ДОМАШНЕЕ ЗАДАНИЕ по курсу "JAVA"

Выполнил: Губенко Антон
ЗАДАНИЕ:

Необходимо написать собственную реализацию HashMap. Обязательные методы: get, put, remove.

Реализация HashMap.

Листинг класс MyHashMapImp.

```
package org.example;

import java.util.*;

public class MyHashMapImp<K,V> implements Map<K,V> {
    private static final int DEFAULT_CAPACITY=16;
    private int capacity;
    private double loadFactor=0.8;
    private int size;
    private Node [] table;

    public MyHashMapImp(){
        table=new Node[DEFAULT_CAPACITY];
        size=0;
        capacity=DEFAULT_CAPACITY;
    }
```

```java
private static class Node<K,V>{
    int hash;
    K key;
    V value;
    Node next;

    Node(int hash,K key,V value, Node next){
        this.hash=hash;
        this.key=key;
        this.value=value;
        this.next=next;
    }
}

@Override
public int size() {
    return size;
}

@Override
public boolean isEmpty() {
    return (size==0);
}
```

```java
@Override
public boolean containsKey(Object key) {
    if(size==0){
        return false;
    }
    return checkKey(key);
}

@Override
public boolean containsValue(Object value) {
    boolean result=false;
    if(size==0){
        return false;
    }
    for(int i=0;i<capacity;i++){
        if(table[i]!=null){
            result=checkValue(table[i],value);
        }
        if(result){break;}
    }
    return result;
}
```

```java
@Override
public V get(Object key) {
    if(size==0){
        return null;
    }
    int hash=getHashCode(key);
    int index=getIndex(hash);
    V resVal=null;
    if(table[index]!=null){
        resVal=getNode(index,hash,key);
    }
    return resVal;
}
```

```java
@Override
public V put(K key, V value) {
    int hash=getHashCode(key);
    int index=getIndex(hash);
    V resVal=null;

    if(table[index]==null){
        Node <K,V>newNode=new Node(hash,key,value,null);
        table[index]=newNode;
        size++;
    }
    else {
        resVal=addNode(index,hash,key,value);
    }
    resize();
    return resVal;
}
```

```java
@Override
public V remove(Object key) {
    if(size==0){
        return null;
    }
    int hash=getHashCode(key);
    int index=getIndex(hash);
    V resVal=null;
    if(table[index]!=null){
        resVal=removeNode(index,hash,key);
    }
    resize();
    return resVal;
}

@Override
public void putAll(Map<? extends K, ? extends V> m) {
Set<? extends K>keys=m.keySet();
for(K key:keys){
    V value=m.get(key);
    this.put(key,value);
}
}
```

```java
@Override
public void clear() {
for(int i=0;i<capacity;i++){
    table[i]=null;
}
size=0;
}

@Override
public Set<K> keySet() {
    if(size==0){
        return Collections.emptySet();
    }
    Set<K>keys=new HashSet<>();
    for(int i=0;i<capacity;i++){
        if(table[i]!=null){
        addKeys(table[i],keys);
        }
    }
    return keys;
}
```

```java
@Override
public Collection<V> values() {
    if(size==0){
        return Collections.emptySet();
    }
    Set<V>values=new HashSet<>();
    Set<K>keys=keySet();
    for(K key:keys){
        values.add(get(key));
    }
    return values;
}

@Override
public Set<Entry<K, V>> entrySet() {
    if(size==0){
        return Collections.emptySet();
    }
    Set<Entry<K,V>>entries=new HashSet<>();
    Set<K>keys=keySet();
    for(K key:keys){
        V value=get(key);
        Entry<K,V>entry=new AbstractMap.SimpleEntry(key,value);
        entries.add(entry);
    }
    return entries;
}
```

```java
//дополнительные утильные методы для работы мапы

private <K> int getHashCode(K key){
    return Objects.hashCode(key);
}

private int getIndex(int hash){
    return hash & (capacity-1);
}

private V addNode(int index,int hash,K key,V value){
    V resVal=null;
    Node<K, V> currentNode = table[index];
    Node<K, V> lastNode = null;

    while (currentNode != null) {
        if (currentNode.hash == hash &&
                (currentNode.key == key || (key != null && key.equals(currentNode.key)))) {
            resVal = currentNode.value;
            currentNode.value = value;
            break;
        }
        lastNode = currentNode;
        currentNode = currentNode.next;
    }
    if(resVal == null && lastNode != null && lastNode.next == null) {
        lastNode.next = new Node<>(hash, key, value, null);
        size++;
    }
    return resVal;
```

```
}

private V getNode(int index,int hash,Object key) {
    V resVal=null;
    Node<K, V> currentNode = table[index];
    while(currentNode!=null){
        if (currentNode.hash == hash &&
                (currentNode.key == key || (key != null && key.equals(currentNode.key)))){
            resVal=currentNode.value;
            break;
        }
            currentNode = currentNode.next;
    }
    return resVal;
}

private V removeNode(int index,int hash,Object key) {
    V resVal = null;
    Node<K, V> currentNode = table[index];
    Node<K, V> lastNode = null;

    while (currentNode != null) {
        if (currentNode.hash == hash &&
                (currentNode.key == key || (key != null && key.equals(currentNode.key)))) {
            resVal = currentNode.value;
            size--;
            break;
        }
        lastNode = currentNode;
        currentNode = currentNode.next;
```

```
    }
    if(resVal!=null && lastNode!=null && currentNode!=null){
    lastNode.next=currentNode.next;
    }
    else if(resVal!=null && lastNode==null){
        table[index]=currentNode.next;
    }
    return resVal;
}

private boolean checkKey(Object key) {
    int hash=getHashCode(key);
    int index=getIndex(hash);
    boolean result=false;
    if(table[index]!=null) {
        Node<K, V> currentNode = table[index];
        while (currentNode != null) {
            if (currentNode.hash == hash &&
                    (currentNode.key == key || (key != null && key.equals(currentNode.key)))) {
                result=true;
                break;
            }
            currentNode = currentNode.next;
        }
    }
    return result;
}
```

```
private boolean checkValue(Node<K,V>node,Object value){
    boolean result=false;
    while (node != null) {
        if (value == null ? node.value == null : value.equals(node.value)) {
            result=true;
            break;
        }
        node = node.next;
    }
    return result;
}

void addKeys(Node<K,V>node,Set<K>set){
    while(node!=null) {
        set.add(node.key);
        node = node.next;
    }
}

private void resize(){
    if(size>=capacity*loadFactor){
        int oldCapacity=capacity;
        int newCapacity=(int)(capacity*1.5);
        Node[]oldTable=table;
        table=new Node[newCapacity];
        capacity=newCapacity;
        addAllNodes(oldTable,oldCapacity);
    }
    else if(size<((capacity/1.5)*loadFactor-1) && capacity>16){
        int oldCapacity=capacity;
```

```
        int newCapacity=(int)(capacity/1.5);
        Node[]oldTable=table;
        table=new Node[newCapacity];
        capacity=newCapacity;
        addAllNodes(oldTable,oldCapacity);
    }
}

private void addAllNodes(Node[]oldTable,int oldCapacity){
    for(int i=0;i<oldCapacity;i++){
        if(oldTable[i]!=null){
            Node<K,V>currentNode=oldTable[i];
            while(currentNode!=null) {
                this.putResize(currentNode.key, currentNode.value);
                currentNode=currentNode.next;
            }
        }
    }
}

public void putResize(K key, V value) {
    int hash=getHashCode(key);
    int index=getIndex(hash);

    if(table[index]==null){
        Node <K,V>newNode=new Node(hash,key,value,null);
        table[index]=newNode;
    }
    else {
        addNodeResize(index,hash,key,value);
```

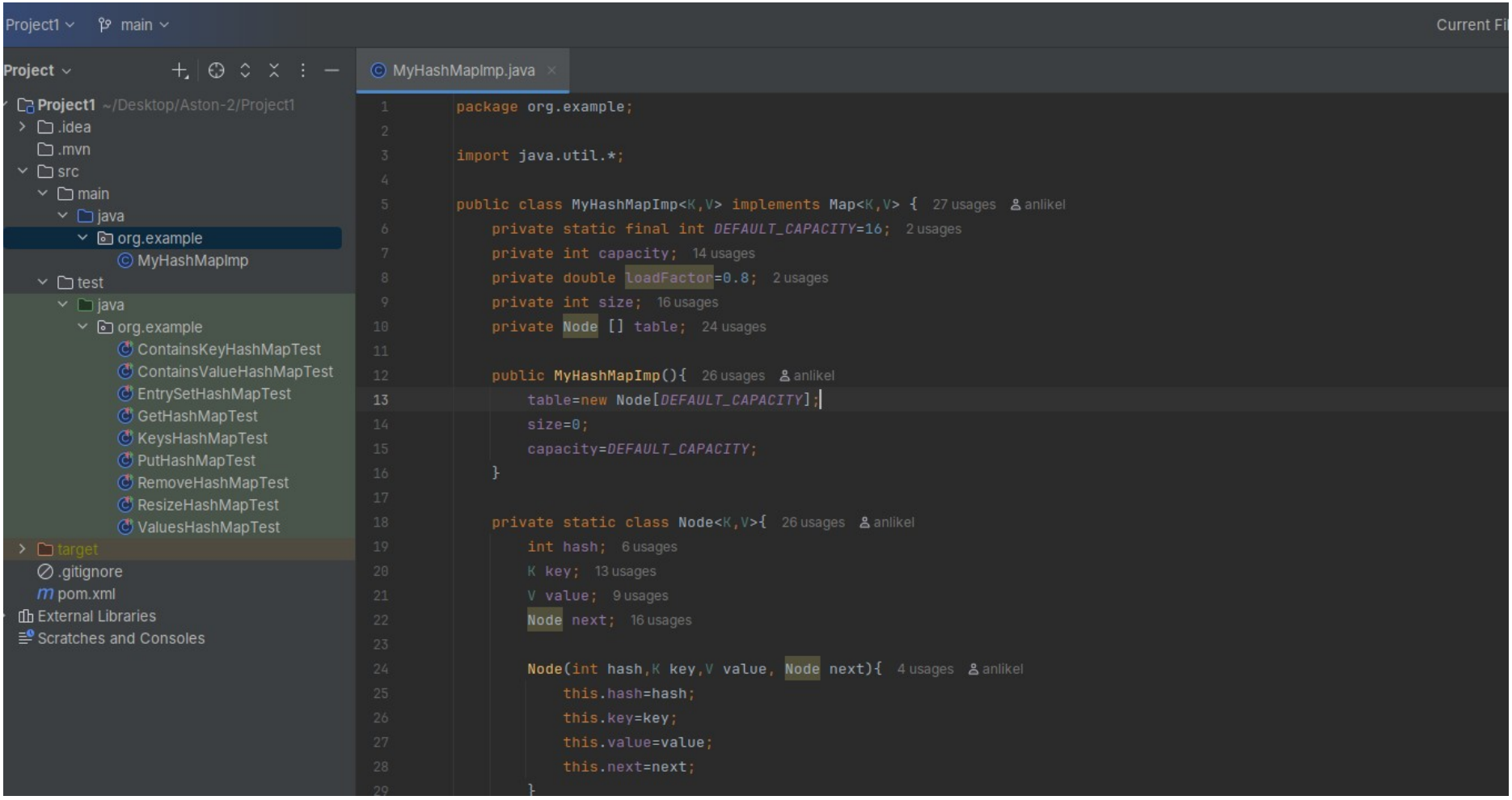
```
        }
    }

    private void addNodeResize(int index,int hash,K key,V value){
        Node<K, V> currentNode = table[index];
        Node<K, V> lastNode = null;
        int flag=0;
        while (currentNode != null) {
            if (currentNode.hash == hash &&
                    (currentNode.key == key || (key != null && key.equals(currentNode.key)))) {
                currentNode.value = value;
                flag=1;
                break;
            }
            lastNode = currentNode;
            currentNode = currentNode.next;
        }
        if(flag==0 && lastNode != null && lastNode.next == null) {
            lastNode.next = new Node<>(hash, key, value, null);
        }
    }

    //утильный метод для проверки метода resize

    public int getTableSize(){
        return table.length;
    }
}
```

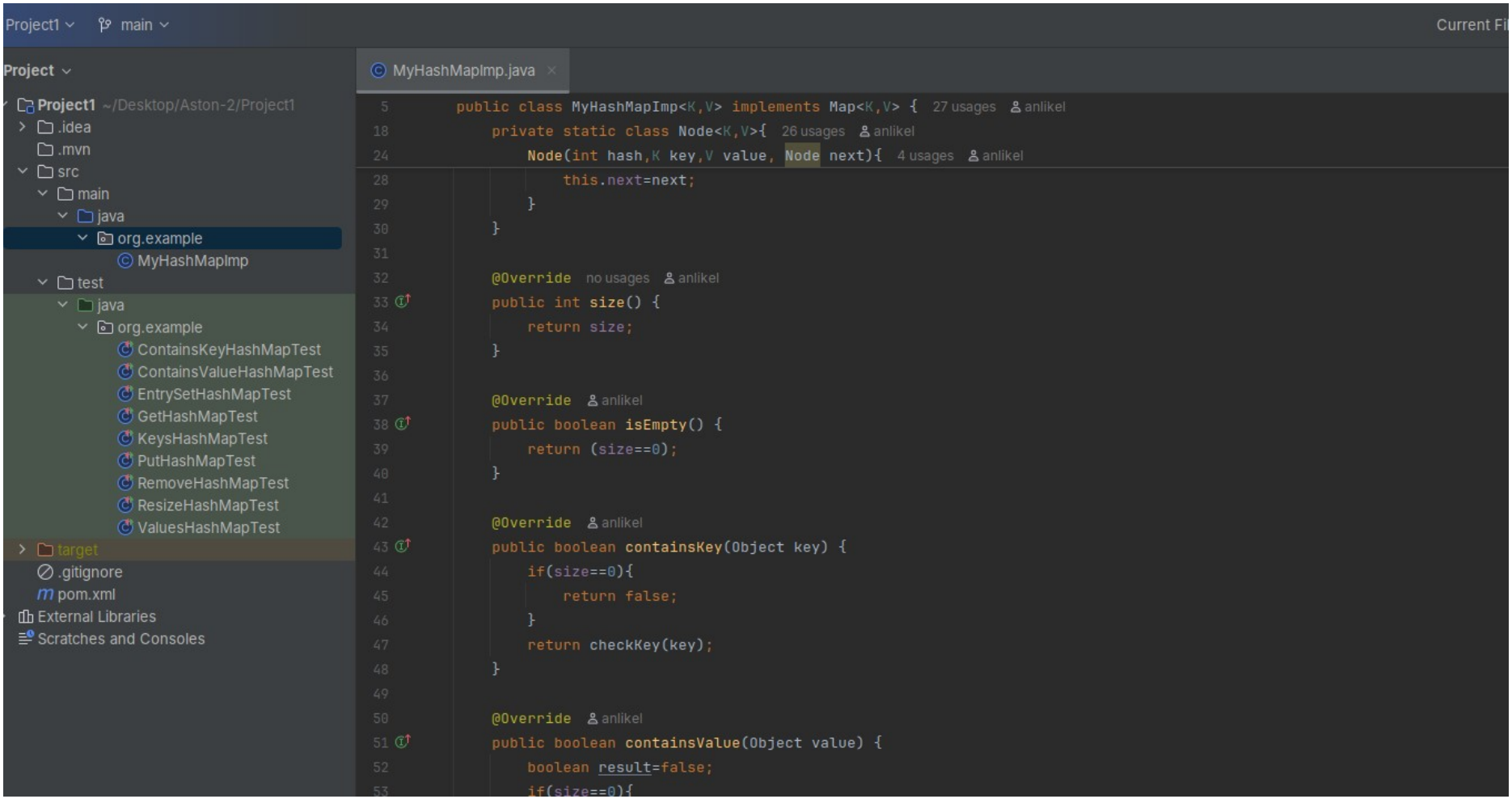
Реализованы методы:

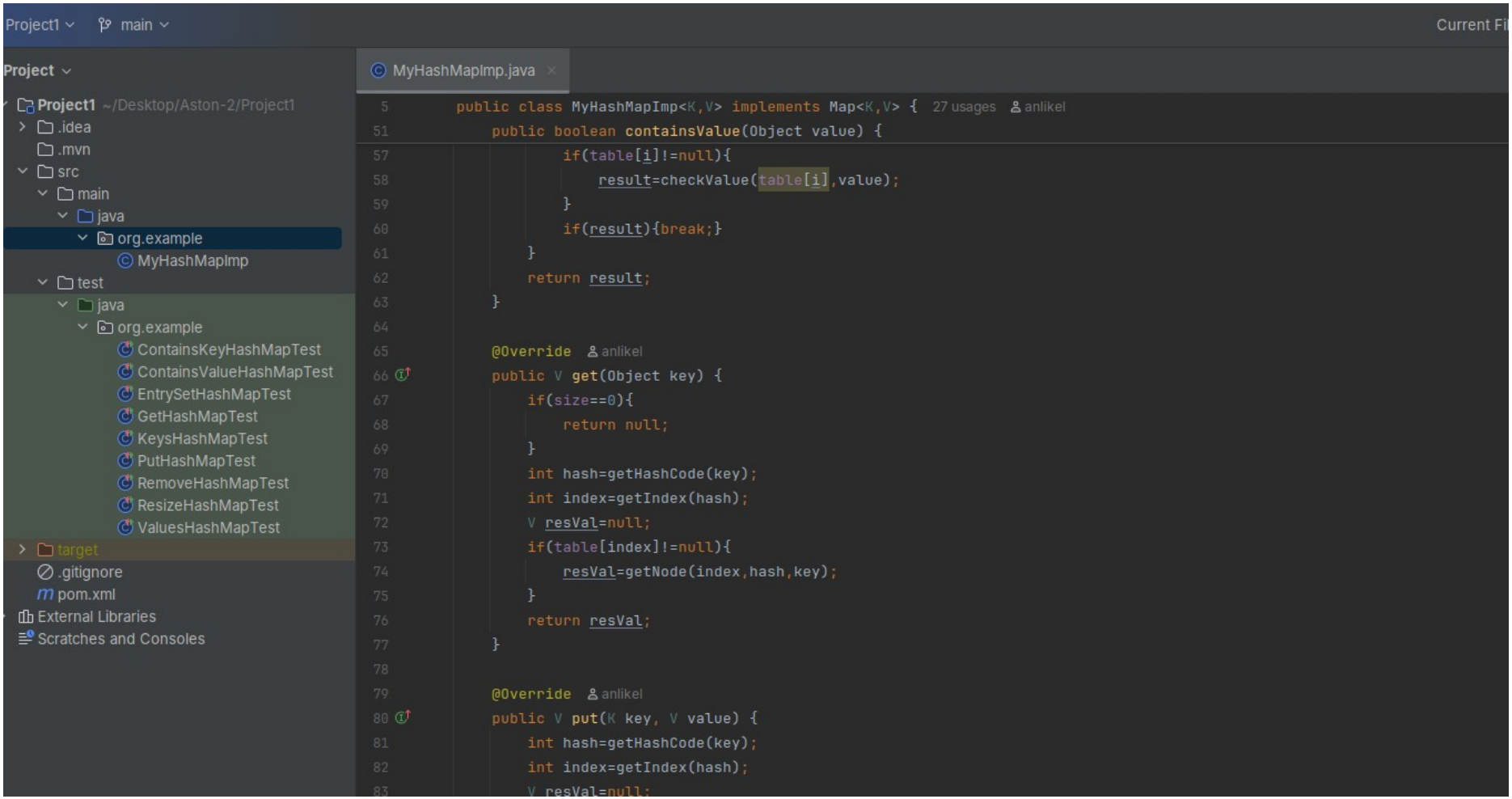
public int size(),
public boolean isEmpty(),
public boolean containsKey(Object key),
public boolean containsValue(Object value),
public V get(Objet key),
public V put(K key,V value),
public V remove(Object key),
public void putAll(Map<? Extends K,? extends V>m),
public void clear(),
public Set<K>keySet(),
public Collection<V>values(),
public Set<Entry<K,V>>entrySet().

