UNIKASSEL VERSITÄT

Fachgebiet für Kommunikationstechnik

Prof. Dr.-Ing. Klaus David

ComTec

Communication Technologies 2

Anwendungsfälle

Dipl.-Inf. Daniel Wilhelm
Kassel, 17.02.2014

Übersicht

ComTec

- Tabellen
- Internetanfragen
- Sensoren

- Quellen

ComTec

Anwendungsfälle

TABELLEN

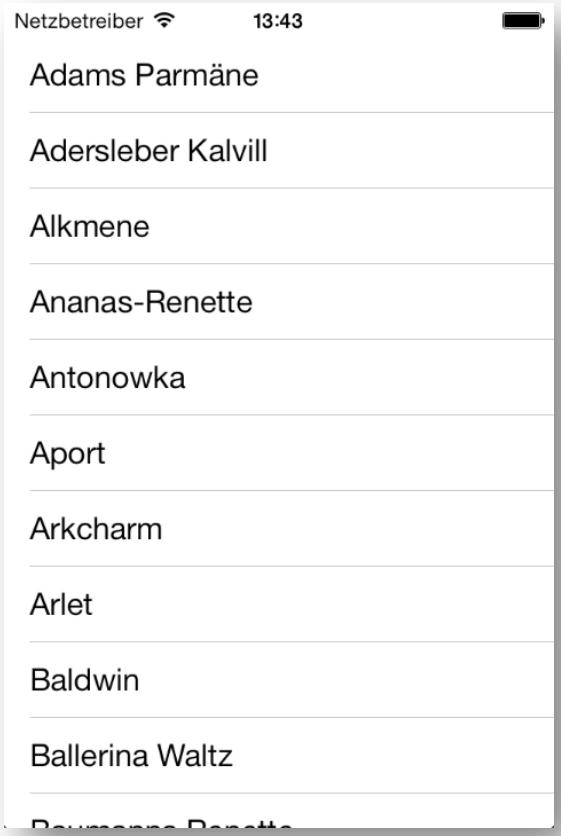
Übersicht Tabellen

ComTec

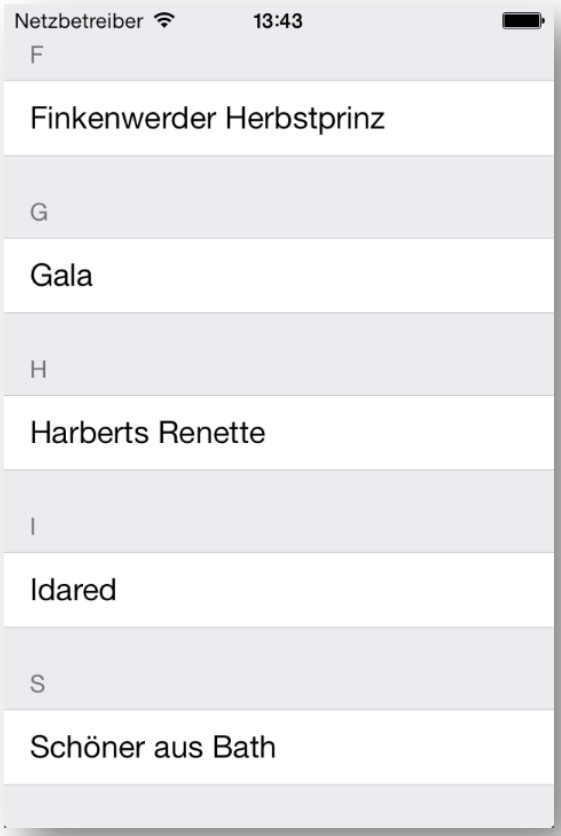
- Tabellenstile
- Tabellenaufbau
- Zellenaufbau
- Zellenstile
- Accessory-Stile
- Datenquelle
- Reaktion auf Berührung

