether a property, house or utility is grouped | Property ‘instance’, ‘utility’ and ‘station’ are all created and added to a different group | isGrouped() should be true for all Property objects | The expected results |  |  | PASS |  |
| 23 | Should be able to check whether a property, utility and station is improvable(i.e. add houses) | Property ‘instance’, ‘utility’ and ‘station’ are all created and added to a different group | isImprovable() should be true for all Property objects | The expected results |  |  | PASS |  |
| 24 | Should be able to check whether a property, house or utility is morgaged | Property ‘instance’, ‘utility’ and ‘station’ are all created and added to a different group | Ismortgaged() should be true for all Property objects | The expected results |  |  | PASS |  |
| 25 | Should be able to check whether a property is owned or not, and it should change when bought/sold | Property ‘instance’ which is bought and then sold  Player ‘buyer’ is used to buy instance | isOwned() should be false before instance.buy(buyer) if called, true after , and false when instance.sell() is called | The expected results |  |  | PASS |  |
| 26 | A player should not be able to interact with a property it has sold | Property ‘instance’ where it has been bought by a Player ‘buyer’ and later sold | Instance.isValid() should be true once instance has been bought, and false once it has been sold | The expected results |  |  | PASS |  |
| 27 | Should be able to mortgage a property once it has been bought | Property ‘instance’ where it has been bought by Player ‘buyer’ and has called mortgage() | getMortgagedPrice() should equal half getPrice() and isMortgaged() should equal true | The expected results |  |  | PASS |  |
| 28 | Should be able to un-mortgage a property once it has been mortgaged | Property ‘instance’ where its bought by Player ‘owner’, mortgaged and then un-mortgaged | Instance.getPrice() should equal 50 (given as cost parameter for instance), and isMortgaged() should return false | The expected results |  |  | PASS |  |
| 29 |  |  |  | The expected results |  |  | PASS |  |
| 30 |  |  |  | The expected results |  |  | PASS |  |
| 31 |  |  |  | The expected results |  |  | PASS |  |
| 32 |  |  |  | The expected results |  |  | PASS |  |
| 33 |  |  |  | The expected results |  |  | PASS |  |
| 34 |  |  |  | The expected results |  |  | PASS |  |
| 35 |  |  |  | The expected results |  |  |  |  |