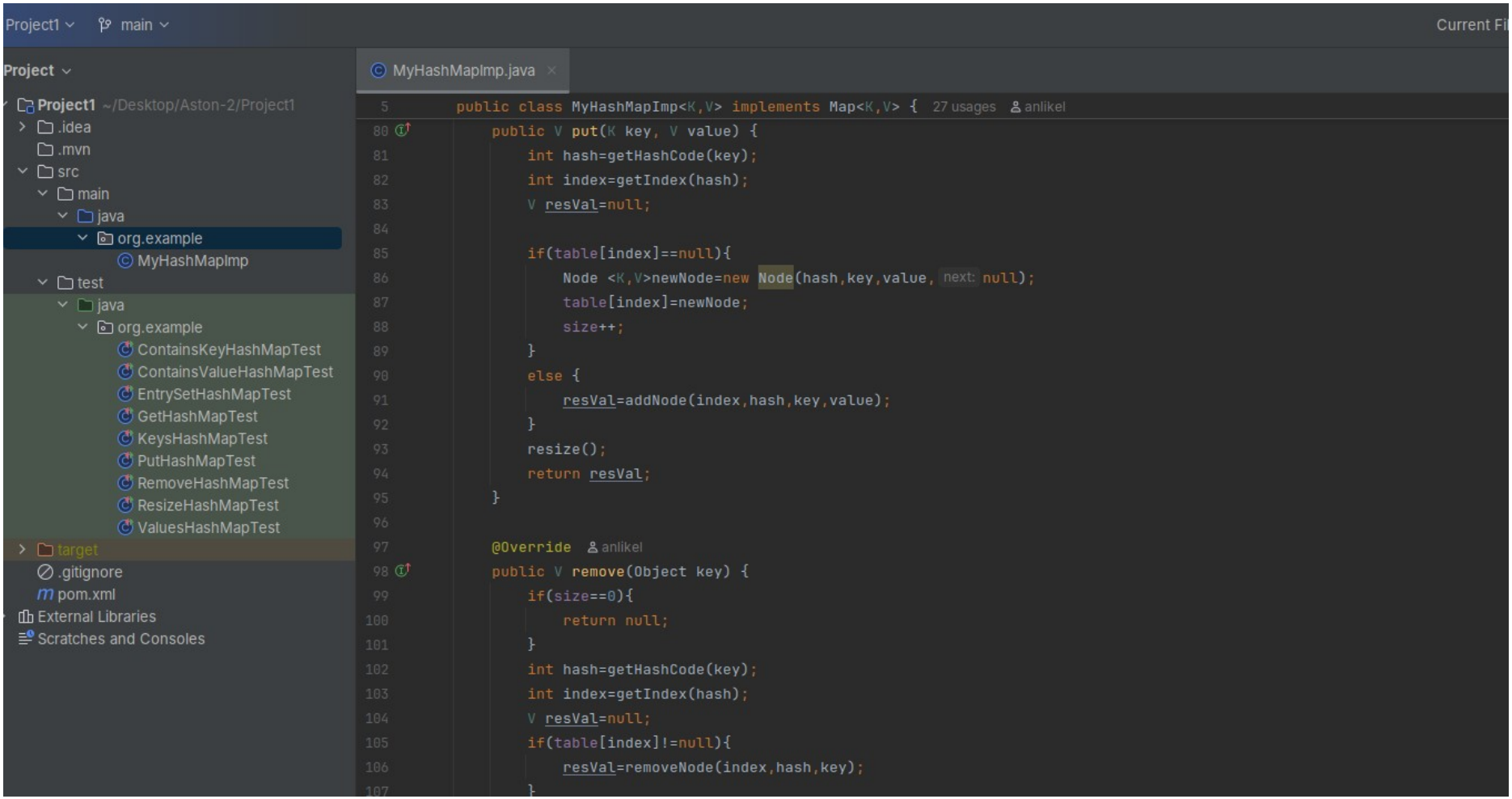
Так же был реализован внутренний статический класс
private static class Node<K,V> для хранения элементов коллекции.

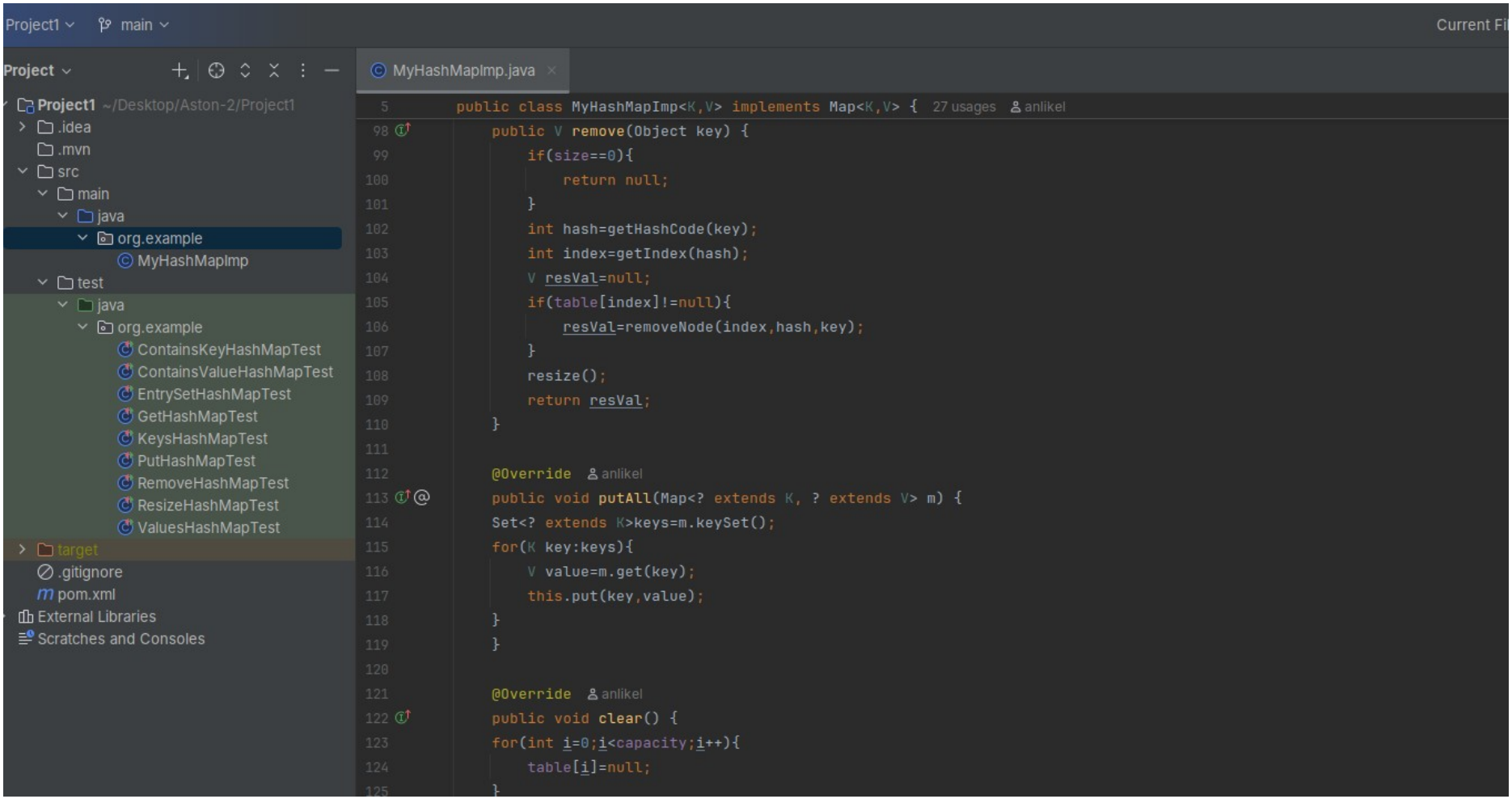
Скриншоты MyHashMapImp;

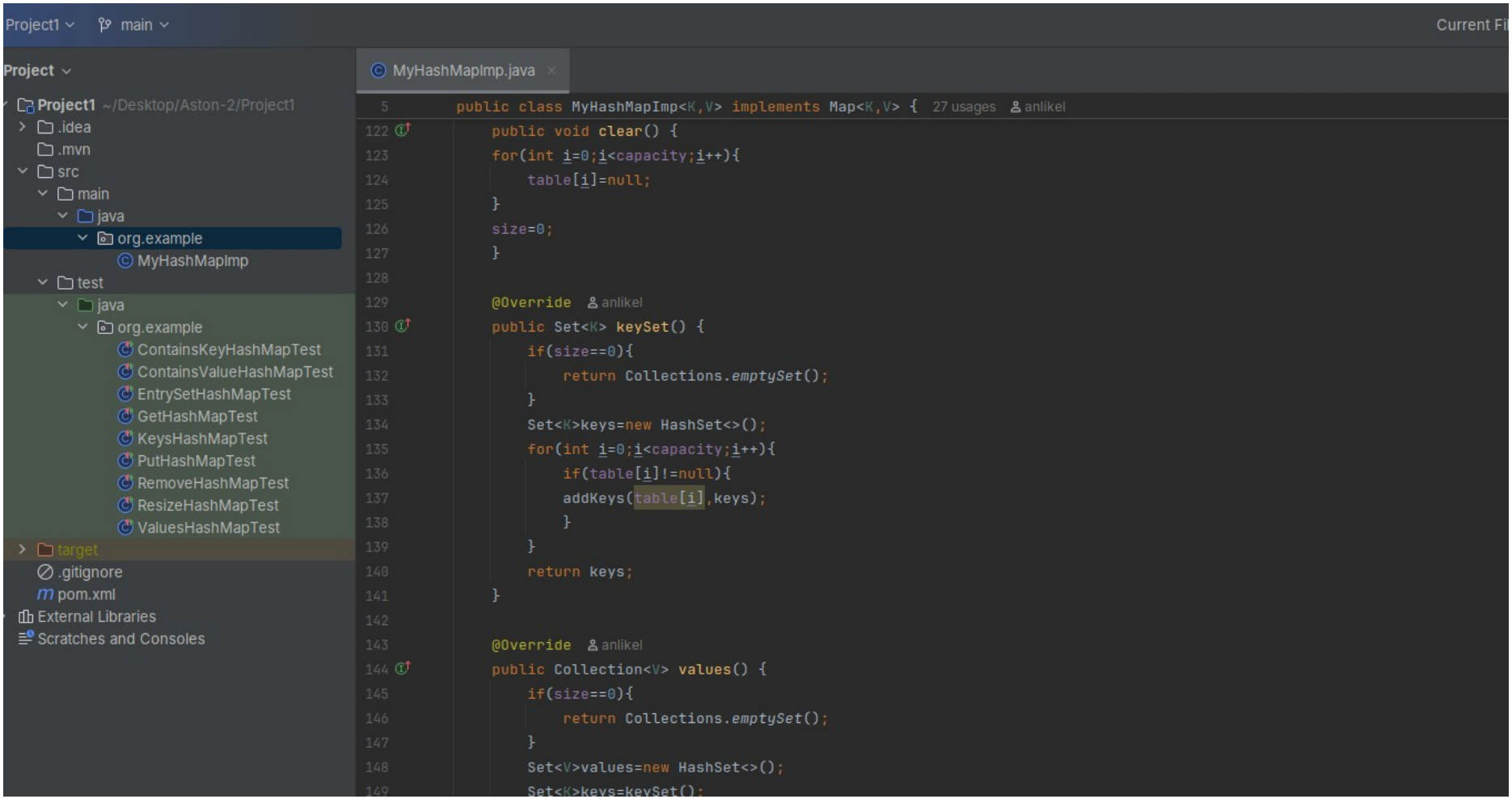


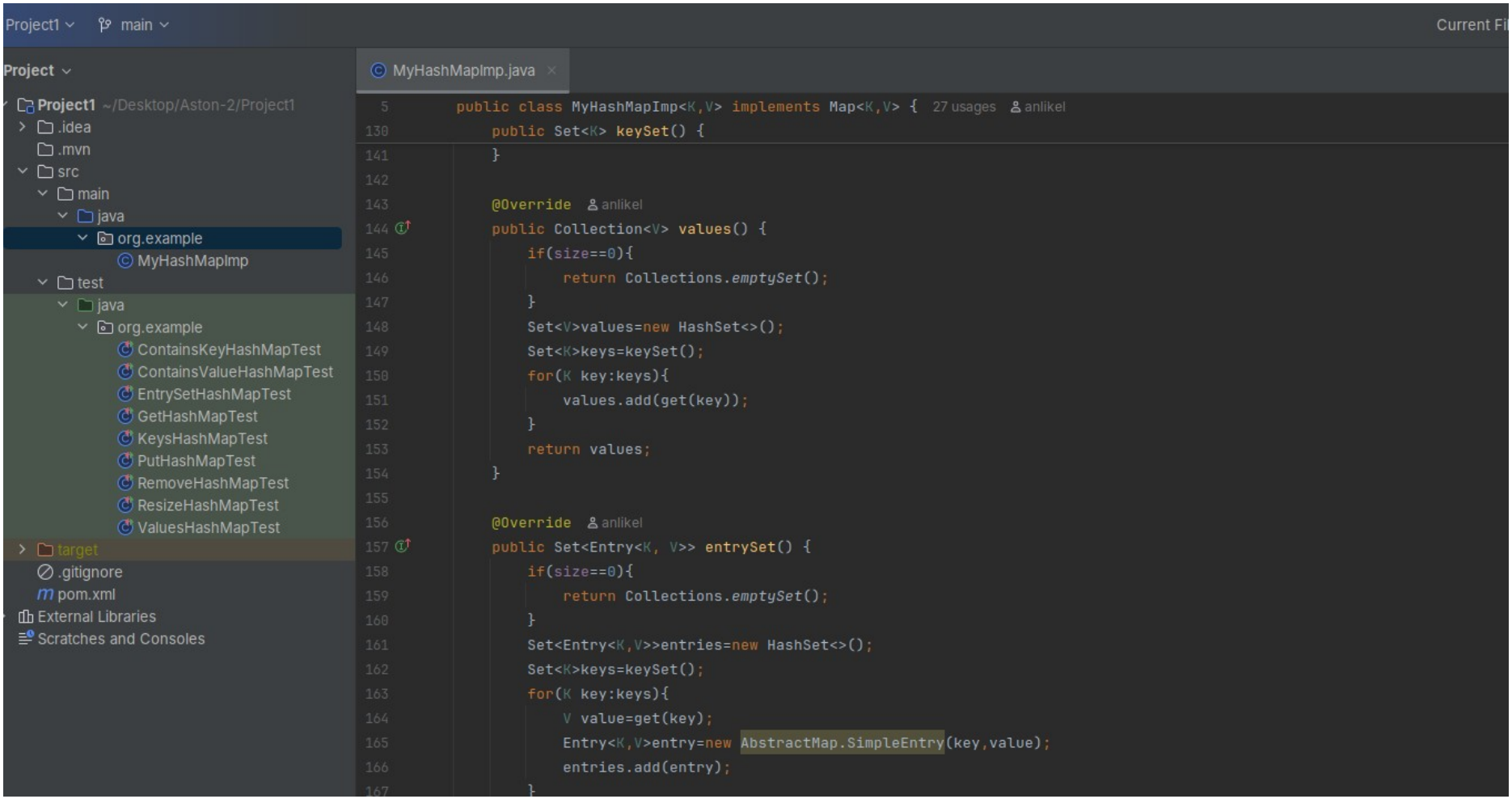

Project1
main
Current Fi
Project
Project1 ~/Desktop/Aston-2/Project1
.idea
.mvn
src
main
java
org.example
MyHashMapImp
test
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
MyHashMapImp.java
public class MyHashMapImp<K,V> implements Map<K,V> { 27 usages anlikel
private static class Node<K,V>{ 26 usages anlikel
Node(int hash,K key,V value, Node next){ 4 usages anlikel
this.next=next;
}
}
@Override no usages anlikel
public int size() {
return size;
}
@Override anlikel
public boolean isEmpty() {
return (size==0);
}
@Override anlikel
public boolean containsKey(Object key) {
if(size==0){
return false;
}
return checkKey(key);
}
@Override anlikel
public boolean containsValue(Object value) {
boolean result=false;
if(size==0){

