Fuktionsprototypen informieren wir den Compiler über den Aufbau einer Funktion, die an einer anderen Stelle definiert ist. Wir deklarieren also die Funktionen und die Prüfziffer liefert einen int wert zurück an die aufrufende Funktion.

Main: Die Funktionen werden in der Reihenfolge aufgerufen, in welcher Sie laufen sollen.

Deklarieren von der Variablen dPuffer und intialisieren mit 0

Solange auf wahr läuft das Programm weiter.

Wir haben keinen Verweis auf die Main gemacht.

1. Pruefzifferberechnung
2. Benutzeroberfläche
3. Dann wird dPuffer auf den Wert der Geldforderung wird von der Auswahlfunktion zurückgegeben
4. Dann wird dPuffer auf Wert des Restgeldes gesetzt
5. Diese wird dann in der Funktion CoinOutput benötigt.
6. Mit der Funktion OutputItem wird sich für den Einkauf bedankt
7. vCloseApplication ist für das Auswahlmenü, ob ein weitere Einkauf gewünscht ist da.

**Int Pruefziffer**

Variablendeklaration und initialisierung der Variablen mit 0

Do While-Schleife – Mindestens ein Durchlauf

Arraydeklaration

Tastaturpuffer wird wieder geleert.

Eingabe der Kundennummer die wird dann in der Funktion Charpointertoint

In int umgewandelt anhand eines Switch-Case

Dann findet die Berechnung statt, die wir anhand vor Vorgabe berechnen nach d Modulo10 Verfahren.

Dann geht’s mit einer Wenn die Pruefzahl gleich Pruefnummer ist dann lösch den Bildschirm

Und setzt iWeiter auf 0 Falls nicht setzte i Weiter auf 1

Und das ganze solange i gleich 1

Rückgabewert ist die Pruefzahl wobei wir die eigentlich gar nicht mehr brauchen

**Void Benutzeroberlaeche**

Ausgabe der Benutzeroberfläche

dPuffer wird auf den Wert des Rückgabewerts der Funktion Auswahl gesetzt

**double Auswahl**

Deklarieren der Variablen Kategorie1, Geldforderung und Kategorie2 mit 0

Do While Schleife – Solange OuterLoopFailure = 1 wiederhole die Ausgabe (Bei Falscheingabe, geht’s wieder von vorne los)

Switch-Case und bei Fall 1 auch wieder eine Do-While Schleife mit der Variablen OuterLoopFailurle

Die Geldforderung wird auf den jeweiligen Wert der Auswahl gesetzt und als Double zurückgegeben.

**double dGeldEinwurf** (double Geldforderung) <- Geldforderung kommt aus der Funktion Auswahl

Wir deklarieren die Variablen iInput und Muenzwert und initialisieren Sie mit 0

Dialog welche Taste für wieviel Cent steht.

While Schleife: Solange die Geldforderung größer ist als der Muenzwert führe aus

Fflush(stdin); leert den Tastaturpuffer

Eingabe

Fallunterscheidung mit switch – case Je nach Taste wird eben die Variable Muenzwert erhöht

Bei Falscher Eingabe geht er zurück

Wenn dann Muenzwert größer ist als die Geldforderung springt er raus aus der Schleife und

Zeigt uns an, was wir Eingezahlt haben.

Den Münzwert (Restgeld) gibt er zurück.

Dieser wird dann in dPuffer gespeichert und an CoinOutput gegeben.

**void CoinOutput**

Die Variable iCurrentMoney wird berechnet = Umrechnung von Euro in Cent

**vCloseApplication**

Die Variable iStatus wird mit 0 initialisiert

Ausgabeauswahl Programm beenden

Eingabe iStatus

Wenn iStatus gleich 1 dann clear screen. Und das Programm startet von vorne.

Ansonsten wird das Programm geschlossen.