Приложение должно эмулировать работу настоящего банкомата,

его взаимодействие с Клиентом и Банком.

Символ "\*" обозначает опциональные требования.

Символ "!" обозначает обязательные требования.

! По крайней мере одно из опциональных требований должно быть реализовано.

Разработка функциональности свыше требований приветствуются и оценивается.

Задание выполняется на Java SE (без использования механизмов Java EE)),

**данные (Счет Клиента, количество купюр по номиналам) хранятся** в БД (любая бесплатная БД, например mySQL),

вариант - **банкомат может хранить свою информацию о купюрах в текстовом (\* вариант в JSON формате) файле.**

Приложение должно содержать главный класс (сценарий), демонстрирующий выполняющие всех возможных

операции взаимодействия Клиент - Банкомат.

Вывод результатов каждой операции производится в консоль (лог).

**! все ошибки клиента и банкомата должны обрабатываться.**

\* приложение может содержать графическую часть (веб- интефейс), который либо демонстрирует все операции,

либо, в более простом варианте, позволяет пользователю просматривать свои счета, и оплачивать платежи.

\* можно учесть ситуацию с банкоматом одного банка, карточкой другого банка и

принадлежностью карточки к одной из платежных систем. Описание деталей такого взаимодействия общедоступно.

\* можно добавить транзакции в приложение

Приложение должно содержать классы, описывающие, как минимум, следующие сущности:

**Клиент**

**Снимает деньги со счета (часть или все сразу), кладет деньги на счет,**

\* оплачивает платежи.

**Карточка клиента**

**(со своим PIN кодом и номером, привязанным к счету в Банке)**

**! пинкод не хранится на карточке, следовательно банкомат не может его считать напрямую**

\* пинкод не хранится в банке в явном (незашифрованном, нехэшированном) виде

**Банкомат**

**Выдает Клиенту деньги по Карточке, списывая их со Счета Клиента в Банке**

\* Позволяет оплачивать зарегистрированные в Банке платежи, с точностью до копейки.

\* Реализует конечный автомат с таблицей переходов

Следит за расходом купюр, соблюдая пропорции.

**! Содержит и выдает купюры, кратные 100 рублям.**

**! За раз выдаёт не более 40 купюр.**

**! Не выдает клиенту денег больше, чем есть на его счете.**

**! Принимает у Клиента деньги на счет.**

**! Информирует пользователя о результате каждой операции.**

**Проверяет, валидность срока действия карточки.**

**Запрашивает ПИН код у Клиента при начале сеанса.**

**Реализует другие очевидные операции.**

\* Рассмотреть выделение внутренних сущностей банкомата (диспенсер, принтер, купюроприменик) в отдельные классы

\* Если в приложении используются транзакции, предлагается рассмотреть

вовлечение вышеупомянутых сущностей в эти транзацкии

**Банк**

**Содержит счета клиентов и счета компаний - получателей платежей.**

**Содержит информацию о привязке номеров карточек к номерам счетов.**

\* взаимодействие с Банком производится через удаленный вызов процедур (RMI лидо веб-сервис, либо сокет).

\* содержит механизм проверки пинкода