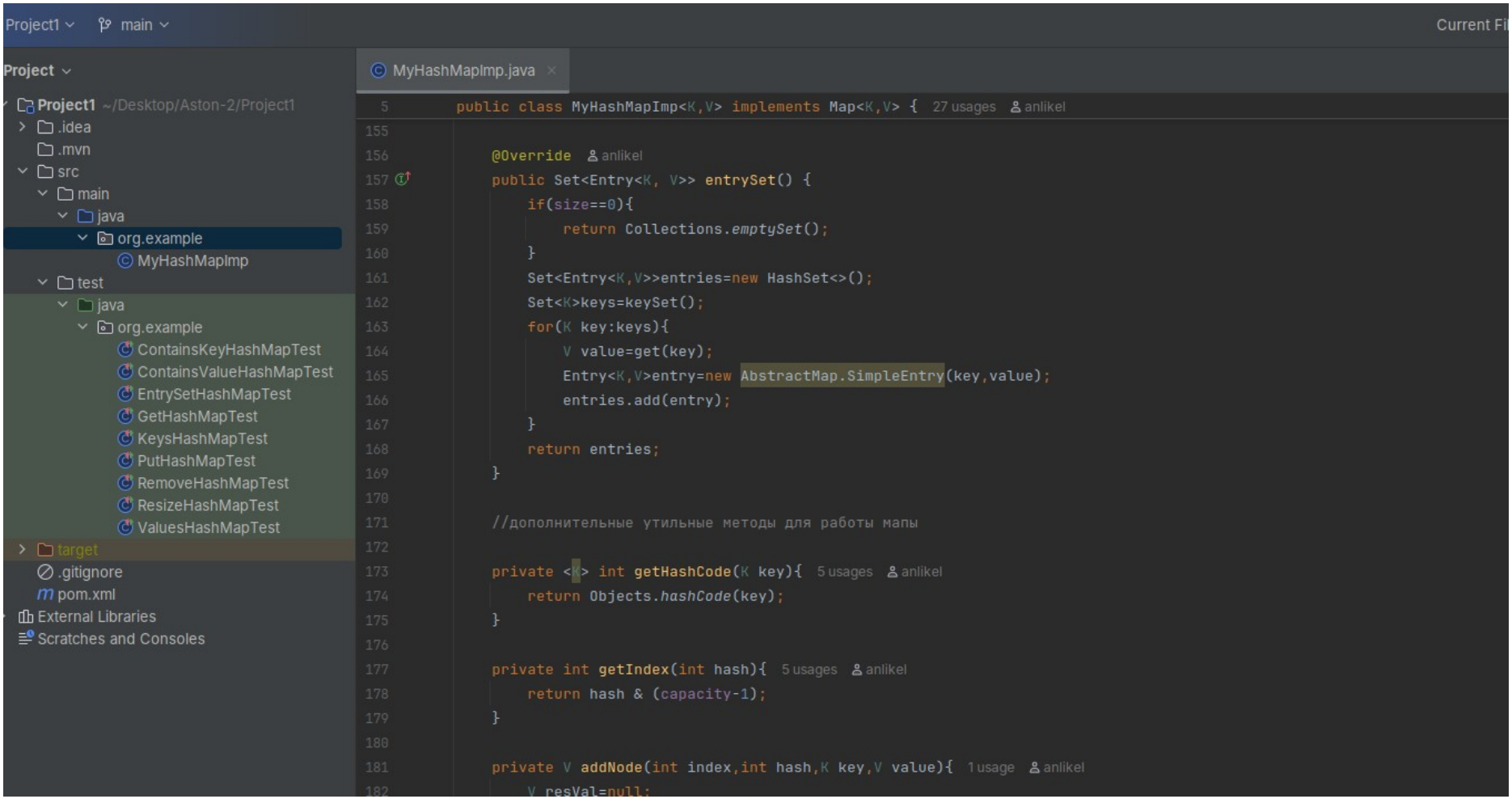

Project1
main
Current Fi
Project
MyHashMapImp.java
Project1 ~/Desktop/Aston-2/Project1
.idea
.mvn
src
main
java
org.example
MyHashMapImp
test
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
5 public class MyHashMapImp<K,V> implements Map<K,V> { 27 usages anlikel
51 public boolean containsValue(Object value) {
57 if(table[i]!=null){
58 result=checkValue(table[i],value);
59 }
60 if(result){break;}
61 }
62 return result;
63 }
64
65 @Override anlikel
66 public V get(Object key) {
67 if(size==0){
68 return null;
69 }
70 int hash=getHashCode(key);
71 int index=getIndex(hash);
72 V resVal=null;
73 if(table[index]!=null){
74 resVal=getNode(index,hash,key);
75 }
76 return resVal;
77 }
78
79 @Override anlikel
80 public V put(K key, V value) {
81 int hash=getHashCode(key);
82 int index=getIndex(hash);
83 V resVal=null;

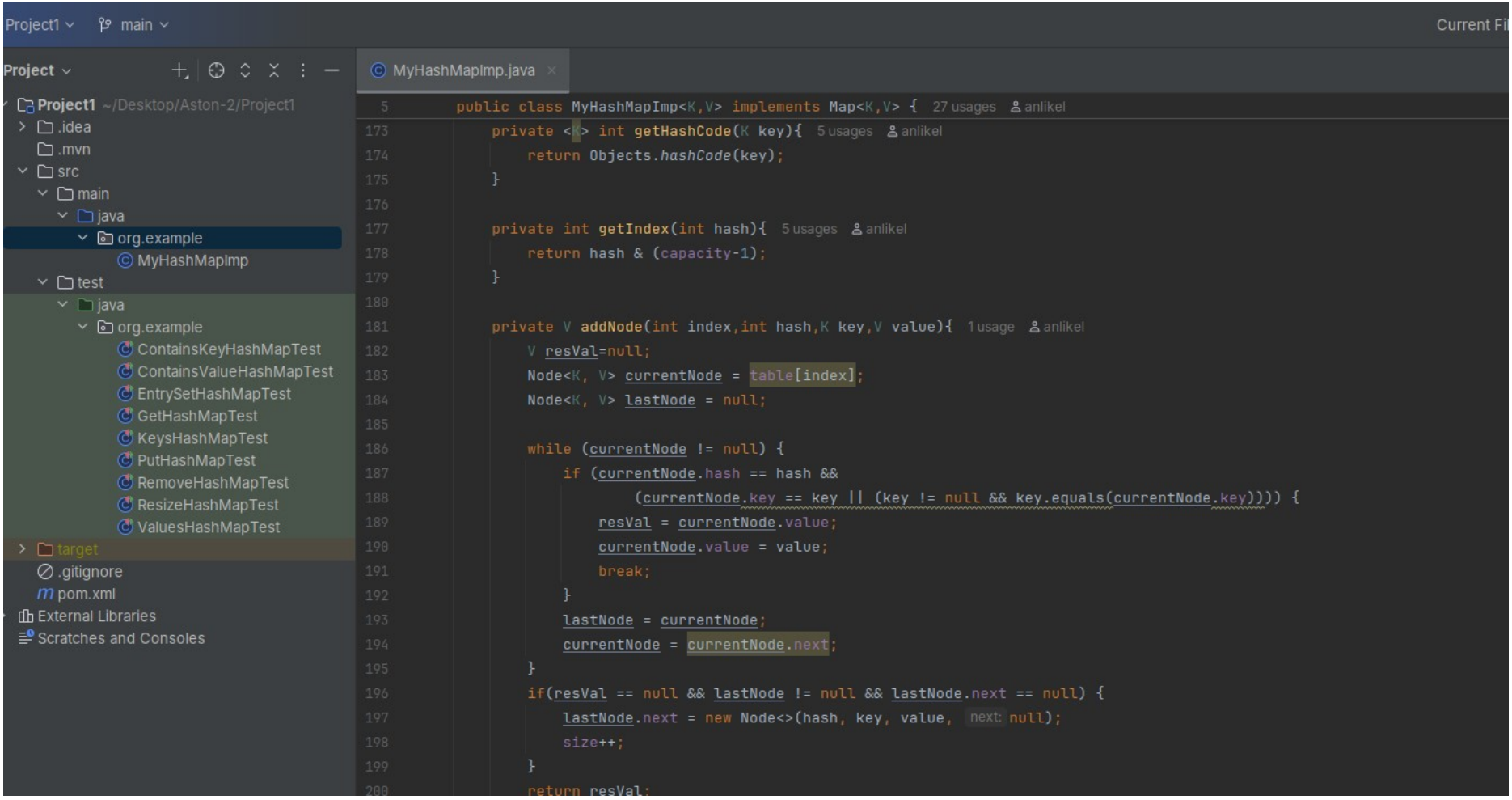

Project1
main
Current Fi
Project
Project1 ~/Desktop/Aston-2/Project1
.idea
.mvn
src
main
java
org.example
MyHashMapImp
test
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
MyHashMapImp.java
public class MyHashMapImp<K,V> implements Map<K,V> { 27 usages anlikel
public V put(K key, V value) {
int hash=getHashCode(key);
int index=getIndex(hash);
V resVal=null;
if(table[index]==null){
Node <K,V>newNode=new Node(hash,key,value, next: null);
table[index]=newNode;
size++;
}
else {
resVal=addNode(index,hash,key,value);
}
resize();
return resVal;
}
@Override anlikel
public V remove(Object key) {
if(size==0){
return null;
}
int hash=getHashCode(key);
int index=getIndex(hash);
V resVal=null;
if(table[index]!=null){
resVal=removeNode(index,hash,key);
}

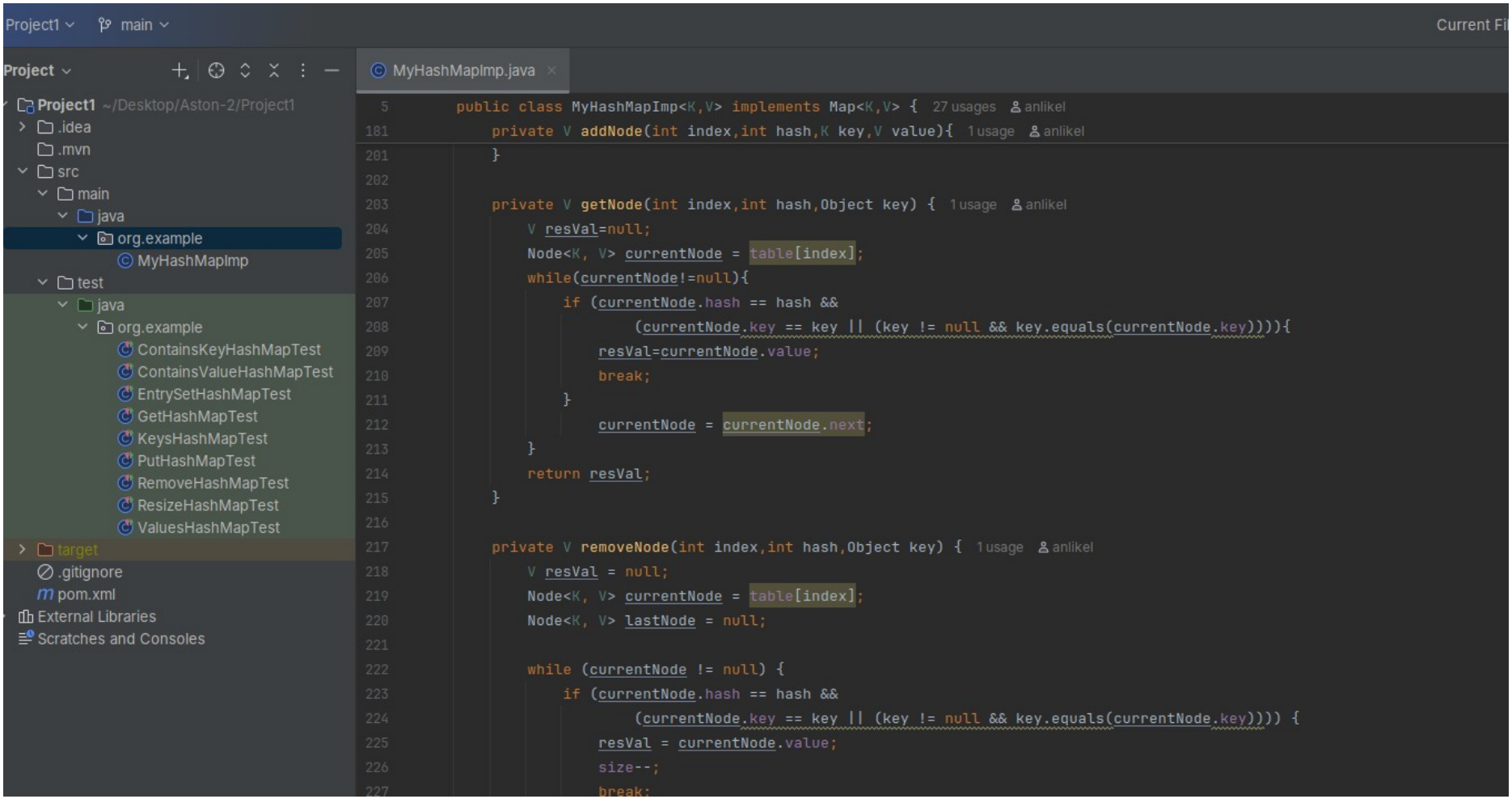

Project1
main
Current Fi
Project
MyHashMapImp.java
Project1 ~/Desktop/Aston-2/Project1
.idea
.mvn
src
main
java
org.example
MyHashMapImp
test
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
5 public class MyHashMapImp<K,V> implements Map<K,V> { 27 usages anlikel
98 public V remove(Object key) {
99 if(size==0){
100 return null;
101 }
102 int hash=getHashCode(key);
103 int index=getIndex(hash);
104 V resVal=null;
105 if(table[index]!=null){
106 resVal=removeNode(index,hash,key);
107 }
108 resize();
109 return resVal;
110 }
111
112 @Override anlikel
113 public void putAll(Map<? extends K, ? extends V> m) {
114 Set<? extends K>keys=m.keySet();
115 for(K key:keys){
116 V value=m.get(key);
117 this.put(key,value);
118 }
119 }
120
121 @Override anlikel
122 public void clear() {
123 for(int i=0;i<capacity;i++){
124 table[i]=null;
125 }

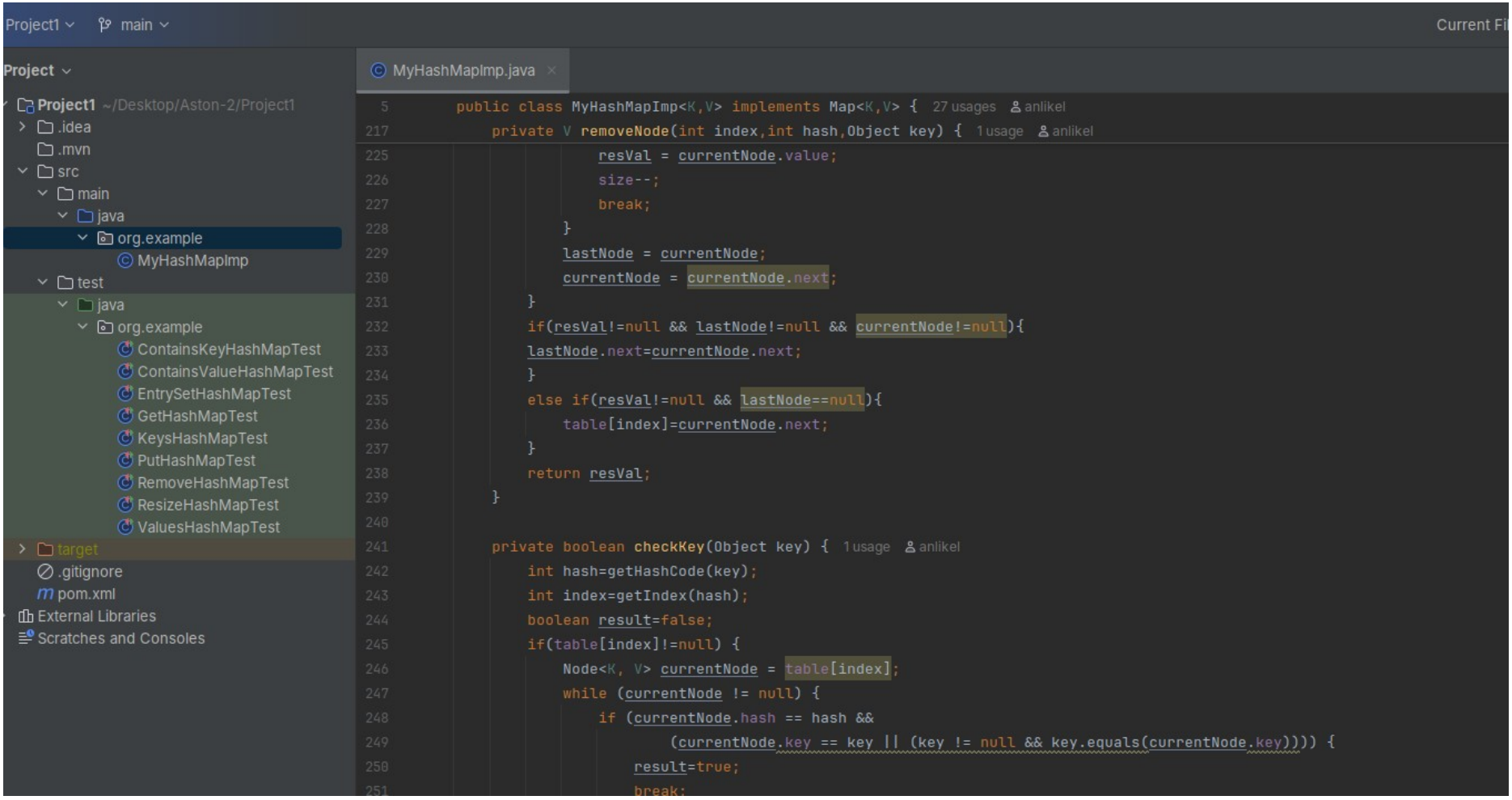

Project1
main
Current Fi
Project
MyHashMapImp.java
Project1 ~/Desktop/Aston-2/Project1
.idea
.mvn
src
main
java
org.example
MyHashMapImp
test
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
5 public class MyHashMapImp<K,V> implements Map<K,V> { 27 usages anlikel
122 public void clear() {
123 for(int i=0;i<capacity;i++){
124 table[i]=null;
125 }
126 size=0;
127 }
128
129 @Override anlikel
130 public Set<K> keySet() {
131 if(size==0){
132 return Collections.emptySet();
133 }
134 Set<K>keys=new HashSet<>();
135 for(int i=0;i<capacity;i++){
136 if(table[i]!=null){
137 addKeys(table[i],keys);
138 }
139 }
140 return keys;
141 }
142
143 @Override anlikel
144 public Collection<V> values() {
145 if(size==0){
146 return Collections.emptySet();
147 }
148 Set<V>values=new HashSet<>();
149 Set<K>keys=keySet();

