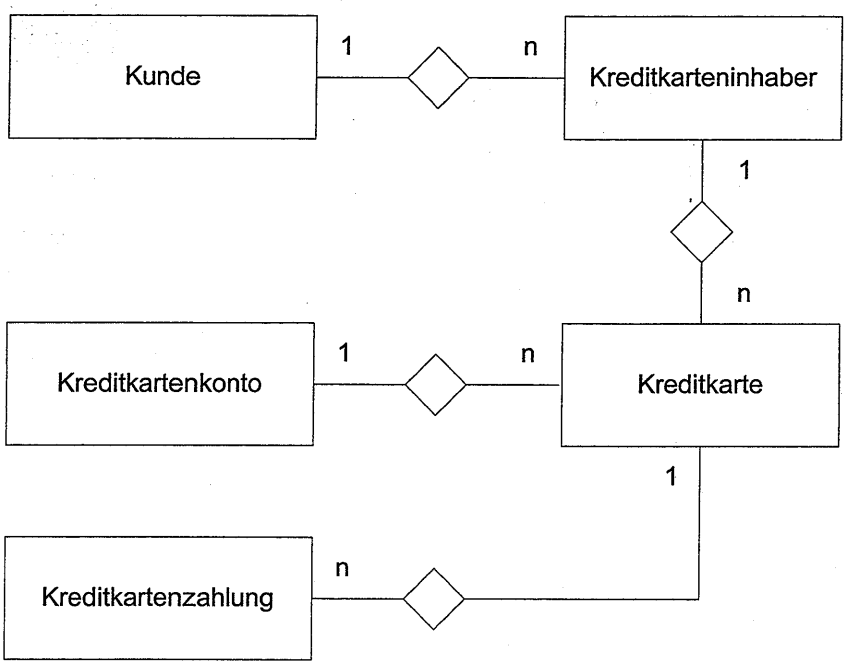
## 1. Handlungsschritt (25 Punkte)

a) 10 Punkte




ba) 5 Punkte

Die „Dritte Normalform" beinhaltet, dass Tabellen in Beziehung stehen und dadurch redundante Daten vermieden werden. Durch die Speicherung des Gesamtbetrages wird ein Wert gespeichert, welcher bereits aus den Tabellen der Buchungen hervorgeht.

bb) 5 Punkte

Bei großen Datenvorkommen wird Redundanz als sinnvolles Werkzeug eingesetzt, da Zugriffe und Abfragen minimiert werden können. In diesem speziellen Fall liegen zu berechnende Werte vor, der Zugriff kann daher ohne Summierung und zusätzlichem Verweis stattfinden und wird dadurch schneller ausgeführt.

bc) 5 Punkte

Eine Transaktion umfasst Arbeitsschritte, deren Ergebnisse erst nach Abschluss der Transaktion in die Datenbank übernommen werden. In diesem Fall die Speicherung des Buchungsbetrags und dessen Addition zum Gesamtbetrag. Es müssen immer beide Aktionen durchgeführt werden, da sonst eine Differenz zwischen der Summe der gebuchten Einzelbeträge und der separat gespeicherten Gesamtsumme entstehen.

## 2. Handlungsschritt (25 Punkte)

Luhn-Algorithmus

```
function checkLuhn(string kkn)
{
    int sum :=0
    int AnzahlZiffern := Länge von kkn
    for i from 0 to AnzahlZiffern – 2
        {
        int ziffer := integer(kkn[i])
        if i modulo 2 = 0
            ziffer := ziffer * 2
        if ziffer > 9
            ziffer := ziffer - 9
        sum := sum + ziffer
        }
        int sum2 := sum
        if sum modulo 10 <> 0
            sum2 := sum + 10 – sum modulo 10
        return (sum2 – sum) = integer(kkn[AnzahlZiffer - 1])
}
```

## 3. Handlungsschritt (25 Punkte)

a) 6 Punkte

```
SELECT Artikel.Artikelbezeichnung, MIN(Preis)
FROM Artikel, Artikelpreis
WHERE Artikel.ArtikelNr = Artikelpreis.ArtikelNr
    AND von_Datum >= ‚01.01.2007‘ AND bis_Datum <= ‚31.12.2007‘
GROUP BY  Artikelbezeichnung
```

b) 9 Punkte

```
SELECT Kunde.KundenNr, SUM(Einkaufsposition.Menge),
    AVG(Einkaufsposition.Verkaufspreis)
FROM Einkaufsposition, Einkauf
WHERE
    Einkaufsposition.EinkaufsNr = Einkauf.EinkaufsNr
GROUP BY KundenNr
ORDER BY 2 DESC
```

c) 10 Punkte

```
UPDATE Einkauf E
SET Gesamtbetrag =
    (SELECT SUM(Verkaufspreis * Menge)
    FROM Einkaufsposition
    WHERE Einkaufsposition.EinkaufsNr = E.EinkaufsNr)
```

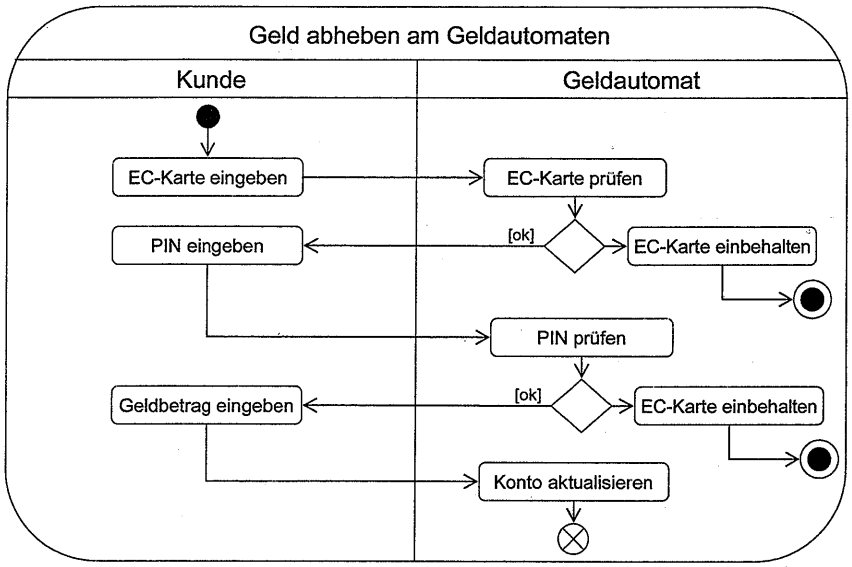
## 4. Handlungsschritt (25 Punkte)

```
Artikel artikel1 = array[0], artikel2 = array[0], artikel3 = array[0]

für i = 1 bis länge von array – 1
    wenn (artikelAnzahl.get(array[i]) > artikelAnzahl.get(artikel1))
        artikel3 = artikel2
        artikel2 = artikel1
        artikel1 = array[i]
    sonst wenn (artikelAnzahl.get(array[i]) > artikelAnzahl.get(artikel2))
        artikel3 = artikel2
        artikel2 = array[i]
    sonst wenn (artikelAnzahl.get(array[i]) > artikelAnzahl.get(artikel3))
        artikel3 = array[i]
    ende wenn
ende für
```

## 5. Handlungsschritt (25 Punkte)

**Aktivitätsdiagramm – Bankautomat**




**Aktivitätsdiagramm – Bankautomat**