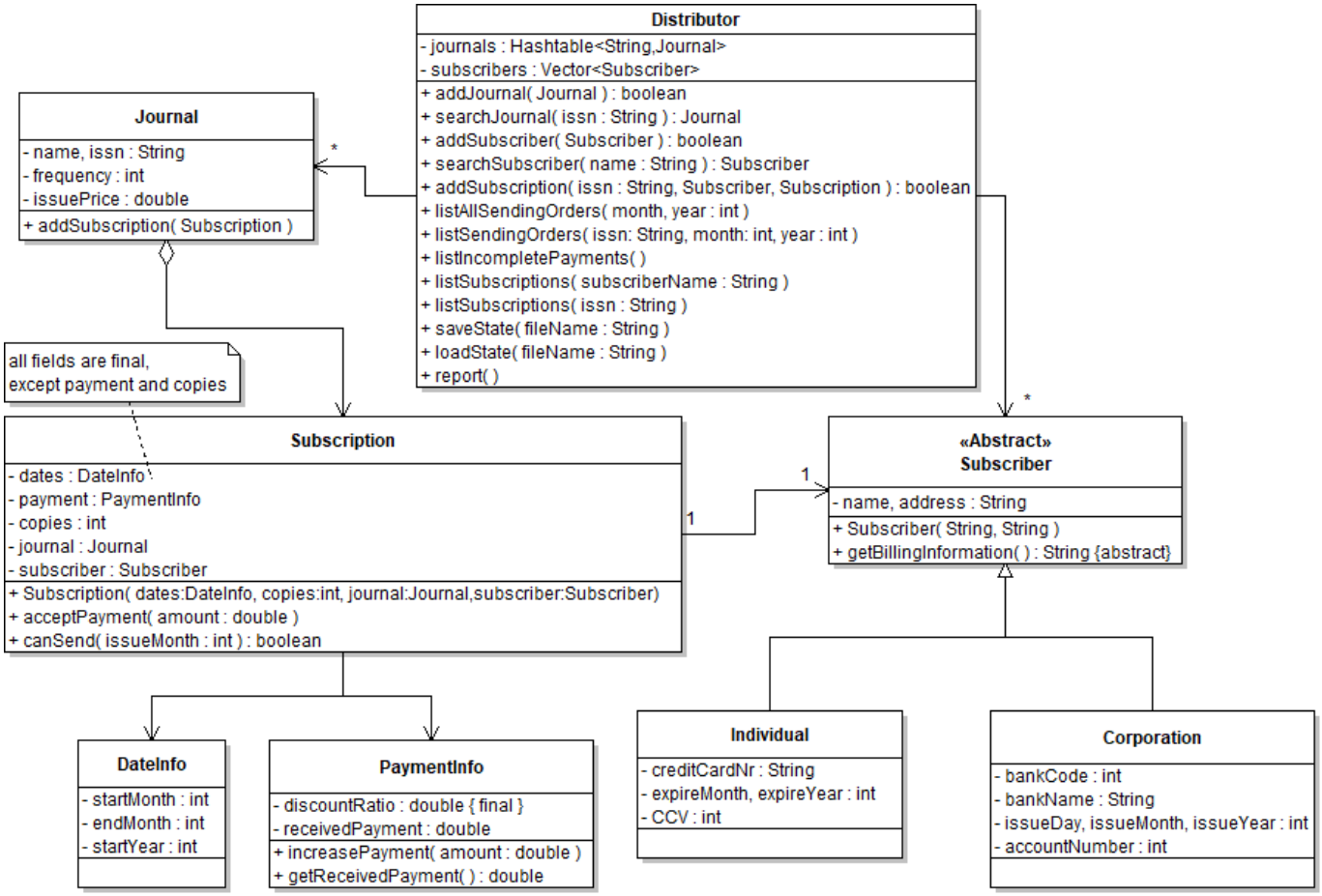
**BLM2012 Object Oriented Prog., Course Project: A Journal Distribution Information System**

**Due 08.01.2024, Monday, 23:59 PM**




In order to have a common ground for coding, testing and evaluation, the main UML class diagram of the business logic is given above. You may need to extract hidden information from the schema and add necessary code while implementing your project. Don't forget to handle multithreading issues. You are required to add a graphical user interface and unit tests. Here are some details about the business logic:

**Problem Domain:** You are writing software for a distributor company that sells journals to subscribers. Individuals pay for their subscription by credit card and corporations pay by bank cheques. A subscriber can pay in partial amounts; therefore the payment that has been received so far is kept in a PaymentInfo object. A subscriber can subscribe to multiple copies of the same journal. A subscription continues for one year but it can begin in any month. For example, a subscription may begin in March 2023 and this means that it will end in February 2024 (inclusive). The ISSN is a unique identifier of a journal. The frequency of a journal denotes how many issues are published within a year. Each journal has an issue price but subscribers can be granted a discount ratio.

**Additional Information:**

- addSubscription method: A subscription for a non-existent subscriber or journal must not be created. If a subscription object for the given subscriber and journal exists, its copies field will be increased by one in this method.
- listAllSendingOrders method: Lists which journals will be sent to which subscribers for a given month and year. We must not send journals that have not been paid for. You can use the result of the canSend method for this purpose.
- canSend method: Determines whether a particular issue of its journal can be sent to its subscriber. If enough payment has been received so far, the method returns true. The acceptPayment method does not interact with a subscriber; it just increases the received payment so far.

- saveState and loadState methods: These methods accept a file name and executes their namesake tasks on entire objects. They should wait for the report method to complete its tasks if it is running.
- report method: This should be run as a multithreaded task run in a separate thread than the application and it generates a report that preferably includes some charts about the items given below. Its output can be displayed in the GUI or it can be forwarded to an external HTML file.
    - Subscriptions that will be expired after a given date.
    - Received annual payments in a given year range.

**What you are expected to do:**

1. You shall implement all the classes given in the UML class diagram.
2. You need to add GUI functionality to your project.
3. You are required to add unit tests to your project, too. There must be at least two jUnit test cases for each class depicted as having member fields in the main UML class diagram.
4. This is an individual project. You are required to do your project on your own. Your code is subject to be checked against other student's code. Cheating is not allowed and will not be tolerated.
5. Projects are due at 11:59pm on the date specified. A grading penalty will be applied to late assignments. (10% penalty up to the first 24 hours, 20% for 24 to 48 hours, with no credit received after that)
6. Submitting a project is mandatory (Submitted projects will be demonstrated according to a schedule that will be announced later).
7. Submissions will be done into two separate areas at Google Classroom:
    - The first area is for submission of your source codes and unit tests as a zip file.
    - The second area is for submission of the single executable jar file of your project.
    - If you use a framework other than the javax.swing library, it must not need a separate installation and it must be included in both the zip and jar files.
    - You must compress all documents and name it with your student number. You must submit only this compressed file.
        - Etc: 2101123**.zip**