- Don't forget to set your Eclipse workspace and working set.
- You must submit the JAR file, exported (with source code), from your Eclipse project.
- You must check your JAR file to make sure all the source files (.java files) are present. It can be opened with file compression programs such as 7-zip or Winrar.
- Failure to export properly will result in your work not getting marked.

1) **To submit:**

   - **Export your project to a JAR file, with source code.**
   - **Name your JAR file ID_Week11_Q2.jar. For example, 6623110021_Week11_Q2.jar**
   - **Submit the JAR file on MyCourseville.**

(10 marks) You are given files for a double hashing hash table, including JUnit "TestHash.java".

Your task is to complete code in class **HashIterator**, which represents an iterator that marks a position in a hash table (it uses integer that represents the position). You have to implement the following methods:

- hasNext()
- hasPrevious()
- next()
- previous()

  **Hint:** implement hasNext() and next() together.
- Assume that:
  - Actual data in hash table must all be positive numbers.
  - So, 0 (which is a default value in integer array) and DELETED (which is -9999 in the given program) can never be actual data in the table.
  - Iterator does not go beyond the leftmost and the rightmost actual data in the array.
- When an iterator is created:
  - It marks the leftmost actual data in the array. For example, if we create an iterator for the hash table that contains array:

    | | | | | | | | | | | |
    |---|---|---|---|---|---|---|---|---|---|---|
    | 0 | 0 | 16 | 0 | 0 | 5 | 0 | 0 | 27 | 20 | 0 |

    The iterator will mark the array slot that contains 16.
- Method hasNext()
  - It checks that there is a next position to be marked. It does not go beyond the rightmost data in the array.
  - If there is a next position, return true.
  - Otherwise, return false.

  For example, in the table above:

    - If the iterator is marking 16, hasNext() will return true because the next data exists (it is 5).

        - If the iterator is marking 20, hasNext() will return false because there is no more actual data to the right.

- Method hasPrevious()
    - It checks that there is a previous position to be marked. It does not go beyond the leftmost data in the array.
    - If there is a previous position, return true.
    - Otherwise, return false.

    For example, in the table above:

        - If the iterator is marking 27, hasPrevious() will return true because the previous data exists (it is 5).
        - If the iterator is marking 16, hasPrevious() will return false because there is no more actual data to the left.

- Method next()
    - It checks that there is a next position to be marked. It does not go beyond the rightmost data in the array.
    - If there is a next position, mark that position and return data at that position.
    - Otherwise, throw an exception.

    For example, in the table

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 16 | 0 | 0 | 5 | 0 | 0 | 27 | 20 | 0 |

        - If the iterator is marking 16, next() mark the next data (which is 5) and will return 5.
        - If the iterator is marking 20, next() will throw exception because there is no more actual data to the right to be marked.

- Method previous()
    - It checks that there is a previous position to be marked. It does not go beyond the leftmost data in the array.
    - If there is a previous position, store value of the current position first.
    - Then mark the previous position
    - Return the stored value.
    - Otherwise, throw an exception.

    For example, in the table

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 16 | 0 | 0 | 5 | 0 | 0 | 27 | 20 | 0 |

        - If the iterator is marking 27, previous() mark the data 5 and will return 27.
        - If the iterator is marking 16, previous() will throw exception because there is no more actual data to the left to be marked.

Important:

- Only modify “HashIterator.java”. Your submission will not be marked if you modify other files.