Shokhrukh Nigmatillaev (APVAVZ)

Group : 2. apvavz@inf.elte.hu

Documentation for 1st assignment

Task: 7th

Implement the set type which contains integers. Represent the set as a sequence of its elements. Implement as methods: inserting an element, removing an element, returning whether the set is empty, returning whether the set contains an element, returning a random element without removing it from the set, returning the sum of the numbers in the set(suggestion: store the sum and update it when the set changes), printing the set. A set can store every element only once.

Set type

Set() = {List<int> | □*a* □ Z}

Integer List type

Operations:

1. **Inserting an element.** Inserting an element into the list.
   Formally :
   A = (List<int> list, number : ℝ)
   Pre = (number = number')
   Post = ((list'.Count == 0) ➞ *list = list ∪ { x } ∧ ¬(list'.Count == 0) ➞ ∀i ∈ [0, list'.Count-1]:(list'[i] ≠ x) ➞ list = list ∪ { x }*

   The operation needs to check if the set already consists of this number, if not then it will add it, if yes, it will skip it.

2. **Removing an element**. Removing particular element from the list.
   Formally :
   A = (List<int>, number : ℝ)
   Pre = (number = number')
   Post = ((|list'| = 0) = err) □ ¬(|list'| = 0) ➞ □i [i..|list'|-1] :
   (((list'[i] = number) ➞ (remove_element list'[i]))

   The operation needs to check if the entered number is in the set. If so then that number will be deleted, if not, nothing will be changed.

3. **Checking if the set is empty.** Checking is there any element in the list.
   Formally :
   Post = ((|list'| = 0)) ➞ isEmpty =TRUE □ ¬(|list'| = 0) ➞ isEmpty = FALSE

   The operation will check the quantity of elements in the List. If it is 0 then method will return true, if it is not 0 then false.
4. **Checking whether the set contains an element.** Checking if the particular element is in the set.

Formally :
A : (List<int>, number : ℝ, isIn: bool, m:ℤ, n:ℤ)
Pre : (number = number' ^ m = m', n = n')
Post : (Pre ^ $SEARCHi=m..n$ number)
The operation will check if the inputted number is in the set.

5. **Returning a random element without removing it from the set.**
   Formally :
   A : (List<int>, number : ℝ, m:ℤ, n:ℤ)
   Pre : (m = m', n = n')
   Post : ($SEARCHi=m..n$ ➞ List[random(I)])
   The operation will loop through whole list and return any random number from it.
6. **Returning the sum of the numbers in the set.**
   Formally :
   A : (List<int>, m:ℤ, n:ℤ, sum : ℤ)
   Pre : (m = m', n = n')
   Post : (Pre ^ sum = ∑ List(i) $i=m$ ..n)
   The operation will loop through the list and add each element to the sum.
7. **Printing the set.** The operation will loop through the list with the help of foreach loop and will print every element of it.
   Formally: A : List
   Post = i [i..| List |-1] : v ⊕ List[i]

Representation

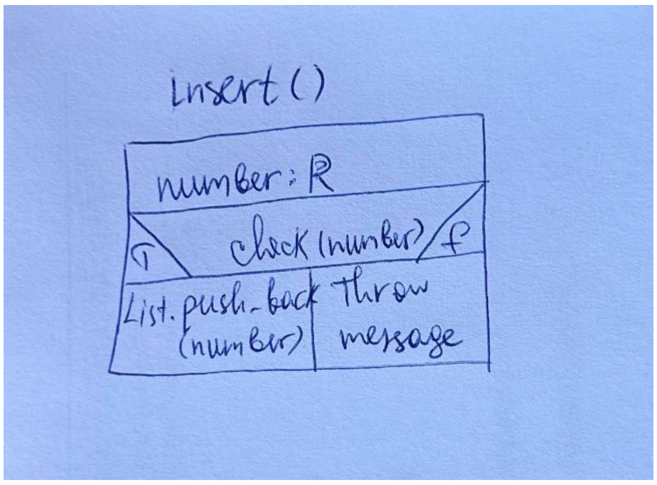
Only one integer type List should be stored.

Int list = <a1, a2, a3 … >

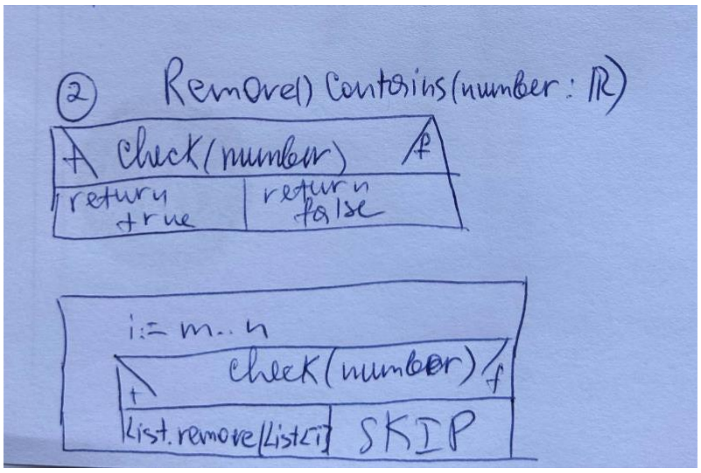
List(int) = <a1 if a == int
                    ∅ if a ¬ int >

