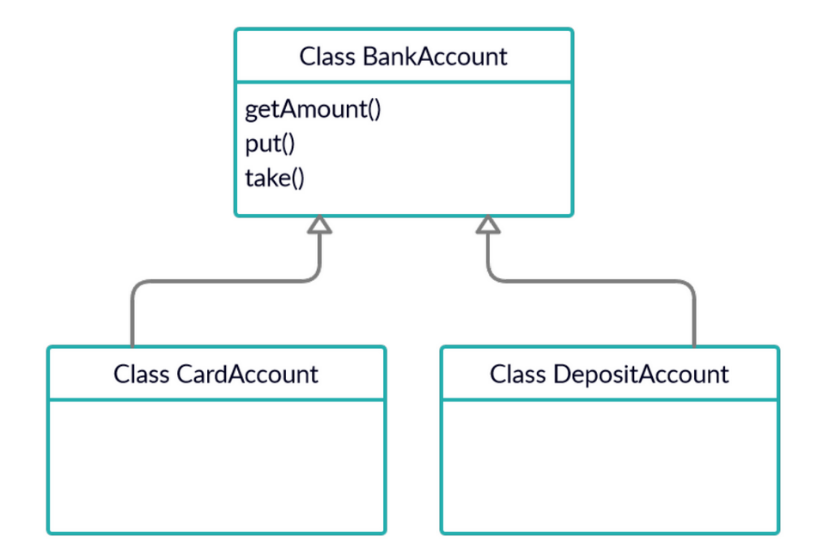
**Домашняя работа Inheritance**

#### **Задание №1**

##### **Цель**

Освоить на практике наследование классов.

##### **Что нужно сделать**

* В проекте найдите класс BankAccount и унаследованные от него классы CardAccount и DepositAccount.
* Реализуйте методы классов, при необходимости переопределите методы в наследниках так, чтобы выполнялись условия пополнения и снятия:
  + **BankAccount** — пополнение и списание происходит без комиссии. Если передать в метод пополнения отрицательное значение, сумма на счёте не должна измениться. При попытке снять большую сумму, чем есть на счёте, сумма не списывается со счёта, сумма на счёте не изменяется.
  + **CardAccount** — карточный счёт, в дополнение к условиям BankAccount — при снятии денег должна взиматься комиссия 1% от суммы списания. Пример: при попытке снять со счёта 100 рублей должен списаться 101 рубль.
  + **DepositAccount** — депозитный расчётный счёт, в дополнение к условиям BankAccount — нельзя снимать деньги в течение одного месяца после последнего пополнения. Переменную, в которой хранится дата последнего внесения, назовите lastIncome. Тип переменной используйте Calendar или LocalDate.  
    

#### **Задание №2\***

##### **Цель**

Научиться выбирать подходящие модификаторы доступа.

##### **Что нужно сделать**

Используйте код предыдущего задания. Установите в коде проекта подходящие модификаторы доступа у методов и переменных классов BankAccount, CardAccount и DepositAccount.

##### **Критерии оценки работы**

Принято:

* для всех переменных и методов установлены подходящие модификаторы доступа.

##### **Материалы для изучения**

* [Наследование в Java: классов, методов.](https://hr-vector.com/java/nasledovanie)
* [Наследование в Java. Доступ к членам класса. Многоуровневая иерархия.](https://www.examclouds.com/ru/java/java-core-russian/nasledovanie)
* [Неявное наследование всех классов java.lang.Object. Методы.](https://urvanov.ru/2016/04/08/java-8-%d0%bd%d0%b0%d1%81%d0%bb%d0%b5%d0%b4%d0%be%d0%b2%d0%b0%d0%bd%d0%b8%d0%b5/#javalangobject)
* [Чем отличаются наследование и композиция в Java.](https://tproger.ru/translations/inheritance-and-composition-in-java/)
* [Полиморфизм на примерах.](http://shkoshka.blogspot.com/2014/06/java.html)