

## 1 Allen's Time Relations:

The program is written in Java Programming language and has been coded using the Eclipse IDE. To compile and run the program open the file in Eclipse IDE and click on the Run button in the menu of Eclipse. The inputs to the program are hardcoded as [5,10] and [7,13].

Timepoint is a datatype which is represented as a class and consists of a single attribute which is an integer. It does have a default constructor and a function `getdata()` which returns the value of the datatype.

Interval is another datatype which is represented as a class with two attributes which are two objects of the datatype Timepoint. It does have a default constructor which creates instances of Timepoint datatypes and assigns values to it.

In the main function two interval variables are created and all the predicate functions are called.

The `Start()` and `End()` functions takes an instance of Interval 'i' and returns the start Timepoint and End Timepoint respectively.

The function `meet` takes two instances of Interval as arguments and displays the output whether the `meet()` predicate holds or not. The function checks if the endpoint of interval i1 is equal to the startpoint of interval i2.

The function `before` takes two instances of interval as arguments and displays the output whether the `before()` predicate holds or not. The function checks whether the endpoint of interval i1 is less than the startpoint of i2.

The function `after` takes two instances of interval as arguments and displays the output whether the `after()` predicate holds or not. The function checks whether the endpoint of i2 is less than the startpoint of i1.

The function `during` takes two instances of interval as arguments and displays the output whether the `during()` predicate holds or not. The function checks whether the startpoint of i2 is less than or equal to the startpoint of i1 and the endpoint of i2 is greater than or equal to the endpoint of i1.

The function `overlap` takes two instances of interval as arguments and displays the output whether the `overlap()` predicate holds or not. The function checks whether the startpoint of i1 is less than startpoint of i2 and the endpoint of i1 is greater than the startpoint of i2. It also checks the viceversa of above condition.

The function `equals` takes two instances of interval as arguments and displays the output whether `equals()` predicate holds or not. The function checks whether the startpoint and endpoints of i1 are equal to the startpoint and endpoints of i2.

The function `finishes` takes two instances of `interval` as arguments and displays the output whether the `finishes()` predicate holds or not. The function checks whether the endpoints of `i1` and `i2` are equal or not.

The function `contains` takes two instances of `interval` as arguments and displays the output whether the `contains()` predicate holds or not. The function checks whether startpoint of `i1` is less than startpoint of `i2` and endpoint of `i1` is greater than the endpoint of `i2`.