Implementation

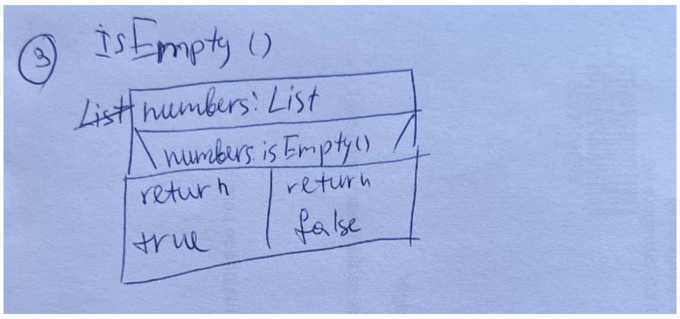
1. **Inserting an element.** Implementation will be as follows:



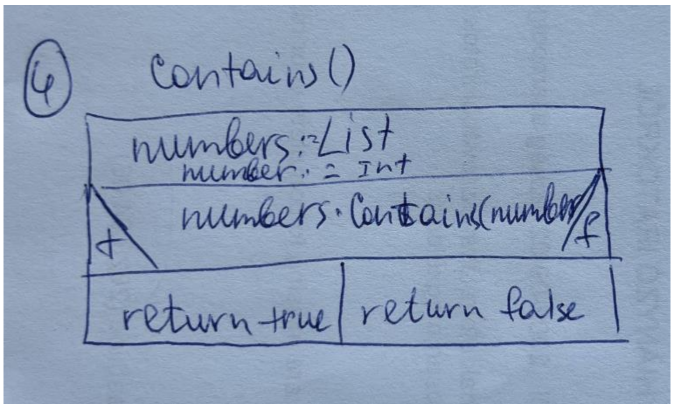
2. **Removing an element**. Implementation will be as follows:



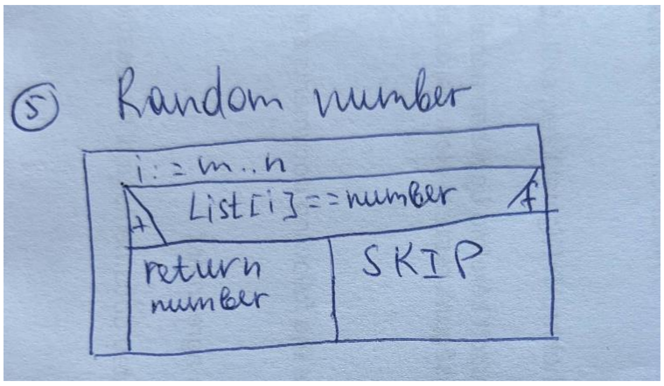
3. **Checking if the set is empty.** Implementation will be as follows :



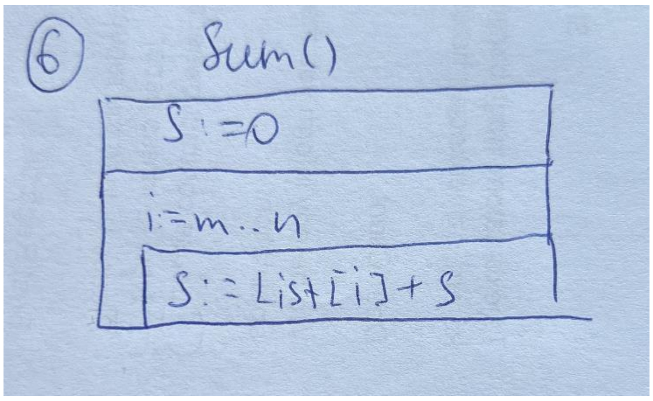
4. **Checking whether the set contains an element.** Implementation will be as follows:



5. **Returning a random element without removing it from the set.** Implementation will be as follows:



6. **Returning the sum of the numbers in the set.** Implementation will be as follows:



Testing

Testing the operations (**black box testing**)

1. Adding the element into the List.
   The List is empty, and the method adds the element into it, and returns true (the element is added).
2. Deleting the element from the list.
   2.1 The List initially is empty but then the method Add () adds into it the element, after we call method Delete (), it deletes that element from the list and returns false (the element is not in the list anymore).
   2.2 The List initially is empty, the method Delete () then deletes element from it and returns false. (The list is empty, and any element deleted from it will result *false*).
   2.3 The List is initially empty, then with loop and Add () method fills it with the elements, then method Delete () tries to delete non-existing in the List element, returning false. (the element was not in the list, so Delete () may not delete it).
3. Testing if the Set is empty.
   3.1 The List initially is empty, method CheckIsEmpty() checks whether List contains elements or not and returns true.
   3.2 The List initially is empty, method Add () adds element into it. Then method CheckIsEmpty() checks whether List contains any element or not and returns false.
4. Testing if the inputted element is in the set.
   4.1 The List is initially empty, method Add () adds element into it after that method CheckIfIn () checks whether the List contains that particular element, and returns true (the element is added and is in the list now).
   4.2 The List is initially empty, method Add () does not implemented here, after that method CheckIfIn () checks whether the List contains the particular element and returns false (the element is not added and is not in the list now).
5. Testing GetRadomNumber() function.
   5.1 The List is initially empty, with the help of foreach loop we add several elements into it and then GetRandomNumber() method returns one random element. We assign it to the separate variable and check if it is in the List, the method returns true.
   5.2 The List is initially empty, we initialize the int variable and make it equal to return value of GetRandomNumber() method. Then we check if this value is in the List, method returns false.
6. Testing GetSum() method.
   The List is initially empty, we also have array with elements and int variable sum, with

foreach loop we add elements from the array to the List and to the sum. Afterall, we check if the variable sum is equal to List.GetSum() value, the method returns true.