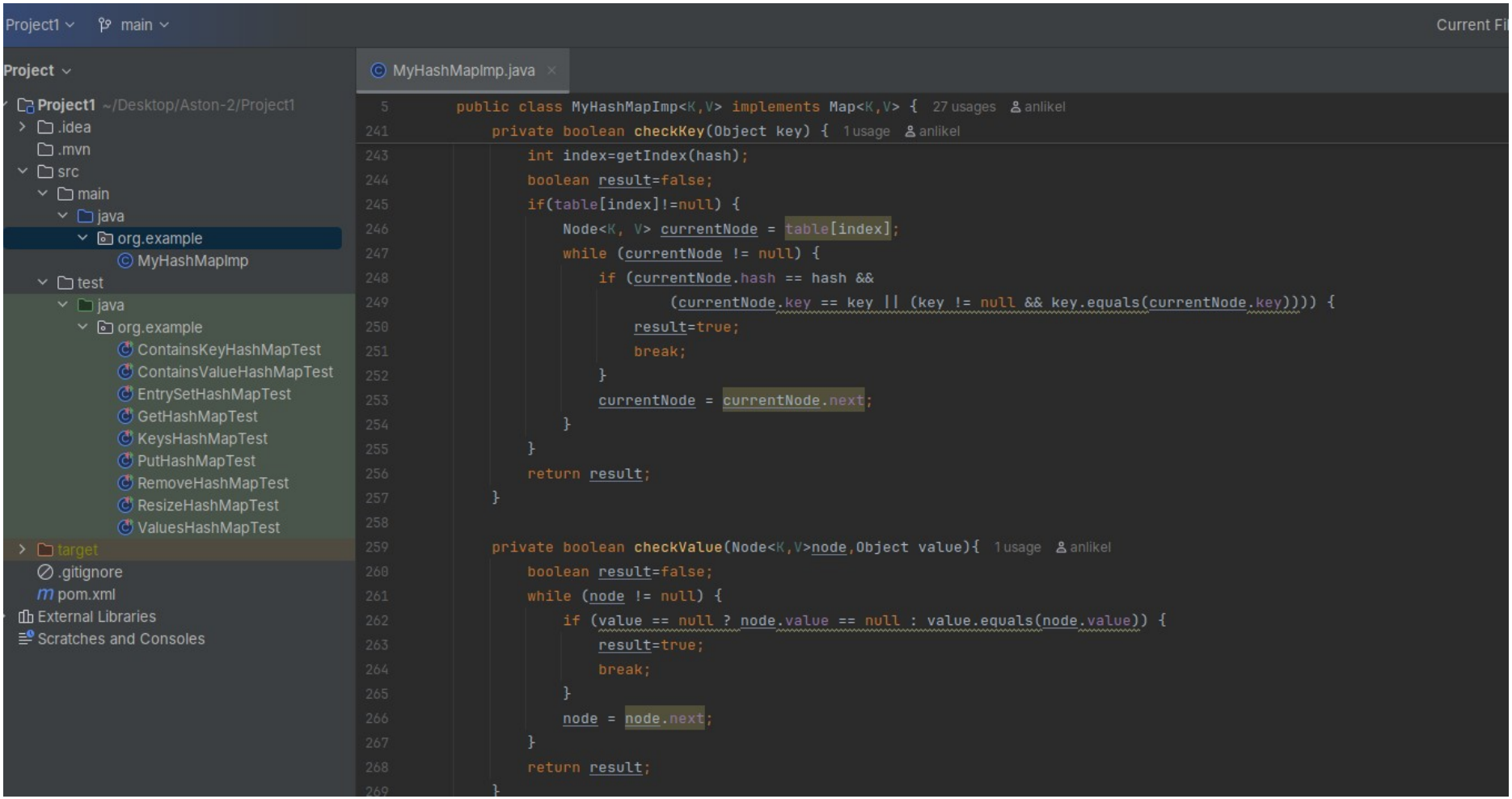

Project1
main
Current Fi
Project
Project1 ~/Desktop/Aston-2/Project1
.idea
.mvn
src
main
java
org.example
MyHashMapImp
test
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
MyHashMapImp.java
5 public class MyHashMapImp<K,V> implements Map<K,V> { 27 usages anlikel
130 public Set<K> keySet() {
141 }
142
143 @Override anlikel
144 public Collection<V> values() {
145 if(size==0){
146 return Collections.emptySet();
147 }
148 Set<V>values=new HashSet<>();
149 Set<K>keys=keySet();
150 for(K key:keys){
151 values.add(get(key));
152 }
153 return values;
154 }
155
156 @Override anlikel
157 public Set<Entry<K, V>> entrySet() {
158 if(size==0){
159 return Collections.emptySet();
160 }
161 Set<Entry<K,V>>entries=new HashSet<>();
162 Set<K>keys=keySet();
163 for(K key:keys){
164 V value=get(key);
165 Entry<K,V>entry=new AbstractMap.SimpleEntry(key,value);
166 entries.add(entry);
167 }

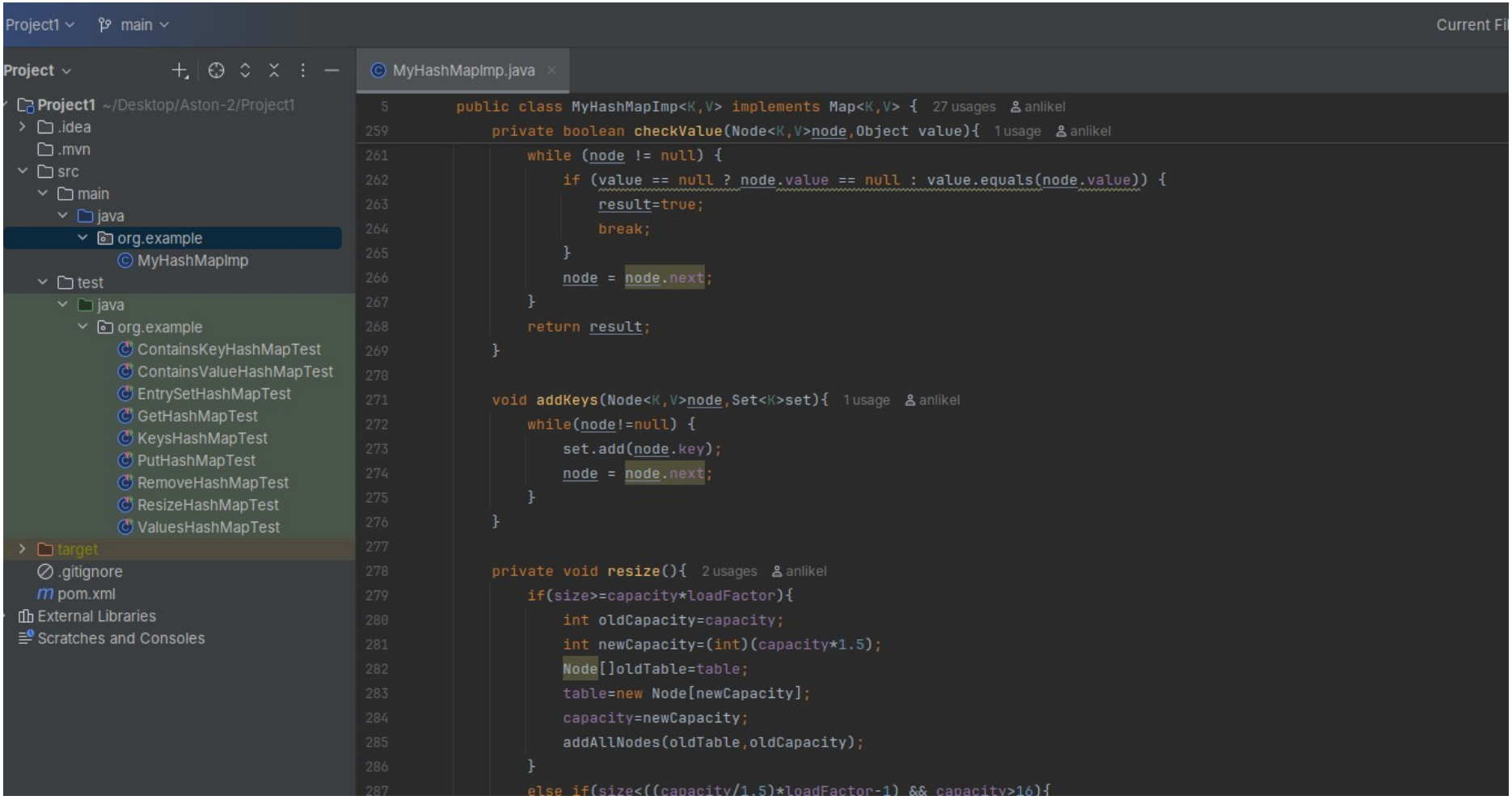

Project1
main
Current Fi
Project
MyHashMapImp.java
Project1 ~/Desktop/Aston-2/Project1
.idea
.mvn
src
main
java
org.example
MyHashMapImp
test
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
5 public class MyHashMapImp<K,V> implements Map<K,V> { 27 usages anlikel
155
156 @Override anlikel
157 public Set<Entry<K, V>> entrySet() {
158 if(size==0){
159 return Collections.emptySet();
160 }
161 Set<Entry<K,V>>entries=new HashSet<>();
162 Set<K>keys=keySet();
163 for(K key:keys){
164 V value=get(key);
165 Entry<K,V>entry=new AbstractMap.SimpleEntry(key,value);
166 entries.add(entry);
167 }
168 return entries;
169 }
170
171 //дополнительные утильные методы для работы мапы
172
173 private <K> int getHashCode(K key){ 5 usages anlikel
174 return Objects.hashCode(key);
175 }
176
177 private int getIndex(int hash){ 5 usages anlikel
178 return hash & (capacity-1);
179 }
180
181 private V addNode(int index,int hash,K key,V value){ 1 usage anlikel
182 V resVal=null;

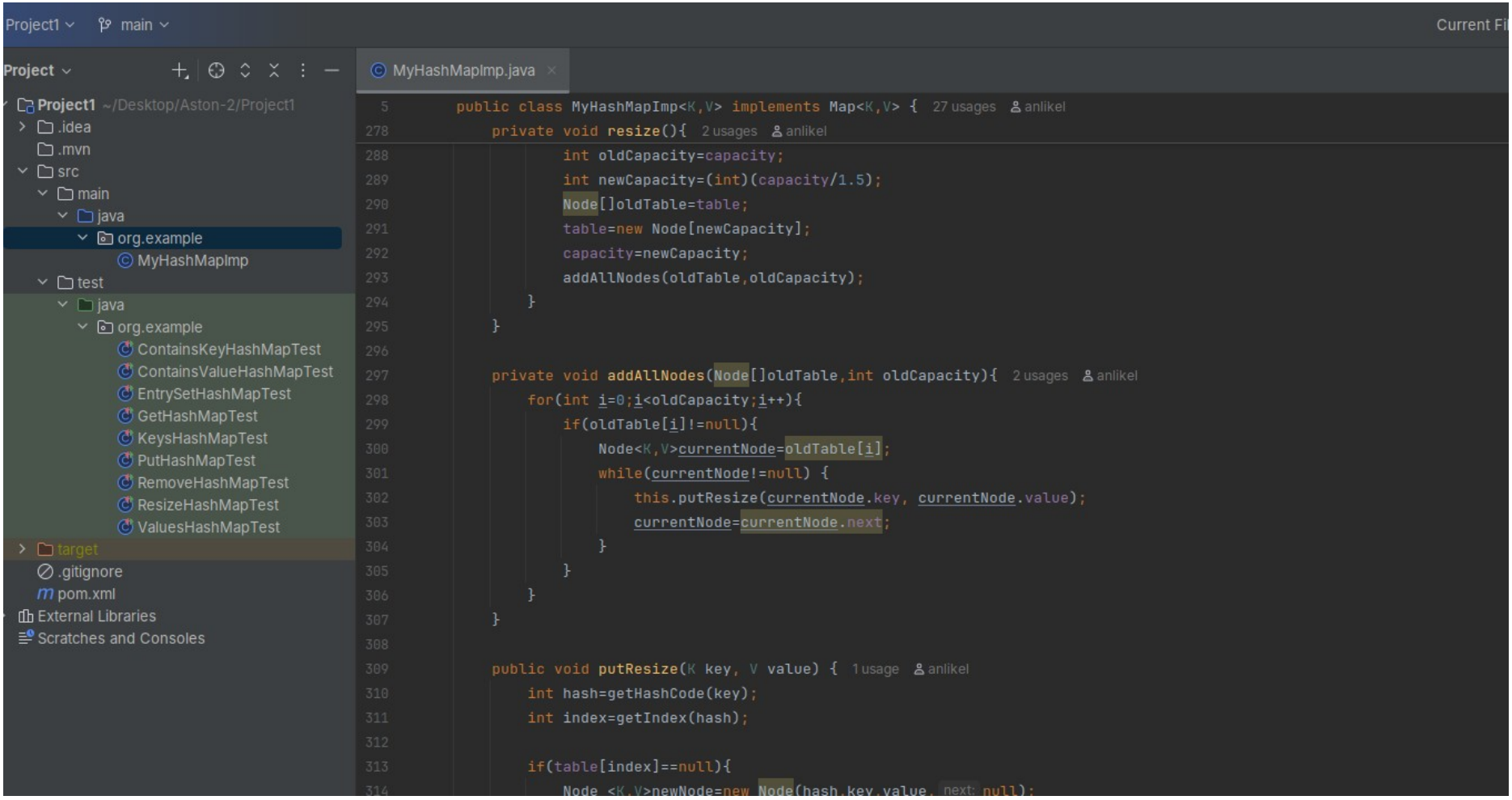

Project1
main
Current Fi
Project
MyHashMapImp.java
Project1 ~/Desktop/Aston-2/Project1
.idea
.mvn
src
main
java
org.example
MyHashMapImp
test
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
5 public class MyHashMapImp<K,V> implements Map<K,V> { 27 usages anlikel
173 private <K> int getHashCode(K key){ 5 usages anlikel
174 return Objects.hashCode(key);
175 }
176
177 private int getIndex(int hash){ 5 usages anlikel
178 return hash & (capacity-1);
179 }
180
181 private V addNode(int index,int hash,K key,V value){ 1 usage anlikel
182 V resVal=null;
183 Node<K, V> currentNode = table[index];
184 Node<K, V> lastNode = null;
185
186 while (currentNode != null) {
187 if (currentNode.hash == hash &&
188 (currentNode.key == key || (key != null && key.equals(currentNode.key)))) {
189 resVal = currentNode.value;
190 currentNode.value = value;
191 break;
192 }
193 lastNode = currentNode;
194 currentNode = currentNode.next;
195 }
196 if(resVal == null && lastNode != null && lastNode.next == null) {
197 lastNode.next = new Node<>(hash, key, value, next: null);
198 size++;
199 }
200 return resVal;

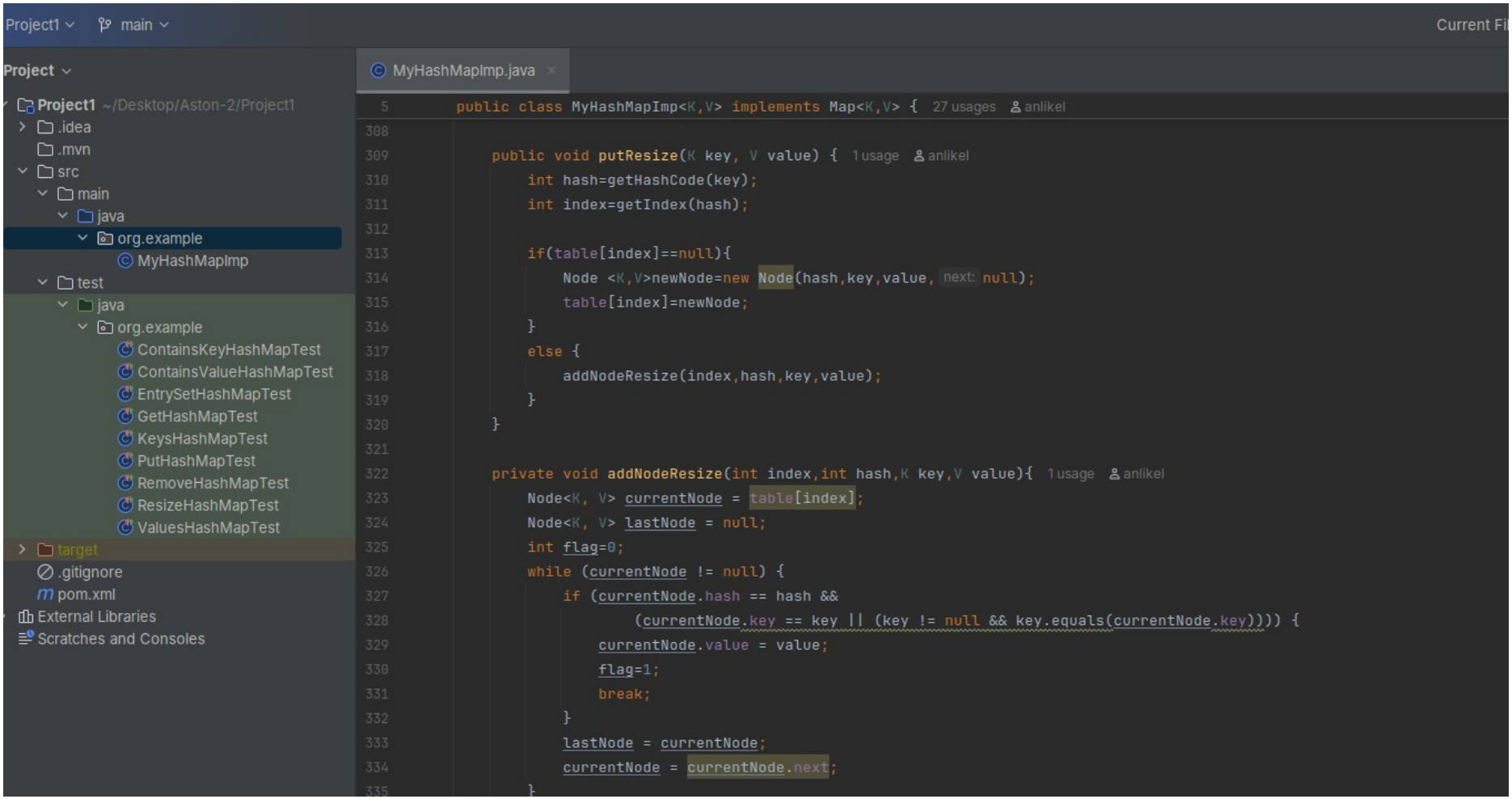

Project1
main
Current Fi
Project
MyHashMapImp.java
Project1 ~/Desktop/Aston-2/Project1
.idea
.mvn
src
main
java
org.example
MyHashMapImp
test
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
5 public class MyHashMapImp<K,V> implements Map<K,V> { 27 usages anlikel
181 private V addNode(int index,int hash,K key,V value){ 1 usage anlikel
201 }
202
203 private V getNode(int index,int hash,Object key) { 1 usage anlikel
204 V resVal=null;
205 Node<K, V> currentNode = table[index];
206 while(currentNode!=null){
207 if (currentNode.hash == hash &&
208 (currentNode.key == key || (key != null && key.equals(currentNode.key)))){
209 resVal=currentNode.value;
210 break;
211 }
212 currentNode = currentNode.next;
213 }
214 return resVal;
215 }
216
217 private V removeNode(int index,int hash,Object key) { 1 usage anlikel
218 V resVal = null;
219 Node<K, V> currentNode = table[index];
220 Node<K, V> lastNode = null;
221
222 while (currentNode != null) {
223 if (currentNode.hash == hash &&
224 (currentNode.key == key || (key != null && key.equals(currentNode.key)))) {
225 resVal = currentNode.value;
226 size--;
227 break;

