1. **Use a method of a class from your project that can create a contract and describe its algorithm specification. Specify the pre- or post- condition and use both Structured English and an activity diagram to specify the algorithm.**

* **Contract**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Method Name:** | run() | | **Class Name:** | CheckOverdueBook | **ID:** |  | |
| **Client(consumers):** | | Time | | | | |
| **Associated Use Case:** | | | | | | |
| Member | | | | | | |
| **Description of Responsibilities:** | | | | | | |
| We use checkOverdueBook to calculate if the book that member borrowed is overdue or not. | | | | | | |
| **Arguments Received:** | | | | | | |
|  | | | | | | |
| **Pre-Conditions:** | | | | | | |
| day=(now.getTime() - beginDate.getTime())/(24\*60\*60\*1000) | | | | | | |
| **Post-Conditions:** | | | | | | |
| **if**(day<=3)  bookState = ("notice");  setnumberOfNoticeBook -1  getnumberOfNoticeBook +1    **else** **if**(day<0)  bookState = ("overdue");  getnumberOfNoticeBook -1  getnumberOfOverdueBook +1  setright = (**false**) | | | | | | |

* **Method Specification**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Method Name: Time()** |  | **Class Name:** | Time | | **ID:** |  |
| **Contract ID:** |  | **Programmer:** | Kendy | | Data Due: | 05/28 |
| **Programming Language:** | | | | | | |
| Java | | | | | | |
| **Triggers/Events:** | | | | | | |
| CheckOverdueBook task = new CheckOverdueBook(); | | | | | | |
| **Arguments Received:**  **Arguments Data Type:** | | **Notes:** | | | | |
|  | |  | | | | |
|  | |  | | | | |
| **Messages Sent & Argument Passed:**  **ClassName.MethodName:** | | **Data Type:** | | **Notes:** | | |
|  | |  | |  | | |
|  | |  | |  | | |
|  | |  | |  | | |
| **Arguments Returned:**  **Data Type:** | | **Notes:** | | | | |
|  | |  | | | | |
|  | |  | | | | |
| **Algorithm Specification:** | | | | | | |
| **if**(day<=3)  {  store.setbookState("notice");    OM.setnumberOfNoticeBook(OM.getnumberOfBorrowBook()-1);  OM.setnumberOfNoticeBook(OM.getnumberOfNoticeBook()+1);  SEP.add(store);  }  **else** **if**(day<0)  {  store.setbookState("overdue");    OM.setnumberOfNoticeBook(OM.getnumberOfNoticeBook()-1);  OM.setnumberOfNoticeBook(OM.getnumberOfOverdueBook()+1);  OM.setright(**false**);  } | | | | | | |
| **Misc.Notes:** | | | | | | |
|  | | | | | | |

* **Activity Diagram**