All code can be found here:
https://github.com/Thug-Lyfe/test_ass_3

a)Equivalence classes

1) *public boolean isEven(int n)*

input	output
x%2 == 0	true
x%2 == 1	false

2) *Mortgage applicant's salary*

input	output
salary < 1.000$	not valid
1.000$ < salary < 75.000$	valid
salary > 75.000$	not valid

3) *public static int getNumDaysinMonth(int month, int year)*

input	output
month<1 || month>12	not valid
year: max Int || min Int	not valid
month:[1,3,5,7,8,10,12]	31
month:[4,6,9,11]	30
month:2 year: not leap year	28
month:2 year leap year	29

b) Boundary Analysis

1) isEven

input	output
minInt+1,-2,0,2,maxInt-1	true
minInt,-1,1,maxInt	false

2) Salary

input	output
999	not valid
1.000,1.001,74.999,75.000	valid
75.001	not valid

3) month:year (test cases for month: [0,1,2,11,12,13] and year: [leap year, not leap year]

input	output
month: 0	not valid
month: 1,12	31
month:11	30
month:13	not valid
month:2 year: not leap year	28
month:2 year leap year	29

c) Decision table

1. Business table

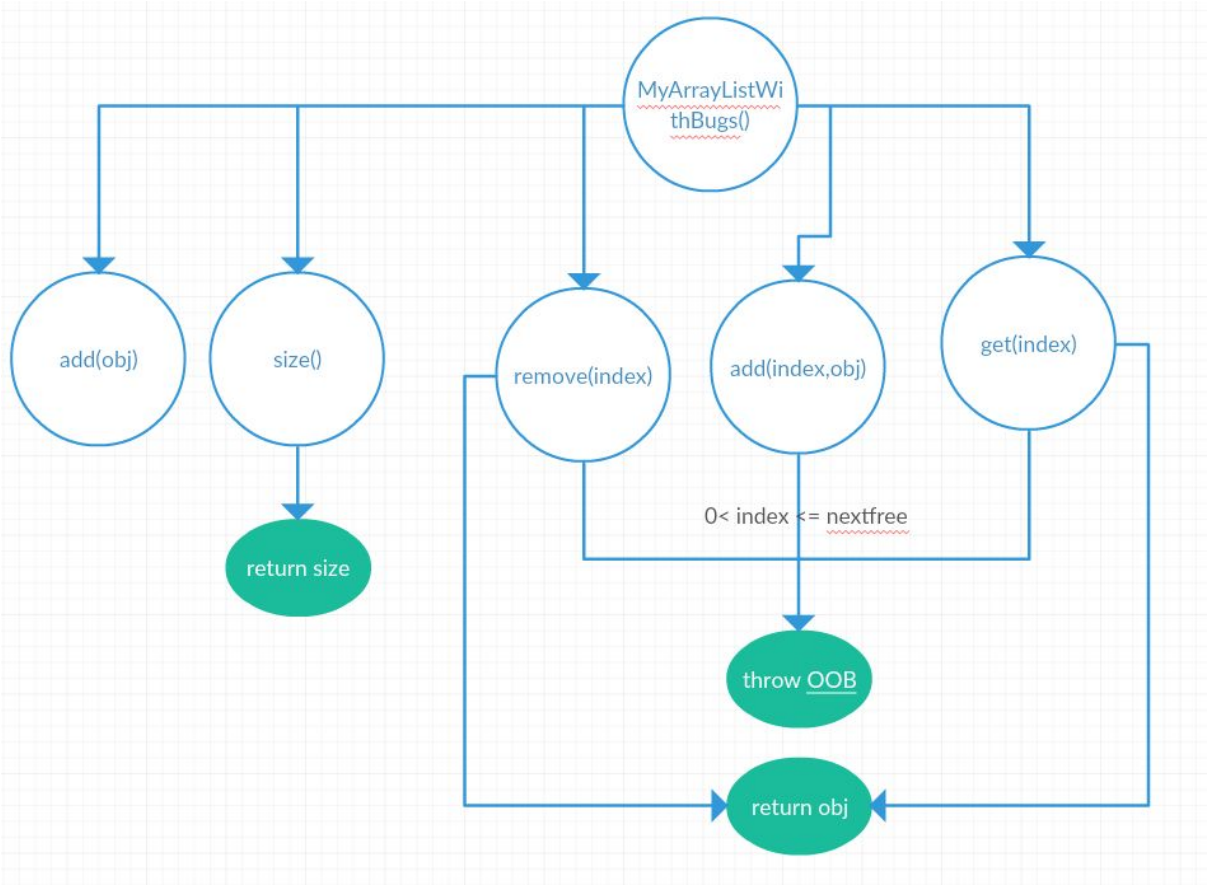
Conditions	Rule 1	Rule 2	Rule 3	Rule 4
Deductible	T	T	F	-
Doctor's office	T	-	-	F
Hospital visit	-	T	-	F
Actions				
Refunded	50%	80%	-	-
Not refunded	-	-	T	T

2. Leap year

Conditions	Rule 1	Rule 2	Rule 3
%4	-	T	-
%100	T	F	-
%400	F	-	T
Actions			
leap year		T	T
not leap year	T		

d) State transition

1.



2. Test cases

add test

steps	output
MyArrayListWithBugs()	
add(obj)	

size test

steps	output
MyArrayListWithBugs()	
add() Repeat 10 times	
size()	6

remove test

steps For an array of 5 elements it should have a boandary of 0-4, so i could test -1 == oob 0 == first element 4 == last element 5 == oob	output
MyArrayListWithBugs()	
add(obj) repeat 5 times	
remove(0)	first obj
remove(3) (since we removed an object the boundary decreased)	last obj
new test with length == 5; remove(-1)	oob
new test with length == 5; remove(5)	oob

add index test

steps For an array of 5 elements it should again have a boundary of 0-4 which we can test again.	output
MyArrayListWithBugs()	
add(obj) repeat 5 times	
add(0,obj0); get(0); size()	obj0,6
add(6,obj6);get(6);size()	obj6,7
new test with length == 5; add(-1,obj);	oob
new test with length == 5; add(5,obj)	oob

get test

steps For an array of 5 elements it should again have a boundary of 0-4 which we can test again.	output
MyArrayListWithBugs()	
add(obj) repeat 5 times	
get(0)	first obj
get(4)	last obj

new test with length == 5; get(-1)	oob
new test with length == 5; get(5)	oob

3. The junit tests can be found in the myArrayListWithBugsTest file, there is in total 11 junit tests.

4. I found 4 errors,

- The remove(index) function returned the new object at the index instead of the removed object.
- The add(index,obj) function did not shift the object already on the index, meaning it got overwritten
- The add(index,obj) function has a second error where it does not increment the nextFree variable.
- The get(index) function had a index <= 0 in the oob exception check, meaning getting 0 would result in oob regardless of there being an object there or not. It has to be in the right section instead. like this: `if(index < 0 || nextFree <= index)`

5. Not quite sure what you mean by this question.

6. All methods tested and all lines tested… so 100% code coverage.