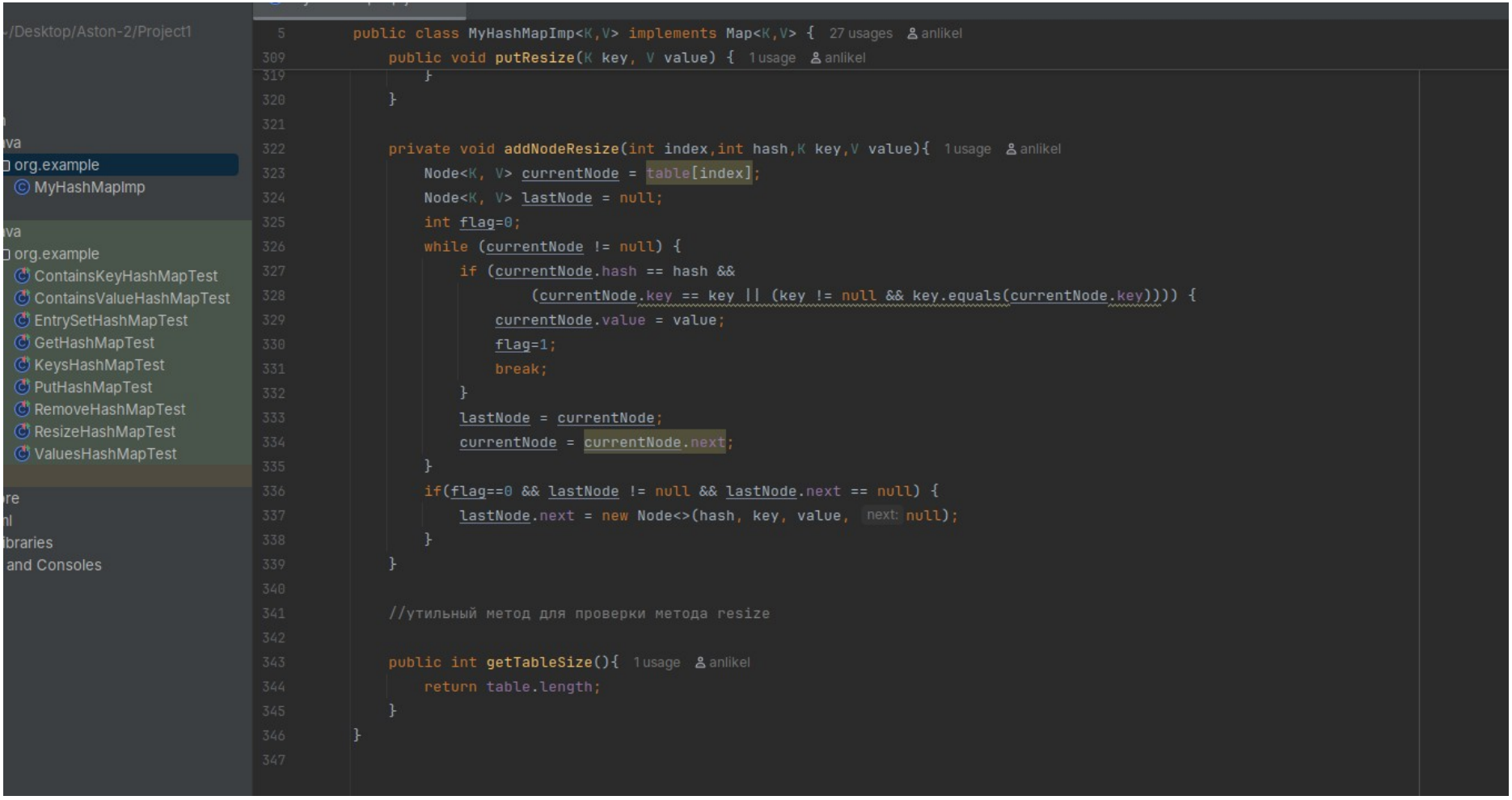

Project1
main
Current Fi
Project
Project1 ~/Desktop/Aston-2/Project1
.idea
.mvn
src
main
java
org.example
MyHashMapImp
test
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
MyHashMapImp.java
5 public class MyHashMapImp<K,V> implements Map<K,V> { 27 usages anlikel
217 private V removeNode(int index,int hash,Object key) { 1 usage anlikel
225 resVal = currentNode.value;
226 size--;
227 break;
228 }
229 lastNode = currentNode;
230 currentNode = currentNode.next;
231 }
232 if(resVal!=null && lastNode!=null && currentNode!=null){
233 lastNode.next=currentNode.next;
234 }
235 else if(resVal!=null && lastNode==null){
236 table[index]=currentNode.next;
237 }
238 return resVal;
239 }
240
241 private boolean checkKey(Object key) { 1 usage anlikel
242 int hash=getHashCode(key);
243 int index=getIndex(hash);
244 boolean result=false;
245 if(table[index]!=null) {
246 Node<K, V> currentNode = table[index];
247 while (currentNode != null) {
248 if (currentNode.hash == hash &&
249 (currentNode.key == key || (key != null && key.equals(currentNode.key)))) {
250 result=true;
251 break;


Project1
main
Current Fi
Project
MyHashMapImp.java
Project1 ~/Desktop/Aston-2/Project1
.idea
.mvn
src
main
java
org.example
MyHashMapImp
test
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
5 public class MyHashMapImp<K,V> implements Map<K,V> { 27 usages anlikel
241 private boolean checkKey(Object key) { 1 usage anlikel
243 int index=getIndex(hash);
244 boolean result=false;
245 if(table[index]!=null) {
246 Node<K, V> currentNode = table[index];
247 while (currentNode != null) {
248 if (currentNode.hash == hash &&
249 (currentNode.key == key || (key != null && key.equals(currentNode.key)))) {
250 result=true;
251 break;
252 }
253 currentNode = currentNode.next;
254 }
255 }
256 return result;
257 }
258
259 private boolean checkValue(Node<K,V>node,Object value){ 1 usage anlikel
260 boolean result=false;
261 while (node != null) {
262 if (value == null ? node.value == null : value.equals(node.value)) {
263 result=true;
264 break;
265 }
266 node = node.next;
267 }
268 return result;
269 }

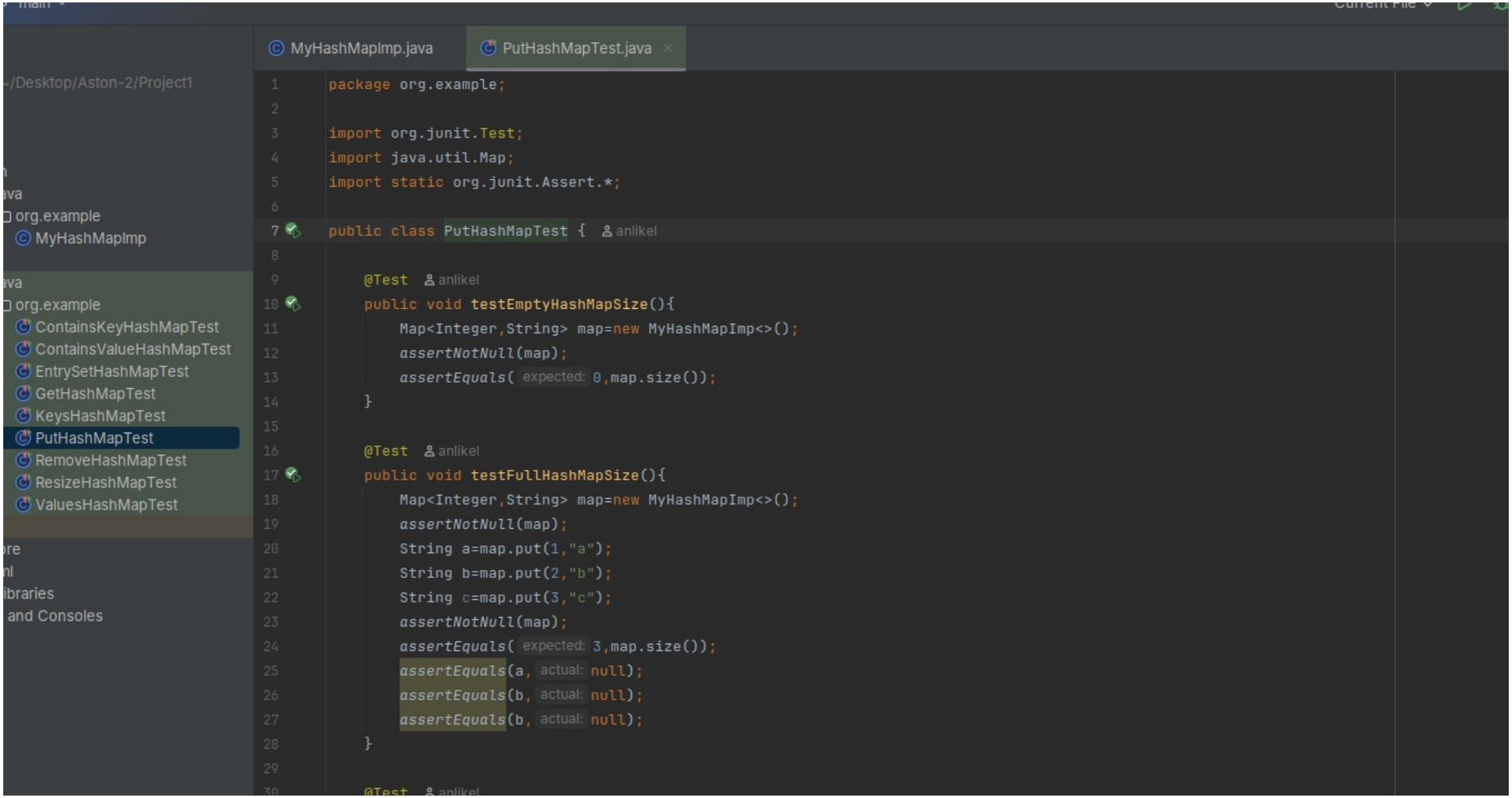

Project1
main
Current Fi
Project
Project1 ~/Desktop/Aston-2/Project1
.idea
.mvn
src
main
java
org.example
MyHashMapImp
test
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
MyHashMapImp.java
5 public class MyHashMapImp<K,V> implements Map<K,V> { 27 usages anlikel
259 private boolean checkValue(Node<K,V>node,Object value){ 1 usage anlikel
261 while (node != null) {
262 if (value == null ? node.value == null : value.equals(node.value)) {
263 result=true;
264 break;
265 }
266 node = node.next;
267 }
268 return result;
269 }
270
271 void addKeys(Node<K,V>node,Set<K>set){ 1 usage anlikel
272 while(node!=null) {
273 set.add(node.key);
274 node = node.next;
275 }
276 }
277
278 private void resize(){ 2 usages anlikel
279 if(size>=capacity*loadFactor){
280 int oldCapacity=capacity;
281 int newCapacity=(int)(capacity*1.5);
282 Node[]oldTable=table;
283 table=new Node[newCapacity];
284 capacity=newCapacity;
285 addAllNodes(oldTable,oldCapacity);
286 }
287 else if(size<((capacity/1.5)*loadFactor-1) && capacity>16){

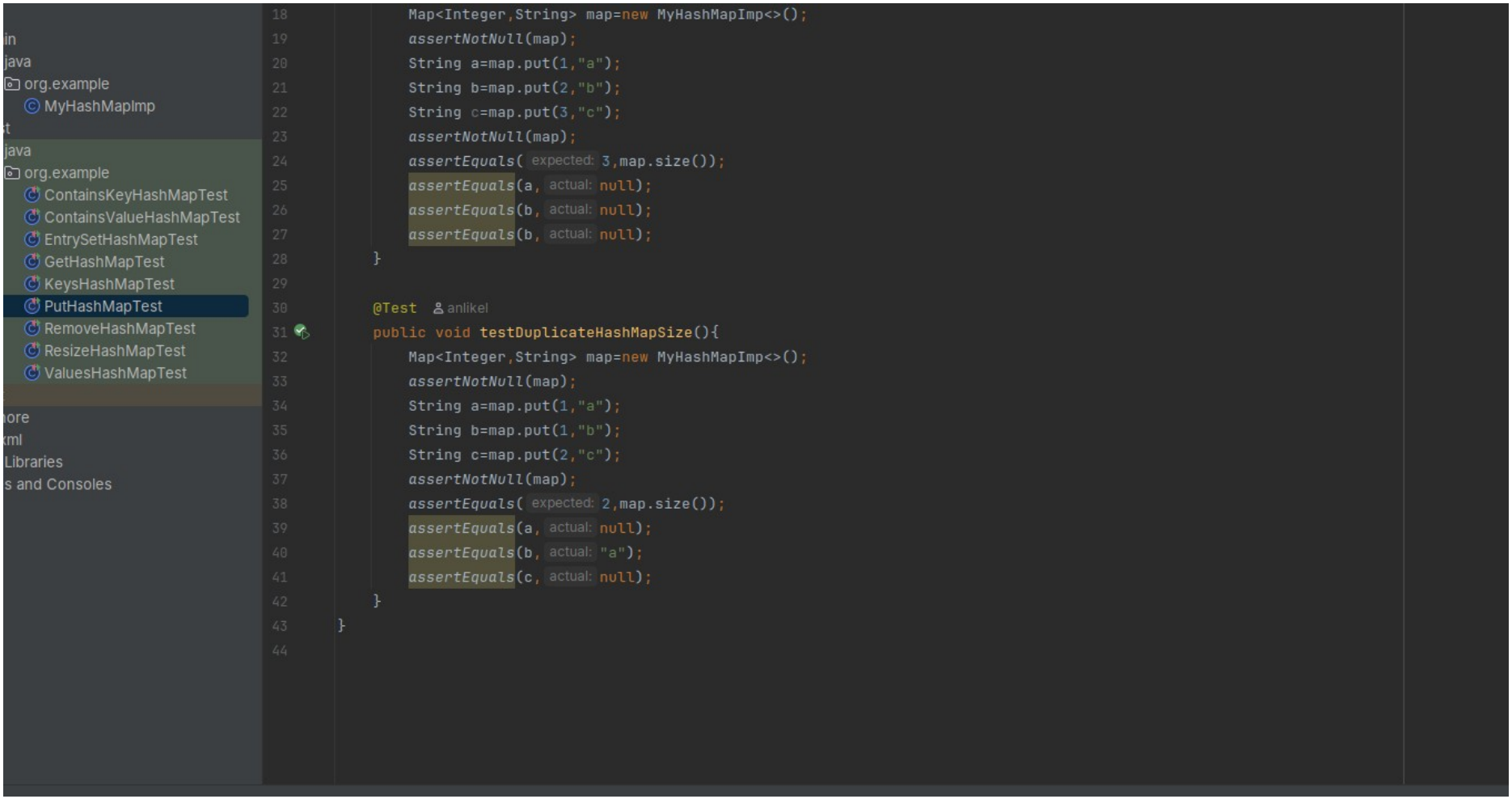

Project1
main
Current Fi
Project
MyHashMapImp.java
Project1 ~/Desktop/Aston-2/Project1
.idea
.mvn
src
main
java
org.example
MyHashMapImp
test
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
5 public class MyHashMapImp<K,V> implements Map<K,V> { 27 usages anlikel
278 private void resize(){ 2 usages anlikel
288 int oldCapacity=capacity;
289 int newCapacity=(int)(capacity/1.5);
290 Node[]oldTable=table;
291 table=new Node[newCapacity];
292 capacity=newCapacity;
293 addAllNodes(oldTable,oldCapacity);
294 }
295 }
296
297 private void addAllNodes(Node[]oldTable,int oldCapacity){ 2 usages anlikel
298 for(int i=0;i<oldCapacity;i++){
299 if(oldTable[i]!=null){
300 Node<K,V>currentNode=oldTable[i];
301 while(currentNode!=null) {
302 this.putResize(currentNode.key, currentNode.value);
303 currentNode=currentNode.next;
304 }
305 }
306 }
307 }
308
309 public void putResize(K key, V value) { 1 usage anlikel
310 int hash=getHashCode(key);
311 int index=getIndex(hash);
312
313 if(table[index]==null){
314 Node <K,V>newNode=new Node(hash,key,value, next: null);

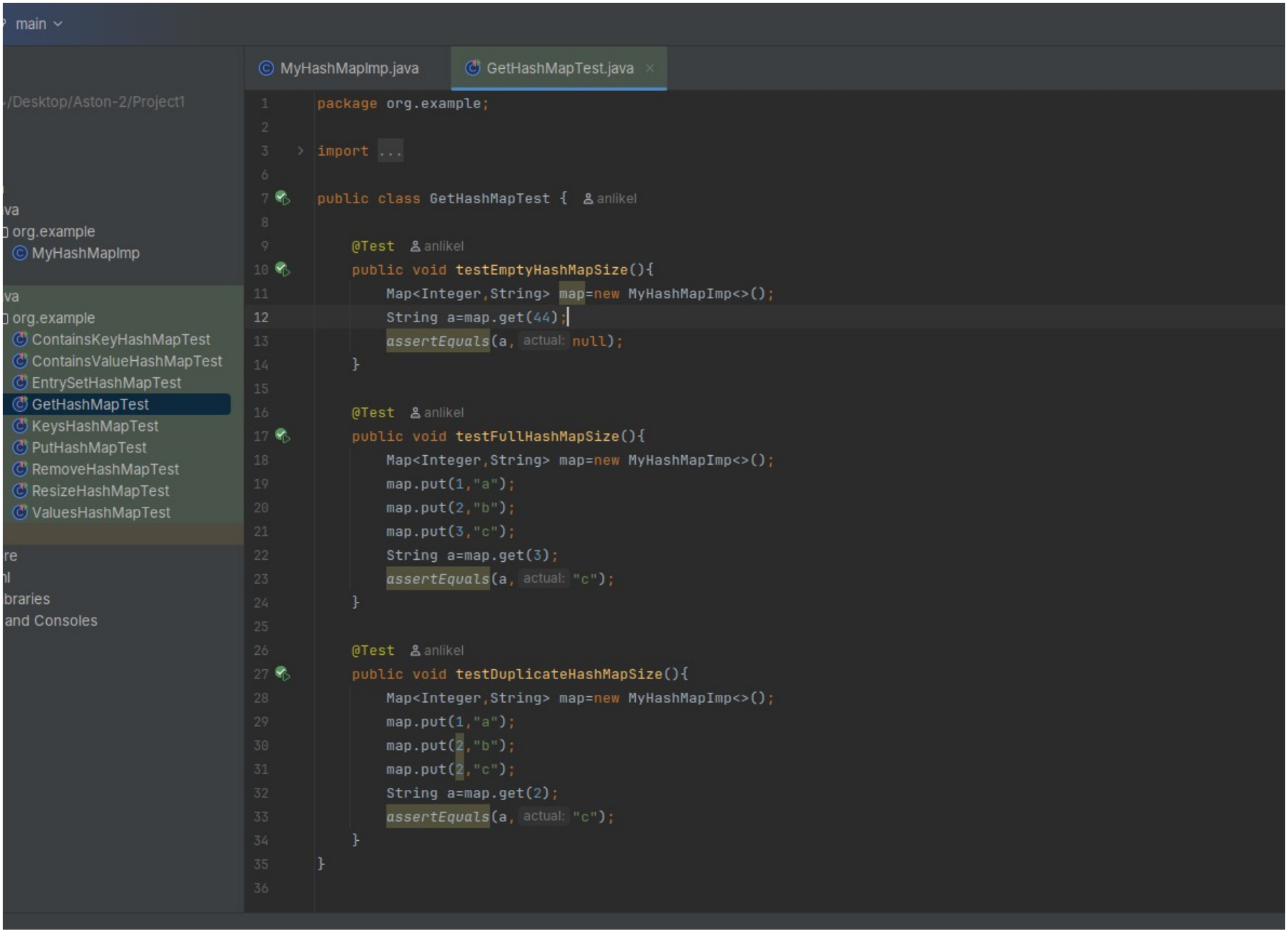

Project1
main
Current Fi
Project
Project1 ~/Desktop/Aston-2/Project1
.idea
.mvn
src
main
java
org.example
MyHashMapImp
test
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
MyHashMapImp.java
5 public class MyHashMapImp<K,V> implements Map<K,V> { 27 usages anlikel
308
309 public void putResize(K key, V value) { 1 usage anlikel
310 int hash=getHashCode(key);
311 int index=getIndex(hash);
312
313 if(table[index]==null){
314 Node <K,V>newNode=new Node(hash,key,value, next: null);
315 table[index]=newNode;
316 }
317 else {
318 addNodeResize(index,hash,key,value);
319 }
320 }
321
322 private void addNodeResize(int index,int hash,K key,V value){ 1 usage anlikel
323 Node<K, V> currentNode = table[index];
324 Node<K, V> lastNode = null;
325 int flag=0;
326 while (currentNode != null) {
327 if (currentNode.hash == hash &&
328 (currentNode.key == key || (key != null && key.equals(currentNode.key)))) {
329 currentNode.value = value;
330 flag=1;
331 break;
332 }
333 lastNode = currentNode;
334 currentNode = currentNode.next;
335 }