Tabellenstile

ComTec



UITableViewStylePlain

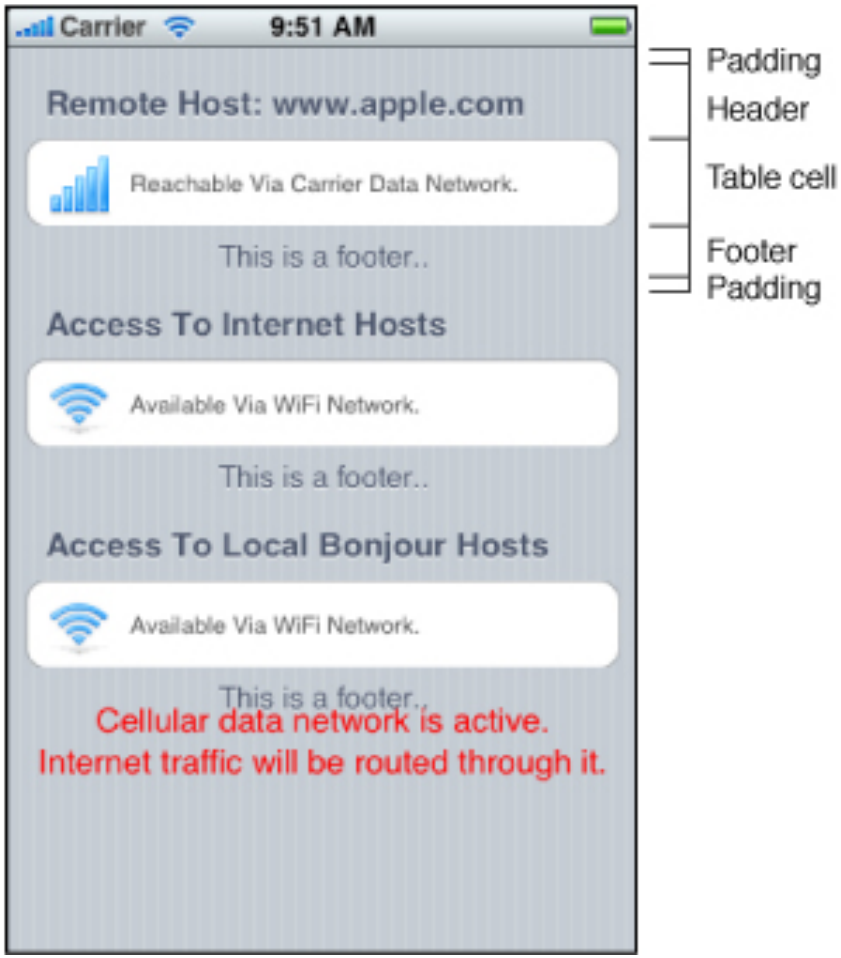


UITableViewStyleGrouped

Quelle: [1]

Tabellenaufbau

ComTec

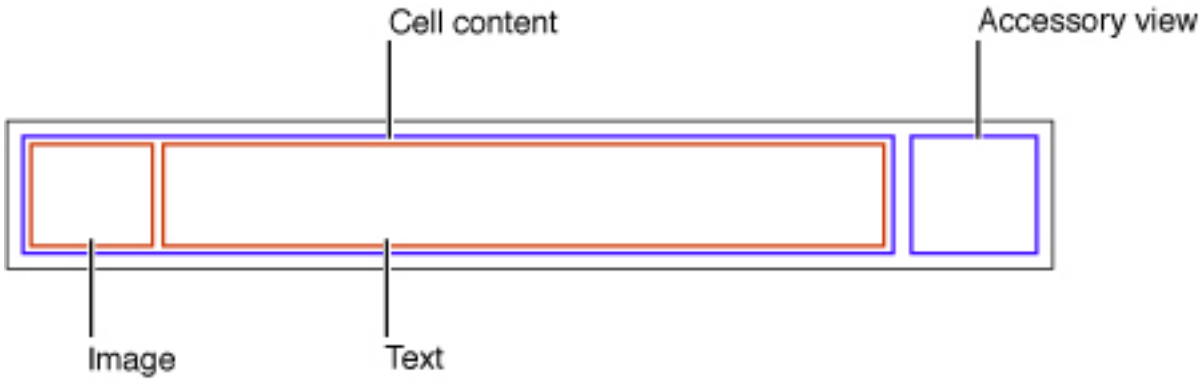


Quelle: [2]

