

```

import java.util.Observable;

/**
 * Die Klasse Meilenkonto stellt die Funktionalitaet eines vereinfachten
 * Meilenkontos bereit. Diese beinhaltet das Hinzufügen von Meilen, das Abbuchen von
 * Meilen
 * (withdraw) sowie die Berechnung der Zusatzmeilen (ohne Beruecksichtigung der
 * Kontobewegungen). Beobachter koennen sich beim Meilenkonto anmelden, um
 * Rueckmeldungen
 * ueber Veränderungen zu bekommen.
 */
abstract class MilageAccount extends Observable {
    /**
     * Aktueller Stand des Meilenkontos
     */
    public float balance;

    /**
     * Kontonummer des Meilenkontos, entspricht der Vielfliegernummer des Inhabers
     */
    private int accountNo;

    /**
     * Konstruktor
     * PRE: true
     * POST: true
     */
    public MilageAccount(int accountNo) {
        this.accountNo = accountNo;
    }

    /**
     * Diese Methode berechnet die Zusatzmeilen welche es fuer die gegebene Anzahl
     * geflogener Meilen gibt und fügt diese dem Meilenkonto hinzu.
     * Anschliessend werden alle Beobachter dieses Konto
     * ueber die Veraenderung des Saldos informiert.
     *
     * PRE true
     * POST balance == balance@PRE + calculateInterests()@PRE
     */
    public void accumulateAdditionalMiles(float amount) {
        float oldBalance = balance;
        balance += calculateAdditionalMiles(amount);
        setChanged();
        notifyObservers(new Float(oldBalance));
    }

    /**
     * Fuegt eine gewisse Anzahl Meilen (amount) dem Meilenkonto hinzu
     * Allfaellige Beobachter werden ueber die Saldoaenderung
     * informiert. Der Betrag darf nicht kleiner sein als null.
     *
     * PRE: amount >= 0
     * POST: balance == balance@PRE + amount
     */
    public void deposit(float amount) {
        float oldBalance = balance;
        balance += amount;
        // Beobachter benachrichtigen
        setChanged();
        notifyObservers(new Float(oldBalance));
    }

    /**
     * Gibt die Kontonummer zurueck.
     * PRE: true
     * POST: result == accountNo

```

```

    */
    public int getAccountNo() {
        return accountNo;
    }

    /**
     * Gibt den den Hashcode dieses Objektes zurueck. Dieser
     * ergibt sich aus der (eindeutigen) MeilenKontonummer.
     * PRE: true
     * POST: result == accountNo
     */
    public int hashCode() {
        return accountNo;
    }

    /**
     * Gibt die Informationen (MeilenKontonummer und Saldo) ueber das MeilenkontoKonto
als String
     * zurueck.
     * PRE: true
     * POST: result != null and !result.equals("")
     */
    public String toString() {
        return "Account " + accountNo + " has balance of " + balance;
    }

    /**
     * Eine Gewisse Anzahl Meilen (amount) werden dem Meilenkonto abgezogen.
     * Der Betrag darf nicht kleiner sein als null.
     * Allfaellige Beobachter des Kontos werden ueber
     * die Saldoaenderung informiert.
     *
     * PRE: amount >= 0
     * POST: balance == balance@PRE - amount
     */
    public abstract void withdraw(float amount);

    /**
     * Berechnet die Zusatzmeilen die es fuer die gegebene Anzahl geflogener Meilen
gibt
     * PRE: true
     * POST: true
     */
    public abstract float calculateAdditionalMiles(float amount);
}

```