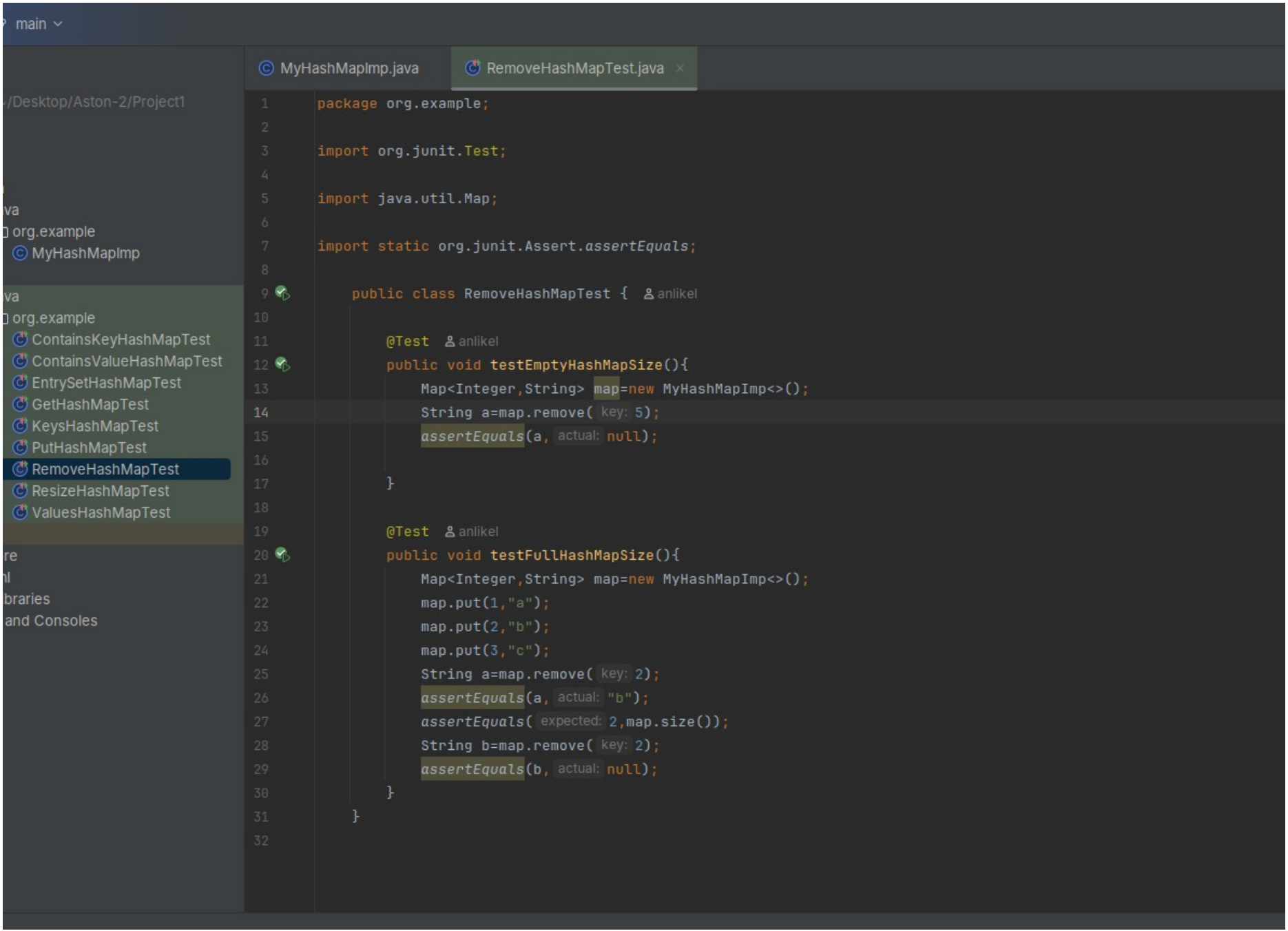

/Desktop/Aston-2/Project1
va
org.example
MyHashMapImp
va
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
re
l
braries
and Consoles
5 public class MyHashMapImp<K,V> implements Map<K,V> { 27 usages anlikel
309 public void putResize(K key, V value) { 1 usage anlikel
319 }
320 }
321
322 private void addNodeResize(int index,int hash,K key,V value){ 1 usage anlikel
323 Node<K, V> currentNode = table[index];
324 Node<K, V> lastNode = null;
325 int flag=0;
326 while (currentNode != null) {
327 if (currentNode.hash == hash &&
328 (currentNode.key == key || (key != null && key.equals(currentNode.key)))) {
329 currentNode.value = value;
330 flag=1;
331 break;
332 }
333 lastNode = currentNode;
334 currentNode = currentNode.next;
335 }
336 if(flag==0 && lastNode != null && lastNode.next == null) {
337 lastNode.next = new Node<>(hash, key, value, next: null);
338 }
339 }
340
341 //утильный метод для проверки метода resize
342
343 public int getTableSize(){ 1 usage anlikel
344 return table.length;
345 }
346 }
347

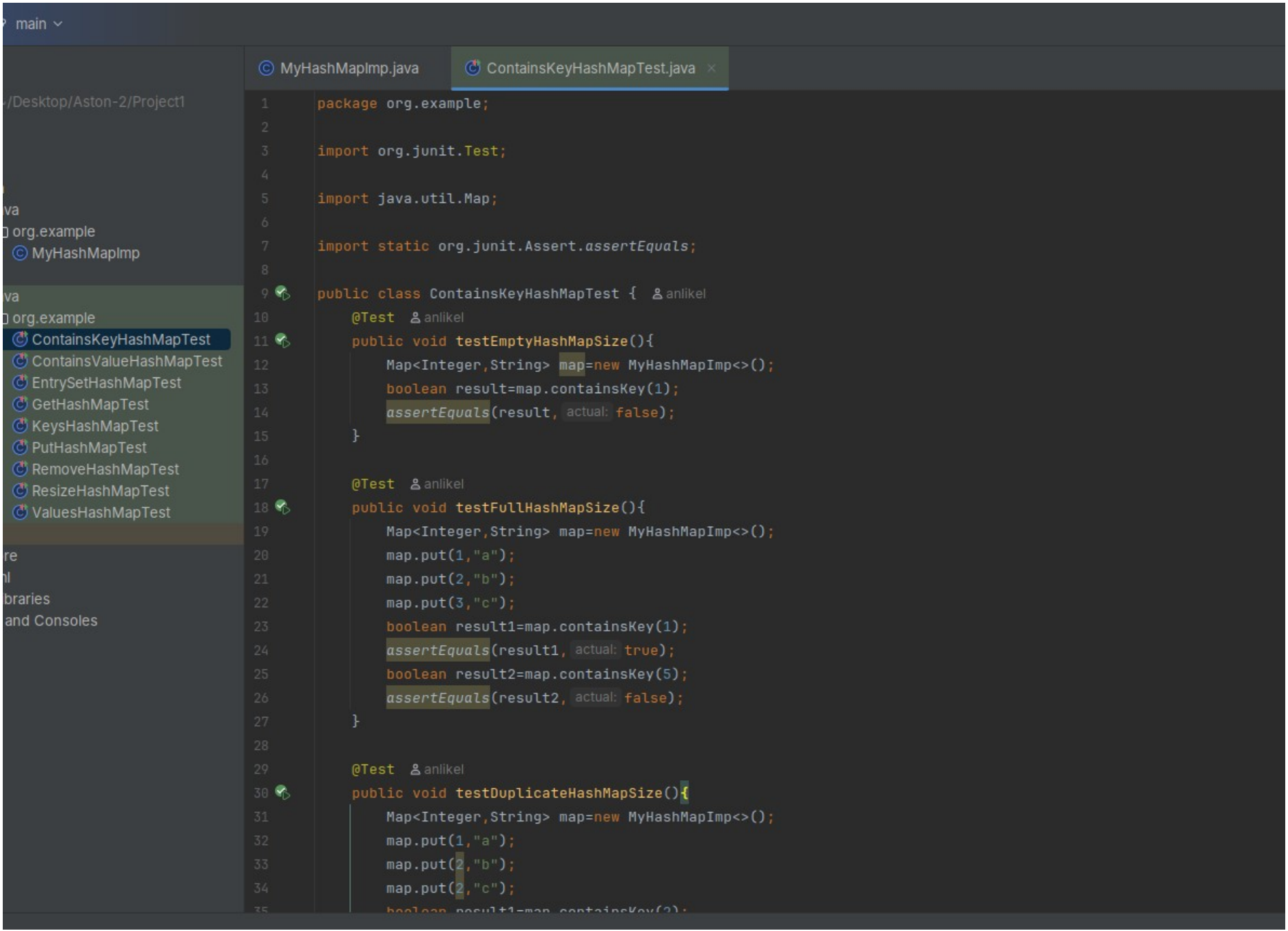
Скриншоты Тестов;

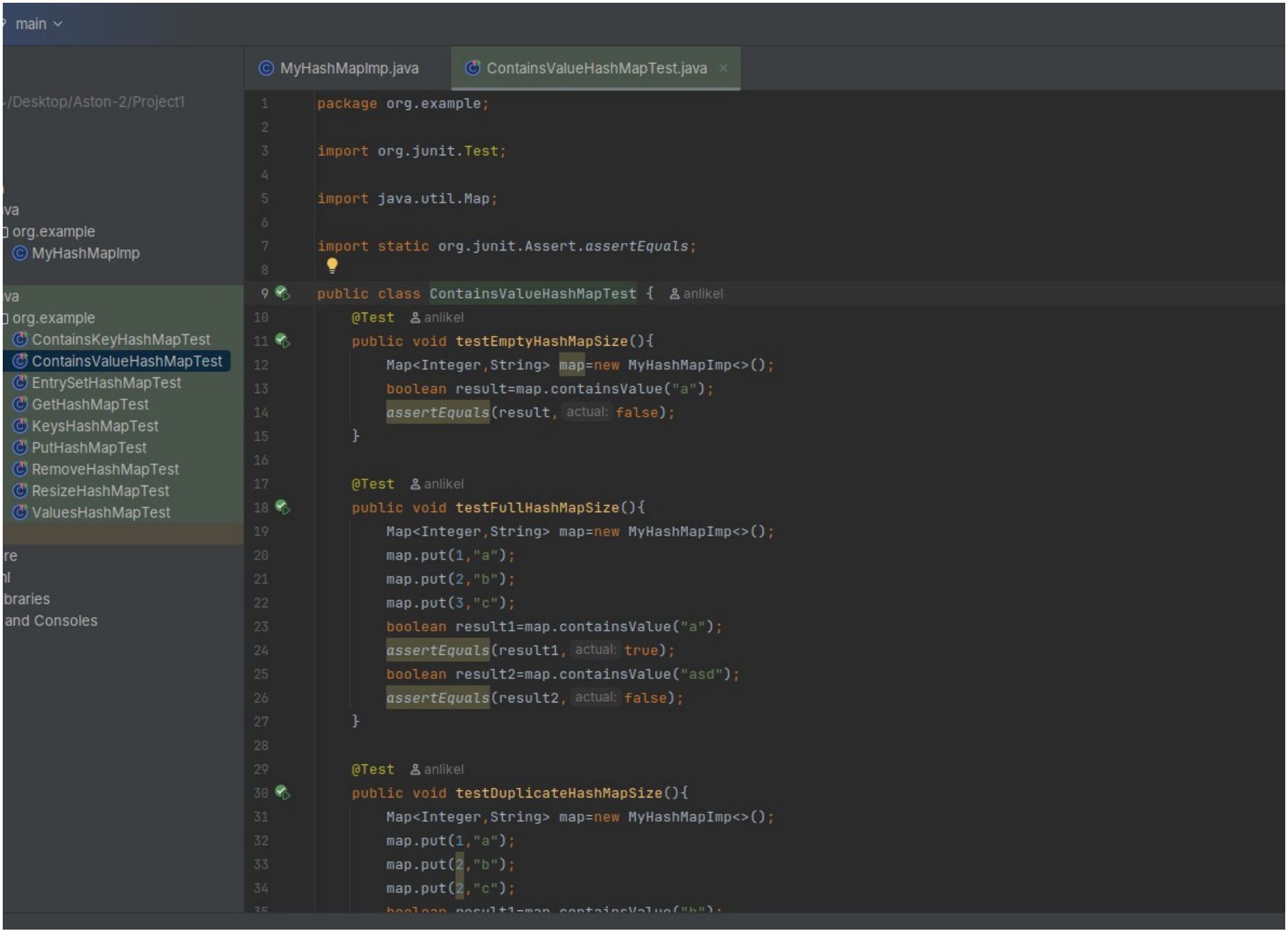


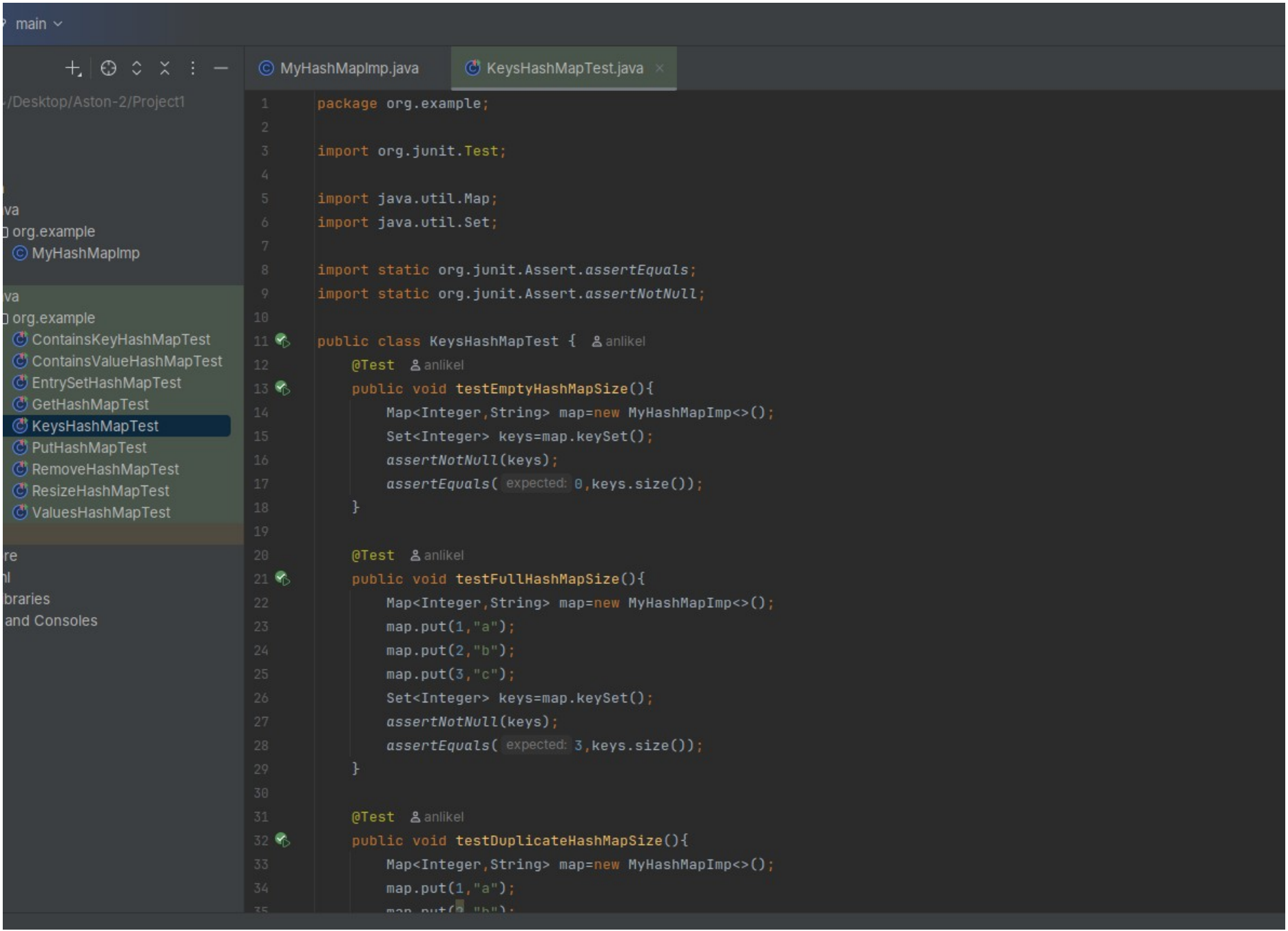

in
java
org.example
MyHashMapImp
t
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
ore
xml
Libraries
s and Consoles
18 Map<Integer,String> map=new MyHashMapImp<>();
19 assertNotNull(map);
20 String a=map.put(1,"a");
21 String b=map.put(2,"b");
22 String c=map.put(3,"c");
23 assertNotNull(map);
24 assertEquals(expected: 3,map.size());
25 assertEquals(a, actual: null);
26 assertEquals(b, actual: null);
27 assertEquals(b, actual: null);
28 }
29
30 @Test anlikel
31 public void testDuplicateHashMapSize(){
32 Map<Integer,String> map=new MyHashMapImp<>();
33 assertNotNull(map);
34 String a=map.put(1,"a");
35 String b=map.put(1,"b");
36 String c=map.put(2,"c");
37 assertNotNull(map);
38 assertEquals(expected: 2,map.size());
39 assertEquals(a, actual: null);
40 assertEquals(b, actual: "a");
41 assertEquals(c, actual: null);
42 }
43 }
44