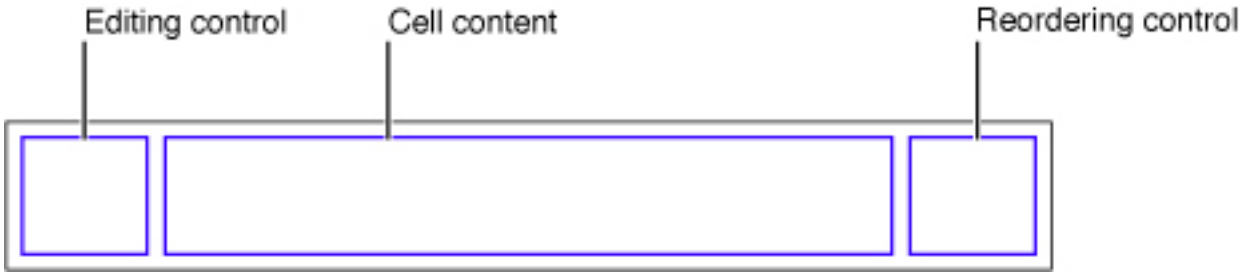
Zellenaufbau

ComTec

- Aufbau Standardzelle (normal):



- Aufbau Standardzelle (Bearbeitungsmodus):



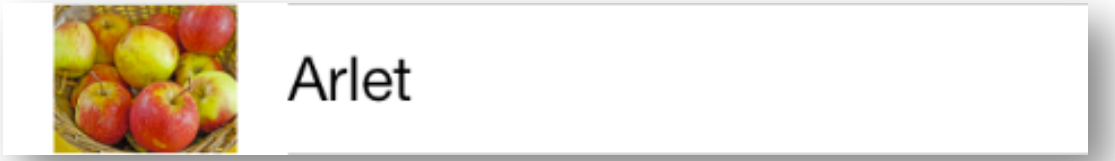
Quelle: [3]

Zellenstile

ComTec

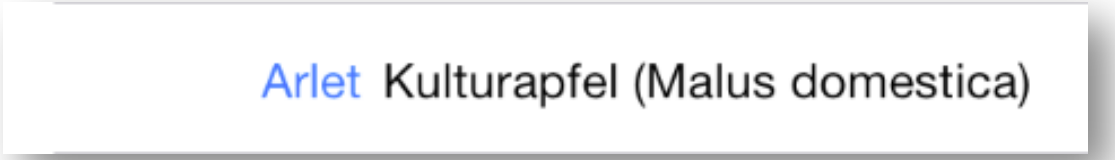
`UITableViewCellStyleDefault`:

`UITableViewCellStyleSubtitle`:



`UITableViewCellStyleValue1`:

`UITableViewCellStyleValue2`:



Quelle: [1]

Accessory-Stile

ComTec

- `UITableViewCellAccessoryDisclosureIndicator`:
  Weitere Tabelle folgt bei Berührung; ganze Zelle reagiert auf Touch

- `UITableViewCellAccessoryDetailButton`:
  Details (nicht zwingend Tabelle) folgen bei Berührung; nur Button reagiert auf Touch

- `UITableViewCellAccessoryCheckmark`:
  Einfach- und Mehrfachmarkierungen möglich; ganze Zelle reagiert auf Touch

Datenquelle (1)

ComTec

- Tabelle „fragt“ Datenquelle nach Tabelleninhalt
- Datenquelle muss `UITableViewDataSource`-Protokoll erfüllen
- Anzahl der Sektionen (optional):

```
- (NSInteger)numberOfSectionsInTableView:
  (UITableView *)tableView;
```

- Anzahl der Zeilen pro Sektion:

```
- (NSInteger)tableView:(UITableView *)tableView
  numberOfRowsInSection:(NSInteger)section;
```

Datenquelle (2)

ComTec

- Zellen (meist recycled) konfigurieren:

```
- (UITableViewCell *)tableView:(UITableView *)tableView
  cellForRowAtIndexPath:(NSIndexPath *)indexPath {
     static NSString *CellIdentifier = @"Cell";

      UITableViewCell *cell = [tableView
      dequeueReusableCellWithIdentifier:CellIdentifier];

      if (cell == nil) cell = [[[UITableViewCell alloc]
      initWithStyle:UITableViewCellStyleDefault
      reuseIdentifier:CellIdentifier] autorelease];

      // ...

      return cell;
}
```

