```java
public abstract class eWalletADT {
    /**
     *
     * postconditions: reads a pin from std-input
     * @return that pin
     */
    abstract int getPin();
    /**
     * preconditions: sin>0
     * postconditions: set the pin to the wallet
     * @param pin sets the pin to the wallet
     */
    abstract void setPin(int pin);
    /**
     * preconditions: pin>0
     * postconditions: unlocks the wallet if the pin is correct
     * @param pin set the unlock pin to the wallet
     */
    abstract boolean unlock(int pin);
    /**
     * preconditions: newPin>0 , oldPin>0
     * postconditions: change the current pin with a new one if the old is correct
     * @param newPin the new pin to be set
     * @param oldPin the old pin that the wallet had
     * @return true if the old pin was correct and the new pin has been setted
     * false otherwise
     */
    abstract boolean changePin(int newPin, int oldPin);
    /**
     * preconditions: add>0
     * postcondition: add money to the wallet if the pin is correct
     * @param add the money to be added in the wallet
     * @return true if the pin was correct and the wallet had space for them and
     * the money had been added, false otherwise
     */
    abstract boolean addMoney(double add);
    /**
     * postconditions: print the money that the wallet has at this point if the pin is correct
     * @return true if he the money that the wallet has at this point if the pin was correct
     */
    abstract boolean seeMoney();
    /**
     * preconditions: cost>0
     * postconditions: remove money from the wallet if the pin is correct
     * @param cost the price the needs to be removed from the wallet
     * @return true if the wallet had this money and the pin was correct
     * false otherwise
     */
    abstract boolean buy(double cost);
}
```