定位与地图

定位CoreLocation

CLLocationManager

- startUpdatingLocation
  - `- (void)locationManager: didUpdateLocations:;`
    - CLLocation
      - `CLLocationCoordinate2D coordinate;`经纬度
        - `CLLocationDegrees latitude;`
        - `CLLocationDegrees longitude;`
      - `CLLocationDistance altitude;`海拔
      - `CLLocationDirection course;`方向
      - `CLLocationSpeed speed:`行走速度
      - `-(CLLocationDistance)distanceFromLocation: (const CLLocation *)location`方法可以计算2个位置之间的距离
- stopUpdatingLocation
- `CLLocationDistance distanceFilter;` 每隔多少米定位一次
- `CLLocationAccuracy desiredAccuracy;` 定位精确度（越精确就越耗电）
- `+ (BOOL)locationServicesEnabled;` 判断定位是否可用

CLGeocoder

- 地理编码方法 `- (void)geocodeAddressString:(NSString *)addressString completionHandler: (CLGeocodeCompletionHandler)completionHandler;`
- 反地理编码方法 `- (void)reverseGeocodeLocation:(CLLocation *)location completionHandler: (CLGeocodeCompletionHandler)completionHandler;`

→ CLPlacemark

- `CLLocation *location;` 地理位置
- `CLRegion *region;`区域
- `NSDictionary *addressDictionary;` 详细的地址信息
- `NSString *name;`地址名称
- `NSString *locality;`城市

地图MapKit

MKMapView

- userTrackingMode
  - MKUserTrackingModeNone
  - MKUserTrackingModeFollow
  - MKUserTrackingModeFollowWithHeading
- mapViewType
  - MKMapTypeStandard
  - MKMapTypeSatellite
  - MKMapTypeHybrid
- 代理方法
  - `- (void)mapView:(MKMapView *)mapView didUpdateUserLocation:(MKUserLocation *)userLocation;` 当监听到用户位置的时候调用
    - MKUserLocation
      - `NSString *title;`在大头针上的标题
      - `NSString *subtitle;`在大头针上的子标题
      - `CLLocation *location;` 地理位置信息
  - `- (void)mapView:(MKMapView *)mapView regionDidChangeAnimated:(BOOL)animated;` 地图的显示区域已经发生改变的时候调用
- 设置MapView的显示区域
  - `- (void)setCenterCoordinate: (CLLocationCoordinate2D)coordinate animated: (BOOL)animated;`设置地图的中心点位置
    - CLLocationCoordinate2D center
  - `- (void)setRegion: (MKCoordinateRegion)region animated: (BOOL)animated;`设置地图的显示区域
    - MKCoordinateRegion
      - CLLocationCoordinate2D center
      - MKCoordinateSpan span
        - `CLLocationDegrees latitudeDelta;`
        - `CLLocationDegrees longitudeDelta;`
- 大头针
  - 基本操作
    - 添加一个大头针 `- (void)addAnnotation:(id<MKAnnotation>)annotation;`
    - 添加多个大头针 `- (void)addAnnotations:(NSArray *)annotations;`
    - 移除一个大头针 `- (void)removeAnnotation:(id<MKAnnotation>)annotation;`
    - 移除多个大头针 `- (void)removeAnnotations:(NSArray *)annotations;`
    - `id <MKAnnotation>)annotation`大头针模型对象：用来封装大头针的数据，比如大头针的位置、标题、子标题等数据

```
新建一个大头针模型类
#import <MapKit/MapKit.h>
@interface MJTuangouAnnotation : NSObject <MKAnnotation>
/** 坐标位置 */
@property (nonatomic, assign) CLLocationCoordinate2D coordinate;
/** 标题 */
@property (nonatomic, copy) NSString *title;
/** 子标题 */
@property (nonatomic, copy) NSString *subtitle;
@end
```

```
MJTuangouAnnotation *anno = [[MJTuangouAnnotation alloc] init];
anno.title = @"传智播客iOS学院";
anno.subtitle = @"全部课程15折，会员20折，老学员30折";
anno.coordinate = CLLocationCoordinate2DMake(40, 116);
[self.mapView addAnnotation:anno];
```

  - 自定义大头针
    - 实现下面的代理方法，返回大头针控件 `- (MKAnnotationView *)mapView:(MKMapView *)mapView viewForAnnotation:(id <MKAnnotation>)annotation;`
    - `id <MKAnnotation> annotation;` 大头针模型
    - MKAnnotationView
      - `UIImage *image;`显示的图片
      - `BOOL canShowCallout;` 是否显示标注
      - `CGPoint calloutOffset;` 标注的偏移量
      - `UIView *rightCalloutAccessoryView;` 标注右边显示什么控件
      - `UIView *leftCalloutAccessoryView;` 标注左边显示什么控件
    - MKPinAnnotationView
      - MKPinAnnotationView是MKAnnotationView的子类
      - `MKPinAnnotationColor pinColor;`大头针颜色
      - `BOOL animatesDrop;`大头针第一次显示时是否从天而降

导航

MKDirections

- MKDirectionsRequest
  - 起始点source
  - 目的地destination
  - transportType交通工具
    - MKDirectionsTransportTypeAutomobile开车
    - MKDirectionsTransportTypeWalking
    - MKDirectionsTransportTypeAny
  - MKMapItem
    - `[MKMapItem mapItemForCurrentLocation]`
    - `[[MKMapItem alloc] initWithPlacemark:mkPM];` MKPlacemark `- (id)initWithPlacemark:(CLPlacemark *)placemark;`
- `- (void)calculateDirectionsWithCompletionHandler: (MKDirectionsHandler)completionHandler;` MKDirectionsResponse → routes所有的线路 → polyline获取多线段 → `[self.mapKit addOverlay:polyline];`

MKOverlayRenderer

`- (MKOverlayRenderer *)mapView:(MKMapView *)mapView rendererForOverlay (id<MKOverlay>)overlay` → MKPolylineRenderer

- strokeColor
- fillColor
- lineWidth