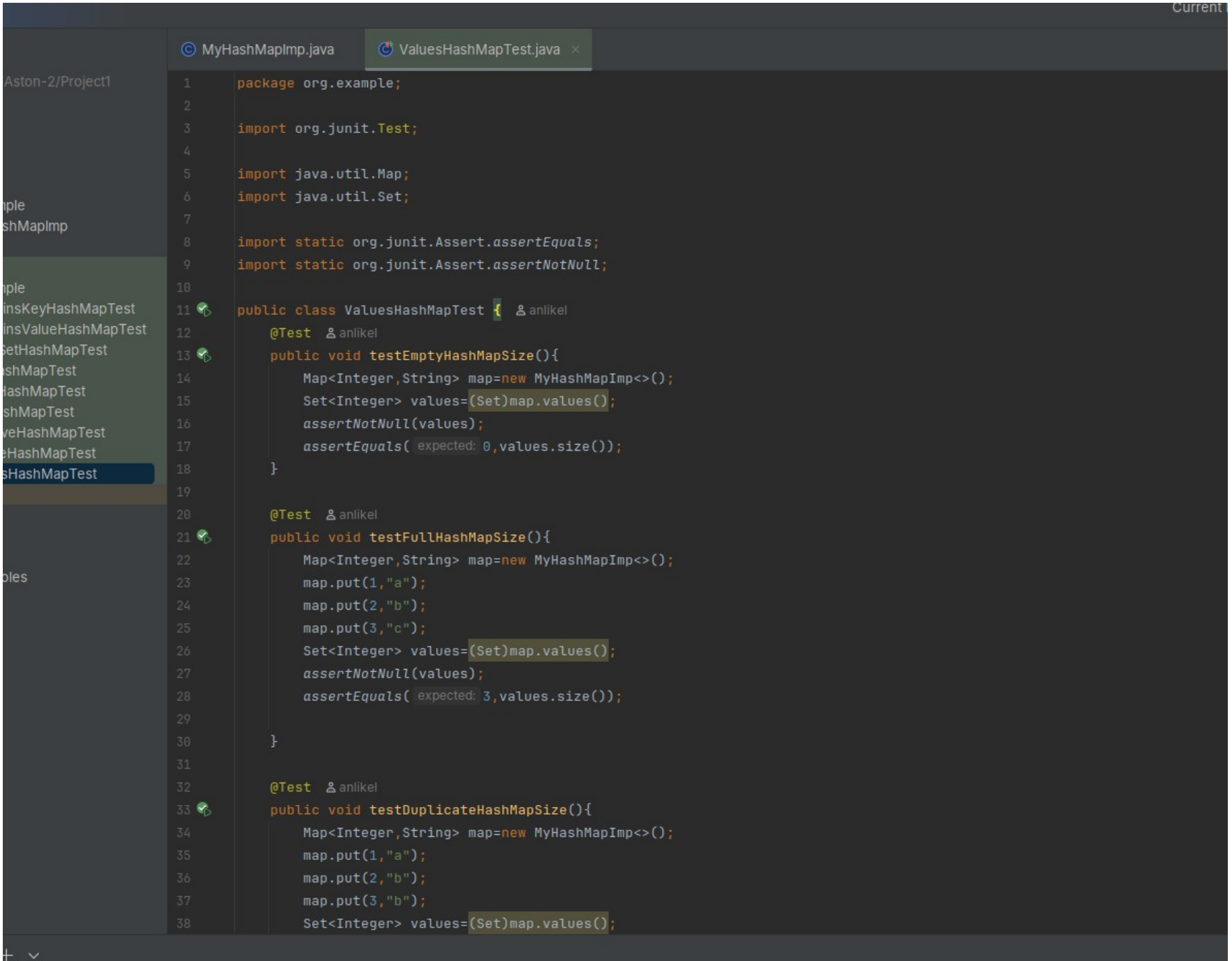

main
MyHashMapImp.java
GetHashMapTest.java
/Desktop/Aston-2/Project1
va
org.example
MyHashMapImp
va
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
re
l
braries
and Consoles
package org.example;
import ...
public class GetHashMapTest { anlikel
@Test anlikel
public void testEmptyHashMapSize(){
Map<Integer,String> map=new MyHashMapImp<>();
String a=map.get(44);
assertEquals(a, actual: null);
}
@Test anlikel
public void testFullHashMapSize(){
Map<Integer,String> map=new MyHashMapImp<>();
map.put(1,"a");
map.put(2,"b");
map.put(3,"c");
String a=map.get(3);
assertEquals(a, actual: "c");
}
@Test anlikel
public void testDuplicateHashMapSize(){
Map<Integer,String> map=new MyHashMapImp<>();
map.put(1,"a");
map.put(2,"b");
map.put(2,"c");
String a=map.get(2);
assertEquals(a, actual: "c");
}
}

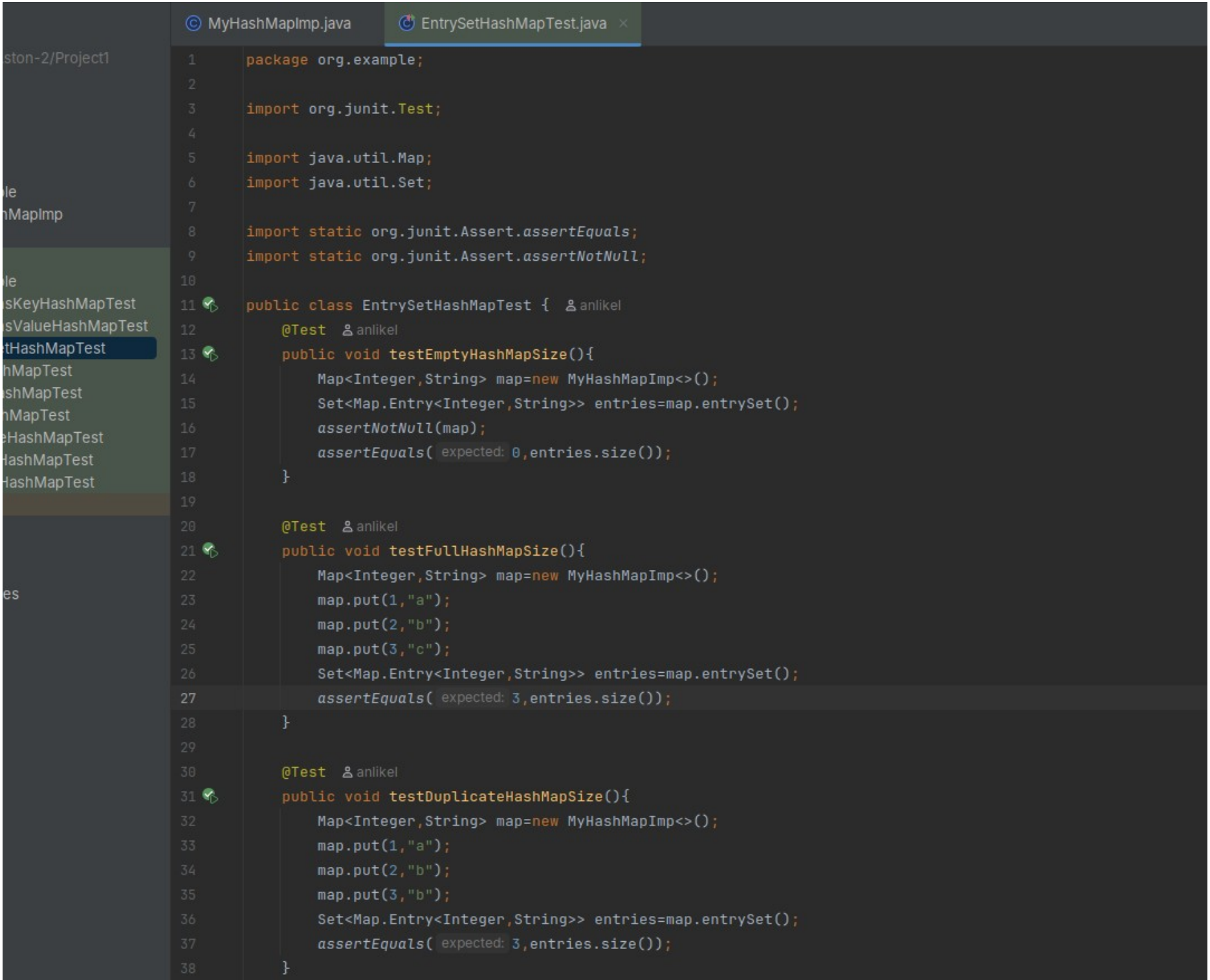

main
MyHashMapImp.java
RemoveHashMapTest.java
/Desktop/Aston-2/Project1
va
org.example
MyHashMapImp
va
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
re
l
braries
and Consoles
package org.example;
import org.junit.Test;
import java.util.Map;
import static org.junit.Assert.assertEquals;
public class RemoveHashMapTest { anlikel
@Test anlikel
public void testEmptyHashMapSize(){
Map<Integer,String> map=new MyHashMapImp<>();
String a=map.remove(key: 5);
assertEquals(a, actual: null);
}
@Test anlikel
public void testFullHashMapSize(){
Map<Integer,String> map=new MyHashMapImp<>();
map.put(1,"a");
map.put(2,"b");
map.put(3,"c");
String a=map.remove(key: 2);
assertEquals(a, actual: "b");
assertEquals(expected: 2,map.size());
String b=map.remove(key: 2);
assertEquals(b, actual: null);
}
}

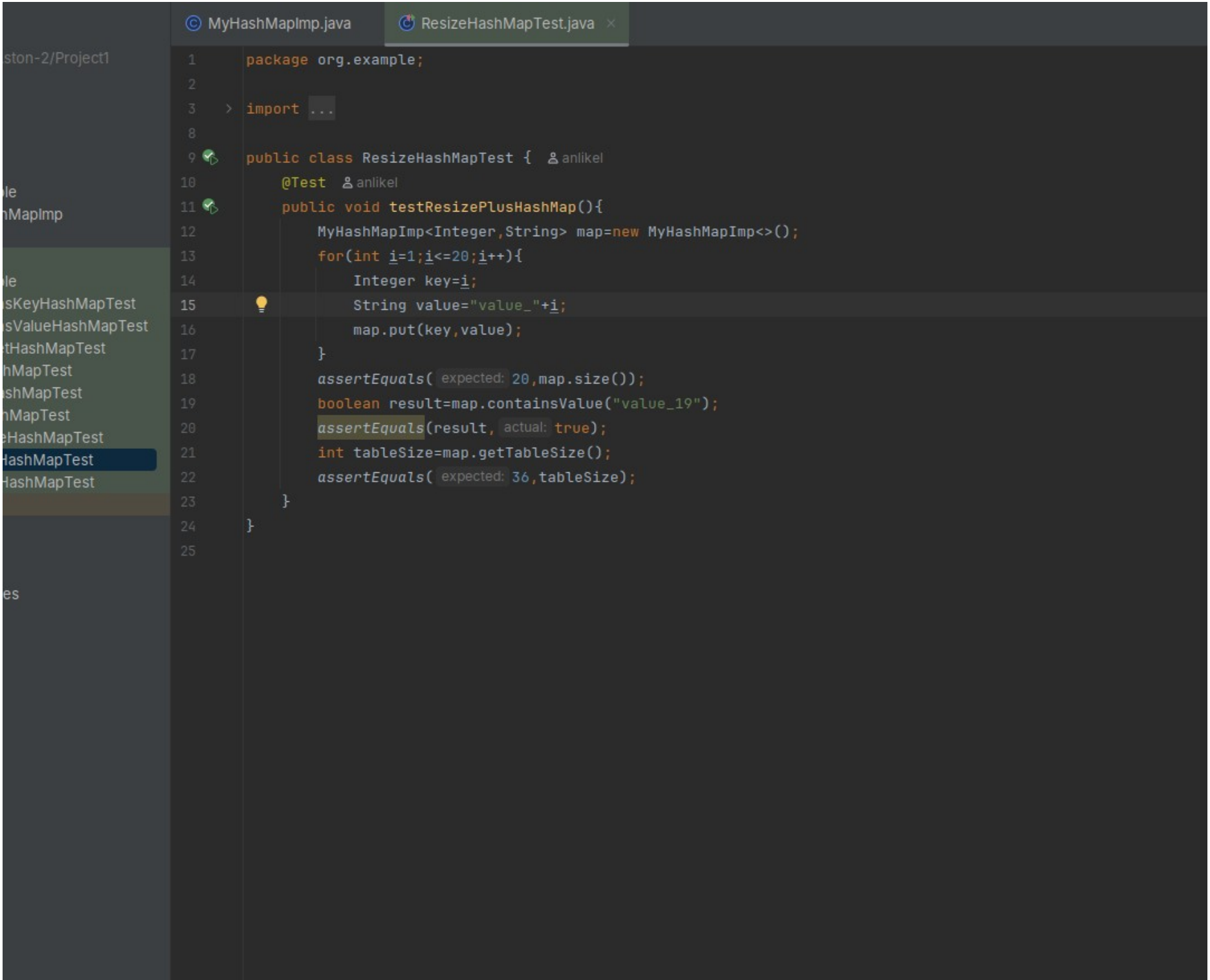

main
MyHashMapImp.java
ContainsKeyHashMapTest.java
/Desktop/Aston-2/Project1
va
org.example
MyHashMapImp
va
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
re
l
braries
and Consoles
package org.example;
import org.junit.Test;
import java.util.Map;
import static org.junit.Assert.assertEquals;
public class ContainsKeyHashMapTest { anlikel
@Test anlikel
public void testEmptyHashMapSize(){
Map<Integer,String> map=new MyHashMapImp<>();
boolean result=map.containsKey(1);
assertEquals(result, actual: false);
}
@Test anlikel
public void testFullHashMapSize(){
Map<Integer,String> map=new MyHashMapImp<>();
map.put(1,"a");
map.put(2,"b");
map.put(3,"c");
boolean result1=map.containsKey(1);
assertEquals(result1, actual: true);
boolean result2=map.containsKey(5);
assertEquals(result2, actual: false);
}
@Test anlikel
public void testDuplicateHashMapSize(){
Map<Integer,String> map=new MyHashMapImp<>();
map.put(1,"a");
map.put(2,"b");
map.put(2,"c");


main
MyHashMapImp.java
ContainsValueHashMapTest.java
/Desktop/Aston-2/Project1
va
org.example
MyHashMapImp
va
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
re
l
braries
and Consoles
package org.example;
import org.junit.Test;
import java.util.Map;
import static org.junit.Assert.assertEquals;
public class ContainsValueHashMapTest { anlikel
@Test anlikel
public void testEmptyHashMapSize(){
Map<Integer,String> map=new MyHashMapImp<>();
boolean result=map.containsValue("a");
assertEquals(result, actual: false);
}
@Test anlikel
public void testFullHashMapSize(){
Map<Integer,String> map=new MyHashMapImp<>();
map.put(1,"a");
map.put(2,"b");
map.put(3,"c");
boolean result1=map.containsValue("a");
assertEquals(result1, actual: true);
boolean result2=map.containsValue("asd");
assertEquals(result2, actual: false);
}
@Test anlikel
public void testDuplicateHashMapSize(){
Map<Integer,String> map=new MyHashMapImp<>();
map.put(1,"a");
map.put(2,"b");
map.put(2,"c");

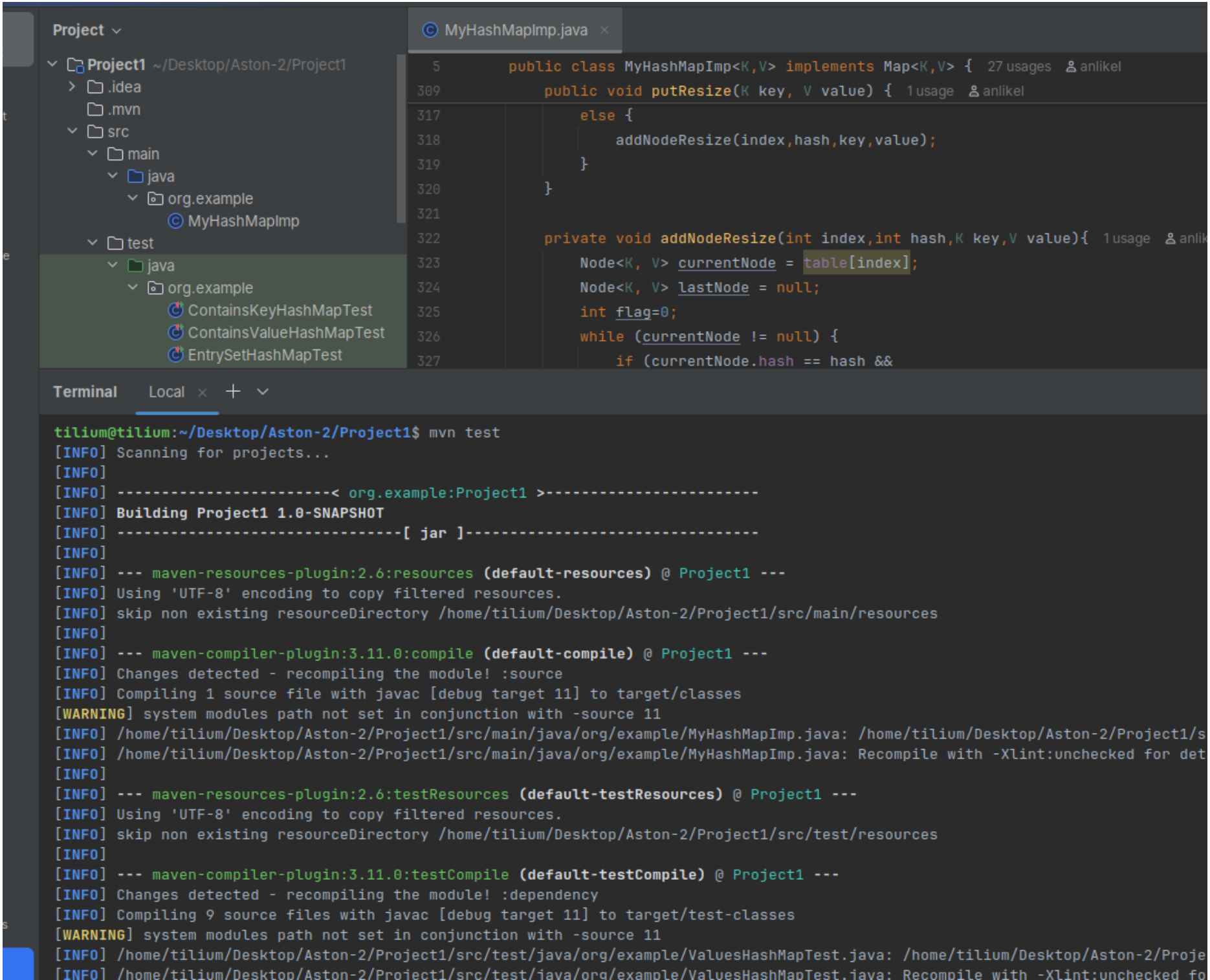

main
MyHashMapImp.java
KeysHashMapTest.java
/Desktop/Aston-2/Project1
va
org.example
MyHashMapImp
va
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
GetHashMapTest
KeysHashMapTest
PutHashMapTest
RemoveHashMapTest
ResizeHashMapTest
ValuesHashMapTest
re
l
braries
and Consoles
package org.example;
import org.junit.Test;
import java.util.Map;
import java.util.Set;
import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertNotNull;
public class KeysHashMapTest { anlikel
@Test anlikel
public void testEmptyHashMapSize(){
Map<Integer,String> map=new MyHashMapImp<>();
Set<Integer> keys=map.keySet();
assertNotNull(keys);
assertEquals(expected: 0,keys.size());
}
@Test anlikel
public void testFullHashMapSize(){
Map<Integer,String> map=new MyHashMapImp<>();
map.put(1,"a");
map.put(2,"b");
map.put(3,"c");
Set<Integer> keys=map.keySet();
assertNotNull(keys);
assertEquals(expected: 3,keys.size());
}
@Test anlikel
public void testDuplicateHashMapSize(){
Map<Integer,String> map=new MyHashMapImp<>();
map.put(1,"a");

