

Шестая задача ООП

Предполагается использование следующих паттернов ООП:

- Билдер
- Фабрика
- Шаблонный метод
- Декоратор
- Цепочка обязанностей

В банке есть Счета и Клиенты. У клиента есть имя, фамилия, адрес и номер паспорта (имя и фамилия обязательны, остальное – опционально, клиент строится Билдером). Счета бывают трёх видов: Текущий счет, Депозит и Кредитный счет. Каждый счет принадлежит какому-то Клиенту.

Текущий счет – обычный счет с фиксированным процентом на остаток. Деньги можно снимать в любой момент, в минус уходить нельзя. Комиссий нет

Депозит – счет, с которого нельзя снимать деньги до тех пор, пока не закончится его срок. Процент на остаток зависит от изначальной суммы. Комиссий нет.

Кредитный счет – имеет кредитный лимит, в рамках которого можно уходить в минус (в плюс тоже можно). Процента на остаток нет. Есть фиксированная комиссия за использование, если клиент в минусе.

Счета создаются с помощью Фабрики, которой известны величины процентов и комиссии. Фабрика возвращает объекты только общего типа Счет, при этом реальный тип либо задается явно, либо по косвенным признакам (кредитный лимит может быть только у Кредитного счета, срок – только у Депозита и так далее)

Общий интерфейс счета предполагает интерфейс снятия, пополнения, и перевода денег (переводить можно только между счетами одного клиента). Для реализации этих методов предлагается использование паттерна Шаблонный метод

Если при создании счета у Клиента не указаны адрес или номер паспорта, мы объявляем такой счет (любого типа) сомнительным, и запрещаем операции снятия и перевода выше определенной суммы. Для реализации этой функциональности предлагается использовать паттерн Декоратор

Также счетам могут приходить запросы – например, запрос на выплату процентов или запрос на вычет комиссии (здесь не надо никакой сложной логики – если пришел запрос, просто начисляем указанный в счете процент или комиссию, никаких проверок на то, было ли что-то уже начислено/снято, не нужно). Для обработки запросов предлагается использовать паттерн Цепочка обязанностей

Ко всем этим требованиям можно, как всегда, относиться гибко. От вас требуется применение паттернов проектирования и защита своего решения. Обращу внимание, что вся эта модель очень условная и, например, совсем не требует каких-то проверок безопасности, ролей и т.п.