Datenquelle (3)

ComTec

- Anzeige von Sektionsüberschriften (optional):

```
- (NSString *)tableView:(UITableView *)tableView
  titleForHeaderInSection:(NSInteger)section;
```

- Anzeige von Sektionsunterschriften (optional):

```
- (NSString *)tableView:(UITableView *)tableView
  titleForFooterInSection:(NSInteger)section;
```

- Anzeige Schnellwahlindex (optional):

```
- (NSArray *)sectionIndexTitlesForTableView:
  (UITableView *)tableView;
```

Reaktion auf Berührung (1)

ComTec

- Tabelle kontaktiert Delegate bei Berührung
- Methoden sind im `UITableViewDelegate`-Protokoll definiert
- Berührung der gesamten Zeile (optional):

```
- (void)tableView:(UITableView *)tableView
  didSelectRowAtIndexPath:(NSIndexPath *)indexPath;

- (void)tableView:(UITableView *)tableView
  didDeselectRowAtIndexPath:(NSIndexPath *)indexPath;
```

Reaktion auf Berührung (2)

ComTec

- Berührung der Accessory View (optional):

```
- (void)tableView:(UITableView *)tableView
  accessoryButtonTappedForRowWithIndexPath:
  (NSIndexPath *)indexPath;
```

ComTec

Anwendungsfälle

INTERNETANFRAGEN

Übersicht Internetanfragen

ComTec

- Datentypen
- Anfrage durchführen
- Caching-Möglichkeiten
- Antwort auswerten

Datentypen

ComTec

- `NSURL` zur Manipulation von URLs (nach RFC 1808, 1738 und 2732)
- `NSURLRequest`:
  - beinhaltet Anfragen unabhängig von Protokoll und URL-Schema
  - enthält URL, Cache-Regeln, Timeout, HTTP-Header-Felder
- `NSURLConnection` führt Anfragen durch
- `NSData` ist Wrapper für Byte-Puffer

Anfrage durchführen

ComTec

```
// URL erstellen
NSURL *url = [NSURL URLWithString:@"http://www.uni-kassel.de"];

// Anfrage erstellen
NSURLRequest *request = [[NSURLRequest alloc] initWithURL:url
                    cachePolicy:NSURLRequestUseProtocolCachePolicy
                    timeoutInterval:10.0];

// Antwortpuffer vorbereiten
NSMutableData *responseData = [[NSMutableData alloc] init];

// Verbindung erstellen; Anfrage asynchron starten
[[[NSURLConnection alloc] initWithRequest:request delegate:self]
  autorelease];
[request release];
```

Caching-Möglichkeiten

ComTec

- `NSURLRequestUseProtocolCachePolicy`:
  Standard; Richtlinie des Protokolls verwenden
- `NSURLRequestReloadIgnoringCacheData`:
  lokalen Cache nicht verwenden
- `NSURLRequestReloadIgnoringLocalAndRemoteCacheData`:
  lokalen und entfernten Cache nicht verwenden
- +3 Konstanten zur expliziten Cache-Nutzung

Antwort auswerten (1)

ComTec

```
// Callback-Funktionen definiert in:
// NSURLConnectionDataDelegate und NSURLConnectionDelegate


// Genügend Daten für Antwort erhalten -> Puffer zurücksetzen
- (void)connection:(NSURLConnection *)connection
  didReceiveResponse:(NSURLResponse *)response {
      responseData.length = 0;
}

// Daten erhalten -> an Puffer anhängen
- (void)connection:(NSURLConnection *)connection
  didReceiveData:(NSData *)data {
      [responseData appendData:data];
}
```

Antwort auswerten (2)

ComTec

```
// Download-Vorgang erfolgreich abgeschlossen
- (void)connectionDidFinishLoading:(NSURLConnection *)connection {
    // Daten verwenden
    // ...
    [responseData release];
}

// Fehler beim Download
- (void)connection:(NSURLConnection *)connection
  didFailWithError:(NSError *)error {
      // Fehler behandeln
      // ...
      [responseData release];
}
```