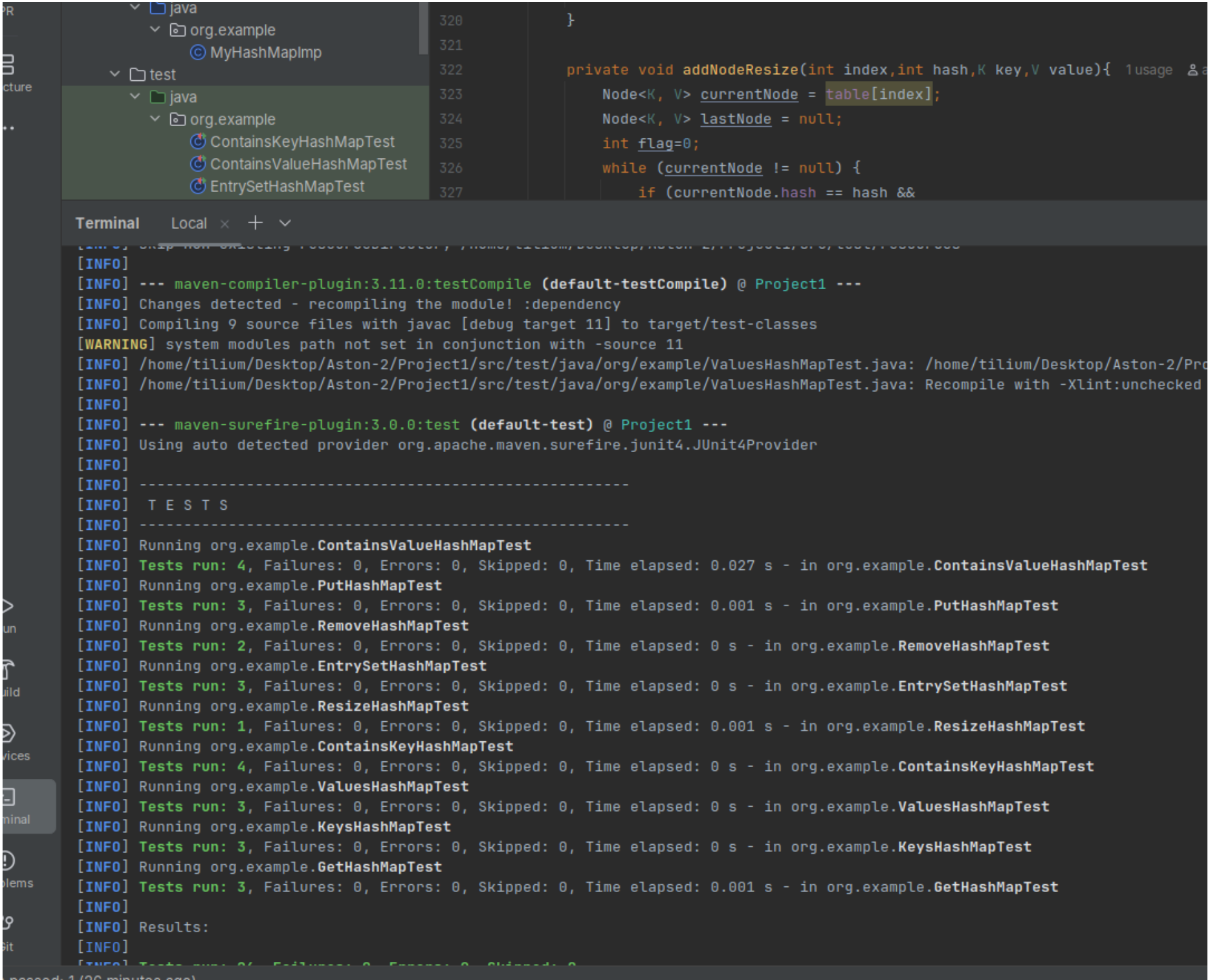

MyHashMapImp.java
ValuesHashMapTest.java
Aston-2/Project1
package org.example;
import org.junit.Test;
import java.util.Map;
import java.util.Set;
import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertNotNull;
public class ValuesHashMapTest {
@Test
public void testEmptyHashMapSize(){
Map<Integer,String> map=new MyHashMapImp<>();
Set<Integer> values=(Set)map.values();
assertNotNull(values);
assertEquals(expected: 0,values.size());
}
@Test
public void testFullHashMapSize(){
Map<Integer,String> map=new MyHashMapImp<>();
map.put(1,"a");
map.put(2,"b");
map.put(3,"c");
Set<Integer> values=(Set)map.values();
assertNotNull(values);
assertEquals(expected: 3,values.size());
}
@Test
public void testDuplicateHashMapSize(){
Map<Integer,String> map=new MyHashMapImp<>();
map.put(1,"a");
map.put(2,"b");
map.put(3,"b");
Set<Integer> values=(Set)map.values();

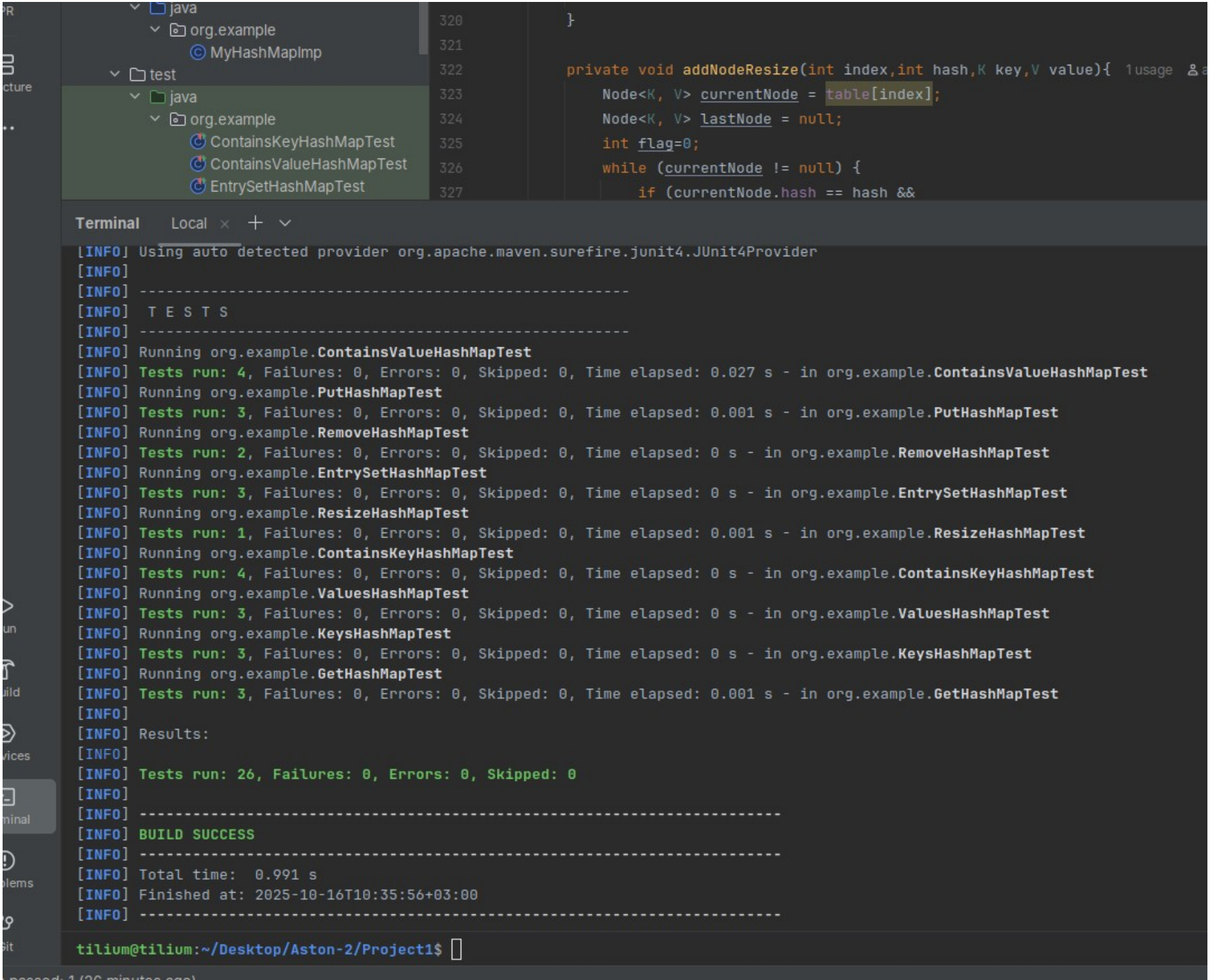

MyHashMapImp.java
EntrySetHashMapTest.java
ston-2/Project1
MapImp
sKeyHashMapTest
sValueHashMapTest
tHashMapTest
hMapTest
shMapTest
MapTest
HashMapTest
HashMapTest
HashMapTest
es
1 package org.example;
2
3 import org.junit.Test;
4
5 import java.util.Map;
6 import java.util.Set;
7
8 import static org.junit.Assert.assertEquals;
9 import static org.junit.Assert.assertNotNull;
10
11 public class EntrySetHashMapTest { anlikel
12 @Test anlikel
13 public void testEmptyHashMapSize(){
14 Map<Integer,String> map=new MyHashMapImp<>();
15 Set<Map.Entry<Integer,String>> entries=map.entrySet();
16 assertNotNull(map);
17 assertEquals(expected: 0,entries.size());
18 }
19
20 @Test anlikel
21 public void testFullHashMapSize(){
22 Map<Integer,String> map=new MyHashMapImp<>();
23 map.put(1,"a");
24 map.put(2,"b");
25 map.put(3,"c");
26 Set<Map.Entry<Integer,String>> entries=map.entrySet();
27 assertEquals(expected: 3,entries.size());
28 }
29
30 @Test anlikel
31 public void testDuplicateHashMapSize(){
32 Map<Integer,String> map=new MyHashMapImp<>();
33 map.put(1,"a");
34 map.put(2,"b");
35 map.put(3,"b");
36 Set<Map.Entry<Integer,String>> entries=map.entrySet();
37 assertEquals(expected: 3,entries.size());
38 }


MyHashMapImp.java
ResizeHashMapTest.java
ston-2/Project1
MapImp
sKeyHashMapTest
sValueHashMapTest
tHashMapTest
hMapTest
shMapTest
MapTest
HashMapTest
HashMapTest
HashMapTest
es
1 package org.example;
2
3 import ...
8
9 public class ResizeHashMapTest { anlikel
10 @Test anlikel
11 public void testResizePlusHashMap(){
12 MyHashMapImp<Integer,String> map=new MyHashMapImp<>();
13 for(int i=1;i<=20;i++){
14 Integer key=i;
15 String value="value_"+i;
16 map.put(key,value);
17 }
18 assertEquals(expected: 20,map.size());
19 boolean result=map.containsValue("value_19");
20 assertEquals(result, actual: true);
21 int tableSize=map.getTableSize();
22 assertEquals(expected: 36,tableSize);
23 }
24 }
25

Результат выполнения тестов:


Project
Project1 ~/Desktop/Aston-2/Project1
.idea
.mvn
src
main
java
org.example
MyHashMapImp
test
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
MyHashMapImp.java
5 public class MyHashMapImp<K,V> implements Map<K,V> { 27 usages anlikel
309 public void putResize(K key, V value) { 1 usage anlikel
317 else {
318 addNodeResize(index,hash,key,value);
319 }
320 }
321
322 private void addNodeResize(int index,int hash,K key,V value){ 1 usage
323 Node<K, V> currentNode = table[index];
324 Node<K, V> lastNode = null;
325 int flag=0;
326 while (currentNode != null) {
327 if (currentNode.hash == hash &&
Terminal
Local
tilium@tilium:~/Desktop/Aston-2/Project1$ mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< org.example:Project1 >------------------------
[INFO] Building Project1 1.0-SNAPSHOT
[INFO] --------------------------------[jar]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ Project1 ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/tilium/Desktop/Aston-2/Project1/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.11.0:compile (default-compile) @ Project1 ---
[INFO] Changes detected - recompiling the module! :source
[INFO] Compiling 1 source file with javac [debug target 11] to target/classes
[WARNING] system modules path not set in conjunction with -source 11
[INFO] /home/tilium/Desktop/Aston-2/Project1/src/main/java/org/example/MyHashMapImp.java: /home/tilium/Desktop/Aston-2/Project1/s
[INFO] /home/tilium/Desktop/Aston-2/Project1/src/main/java/org/example/MyHashMapImp.java: Recompile with -Xlint:unchecked for det
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ Project1 ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/tilium/Desktop/Aston-2/Project1/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.11.0:testCompile (default-testCompile) @ Project1 ---
[INFO] Changes detected - recompiling the module! :dependency
[INFO] Compiling 9 source files with javac [debug target 11] to target/test-classes
[WARNING] system modules path not set in conjunction with -source 11
[INFO] /home/tilium/Desktop/Aston-2/Project1/src/test/java/org/example/ValuesHashMapTest.java: /home/tilium/Desktop/Aston-2/Proje
[INFO] /home/tilium/Desktop/Aston-2/Project1/src/test/java/org/example/ValuesHashMapTest.java: Recompile with -Xlint:unchecked fo


java
org.example
MyHashMapImp
test
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
}
private void addNodeResize(int index,int hash,K key,V value){ 1 usage
Node<K, V> currentNode = table[index];
Node<K, V> lastNode = null;
int flag=0;
while (currentNode != null) {
if (currentNode.hash == hash &&
Terminal
Local
[INFO]
[INFO] --- maven-compiler-plugin:3.11.0:testCompile (default-testCompile) @ Project1 ---
[INFO] Changes detected - recompiling the module! :dependency
[INFO] Compiling 9 source files with javac [debug target 11] to target/test-classes
[WARNING] system modules path not set in conjunction with -source 11
[INFO] /home/tilium/Desktop/Aston-2/Project1/src/test/java/org/example/ValuesHashMapTest.java: /home/tilium/Desktop/Aston-2/Pro
[INFO] /home/tilium/Desktop/Aston-2/Project1/src/test/java/org/example/ValuesHashMapTest.java: Recompile with -Xlint:unchecked
[INFO]
[INFO] --- maven-surefire-plugin:3.0.0:test (default-test) @ Project1 ---
[INFO] Using auto detected provider org.apache.maven.surefire.junit4.JUnit4Provider
[INFO]
[INFO] ---
[INFO] T E S T S
[INFO] ---
[INFO] Running org.example.ContainsValueHashMapTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.027 s - in org.example.ContainsValueHashMapTest
[INFO] Running org.example.PutHashMapTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.001 s - in org.example.PutHashMapTest
[INFO] Running org.example.RemoveHashMapTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in org.example.RemoveHashMapTest
[INFO] Running org.example.EntrySetHashMapTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in org.example.EntrySetHashMapTest
[INFO] Running org.example.ResizeHashMapTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.001 s - in org.example.ResizeHashMapTest
[INFO] Running org.example.ContainsKeyHashMapTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in org.example.ContainsKeyHashMapTest
[INFO] Running org.example.ValuesHashMapTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in org.example.ValuesHashMapTest
[INFO] Running org.example.KeysHashMapTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in org.example.KeysHashMapTest
[INFO] Running org.example.GetHashMapTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.001 s - in org.example.GetHashMapTest
[INFO]
[INFO] Results:
[INFO]


java
org.example
MyHashMapImp
test
java
org.example
ContainsKeyHashMapTest
ContainsValueHashMapTest
EntrySetHashMapTest
320 }
321
322 private void addNodeResize(int index,int hash,K key,V value){ 1usage
323 Node<K, V> currentNode = table[index];
324 Node<K, V> lastNode = null;
325 int flag=0;
326 while (currentNode != null) {
327 if (currentNode.hash == hash &&
Terminal
Local
[INFO] Using auto detected provider org.apache.maven.surefire.junit4.JUnit4Provider
[INFO]
[INFO] ---
[INFO] T E S T S
[INFO] ---
[INFO] Running org.example.ContainsValueHashMapTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.027 s - in org.example.ContainsValueHashMapTest
[INFO] Running org.example.PutHashMapTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.001 s - in org.example.PutHashMapTest
[INFO] Running org.example.RemoveHashMapTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in org.example.RemoveHashMapTest
[INFO] Running org.example.EntrySetHashMapTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in org.example.EntrySetHashMapTest
[INFO] Running org.example.ResizeHashMapTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.001 s - in org.example.ResizeHashMapTest
[INFO] Running org.example.ContainsKeyHashMapTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in org.example.ContainsKeyHashMapTest
[INFO] Running org.example.ValuesHashMapTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in org.example.ValuesHashMapTest
[INFO] Running org.example.KeysHashMapTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in org.example.KeysHashMapTest
[INFO] Running org.example.GetHashMapTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.001 s - in org.example.GetHashMapTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 26, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] --
[INFO] BUILD SUCCESS
[INFO] --
[INFO] Total time: 0.991 s
[INFO] Finished at: 2025-10-16T10:35:56+03:00
[INFO] --
tilium@tilium:~/Desktop/Aston-2/Project1$