ComTec

Anwendungsfälle

SENSOREN

Übersicht Sensoren

ComTec

- Positionsbestimmung & Kompass
- Beschleunigung
- Gyroskop

Positionsbestimmung (1)

ComTec

- Bestimmung über WLAN ➟ Mobilfunk ➟ GPS
- Wahl der Mittel nicht möglich...
- ...Angabe der gewünschten Genauigkeit schon
- Framework liefert Positions- und Kompasswerte

Positionsbestimmung (2)

ComTec

```
// Framework einbinden
#import <CoreLocation/CoreLocation.h>

// LocationManager erzeugen und konfigurieren
CLLocationManager *locationManager = [[CLLocationManager alloc]
                                      init];
locationManager.delegate = self;
locationManager.desiredAccuracy = kCLLocationAccuracyBest;

// Positionsdaten erhalten
[locationManager startUpdatingLocation];

// Kompassdaten erhalten
[locationManager startUpdatingHeading];
```

Positionsbestimmung (3)

ComTec

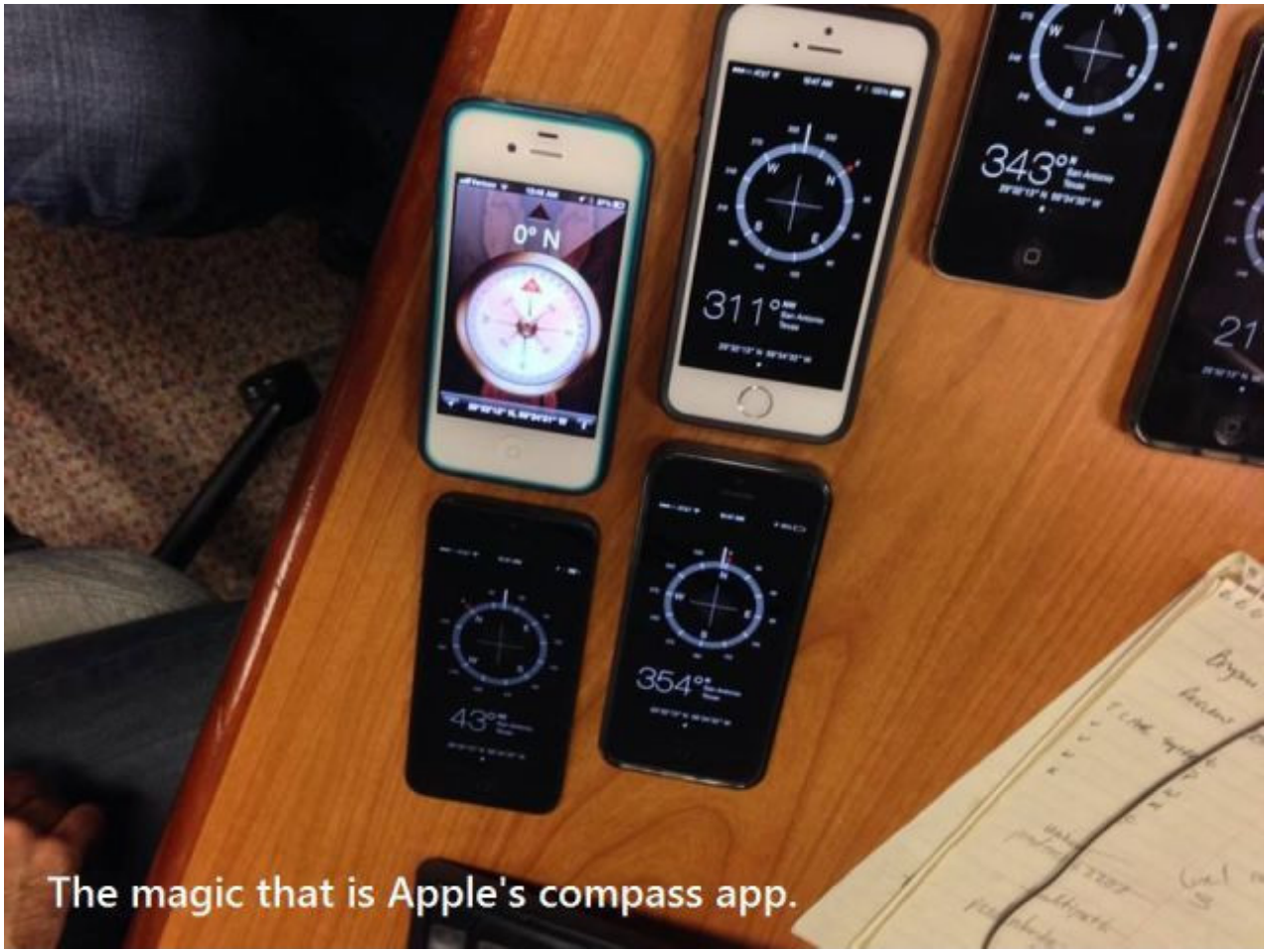
```
// Daten auswerten mittels CLLocationManagerDelegate-Protokoll

// Position gefunden
- (void)locationManager:(CLLocationManager *)manager
  didUpdateLocations:(NSArray *)locations {
     for (CLLocation *location in locations) {
        // Position verwenden ...
     }
}

// Kompasswert gefunden
- (void)locationManager:(CLLocationManager *)manager
  didUpdateHeading:(CLHeading *)newHeading {
     // Kompasswert verwenden ...
}
```

Positionsbestimmung (4)

ComTec

```
// Abfangen von Fehlern (bei Bedarf)
- (void)locationManager:(CLLocationManager *)manager
  didFailWithError:(NSError *)error;



// Positionsdaten nicht mehr erhalten
[locationManager stopUpdatingLocation];

// Kompassdaten nicht mehr erhalten
[locationManager stopUpdatingHeading];


// Zur Sicherheit: Überprüfen, ob Positionsbestimmung erlaubt
if ([CLLocationManager locationServicesEnabled]) // ...
```

Positionsbestimmung (5)



Quelle: [4]

Beschleunigung

ComTec

```
// Instanz holen und konfigurieren
UIAccelerometer *accelerometer = [UIAccelerometer
                                  sharedAccelerometer];
accelerometer.delegate = self;
accelerometer.updateInterval = 1.0;


// Daten auswerten (Protokoll: UIAccelerometerDelegate)
- (void)accelerometer:(UIAccelerometer *)accelerometer
  didAccelerate:(UIAcceleration *)acceleration {
     // acceleration.x, acceleration.y, acceleration.z;
}
```

Gyroskop (1)

ComTec

```
// Framework einbinden
#import <CoreMotion/CoreMotion.h>

// MotionManager erzeugen und konfigurieren
CMMotionManager *motionManager = [[CMMotionManager alloc] init];
motionManager.gyroUpdateInterval = 1.0;

// Rotation abfangen
NSOperationQueue *gyroQueue = [[[NSOperationQueue alloc] init]
                                autorelease];
[motionManager startGyroUpdatesToQueue:gyroQueue
                  withHandler:^(CMGyroData *gyroData, NSError *err) {
   CMRotationRate rotate = gyroData.rotationRate;
   // rotate.x, rotate.y, rotate.z
}];
```

Gyroskop (2)

ComTec

```
// Gyroskop deaktivieren
[motionManager stopGyroUpdates];
```

Einführung in Objective-C

QUELLEN

Quellen (2)

ComTec

- [1] Wikipedia Foundation Inc.: „Liste von Apfelsorten“ (http://de.wikipedia.org/wiki/Liste_von_Apfelsorten, 18.11.2013)
- [2] Apple, Inc.: „Table View Styles and Accessory Views“ (https://developer.apple.com/library/ios/documentation/userexperience/conceptual/tableview_iphone/tableviewstyles/tableviewcharacteristics.html , 18.09.2013)
- [3] Apple, Inc.: „A Closer Look at Table View Cells“ (https://developer.apple.com/library/ios/documentation/userexperience/conceptual/TableView_iPhone/TableViewCells/TableViewCells.html, 18.09.2013)
- [4] isnichwahr.de: „Picdump #04112013“ (http://www.isnichwahr.de/r97808391-isnichwahr-de-picdump-04112013.html, 21.11.2013)

Stand: 26. November 2013