PAM
Exam – 20 March 2015
Android
F1 – F2 – F5

No documents allowed. The marking-scheme is only an indication.
Please write your answers on a separate sheet. Do not use the same sheet as for the iOS part.

1 Android questions (6 marks)

1. What is ART, what is Dalvik? Why such components are needed in Android?
2. What is an `Activity`? What is a `Fragment`? Explain their interactions.
3. The R class is a fundamental class in Android. Why? How is it built?
4. What are `Intents`? How did we use them during the Forum App?
5. Network calls in Android need to use special classes. Which ones can be used? Why do we need such classes?

2 Android exercise (2 marks)

The following code sample is used in application displaying `Articles` in a `ListView`. Modify the sample to make it load the articles from a web-service and then displaying them in the `ListView`.

Make sure you follow these steps:

- make a network call to this address: `http://forum.openium.fr/news?limit=100`
- the web-service return the data in JSON. Parse the content to build an `Article List`.
- display the content in the `ListView` using an `ArrayAdapter`

Assume that:

- the `ListView` is accessible via an attribute named `mListView`
- the correct configuration is set in the `AndroidManifest.xml`
- the parsing library is included in the project

First name	Surname

```
private void loadAndDisplayArticles(){
      new AsyncTask<Void, Void, List<Article>>() {

          @Override
          protected List<Article> doInBackground(Void... params) {









              return null;
          }

          protected void onPostExecute( List<Article> result) {









          }

      }.execute();
  }
```

Développement d'application mobile - F1 - F2 - F5

Sans documents. Chaque réponse doit être argumentée. Le barème est donné à titre indicatif. Un soin particulier devra être apporté à l'écriture et à l'ortaugrafe. Chaque partie (iOS / android) devra être réalisée sur des copies différentes.

Questions iOS (12pt)

1. Nommer le composant permettant d'afficher une liste. Expliquez le mécanisme qui permet à ce composant d'être efficace sur un appareil ayant très peu de ressources (iPhone "edge"). (2pt)

2. Améliorer le code suivant : (1pt)

```
[self.navigationItem setRightBarButtonItem:
[[UIBarButtonItem alloc]
initWithBarButtonSystemItem:UIBarButtonSystemItemAdd
target:self action:@selector(insertNewObject:)]];
```

3. Un problème se trouve dans les tests unitaires ci-joints. Lequel ? (1pt)

4. D'après les tests unitaires et la documentation ci-joints, écrivez le code de la classe testée. (8pt)

```
//
//  ChronoTests.m
//  ChronoTests
//
//  Created by Richard Bergoin on 10/02/2015.
//  Copyright (c) 2015 Openium. All rights reserved.
//

/** Extrait de NSDate.h
@interface NSDate (NSExtendedDate)

- (NSTimeInterval)timeIntervalSinceDate:(NSDate *)anotherDate;
@property (readonly) NSTimeInterval timeIntervalSinceNow;
@property (readonly) NSTimeInterval timeIntervalSince1970;

- (id)addTimeInterval:(NSTimeInterval)seconds NS_DEPRECATED(10_0, 10_6, 2_0,
4_0);
- (instancetype)dateByAddingTimeInterval:(NSTimeInterval)ti NS_AVAILABLE(10_6,
2_0);

- (NSDate *)earlierDate:(NSDate *)anotherDate;
- (NSDate *)laterDate:(NSDate *)anotherDate;
- (NSComparisonResult)compare:(NSDate *)other;
- (BOOL)isEqualToDate:(NSDate *)otherDate;

@property (readonly, copy) NSString *description;
- (NSString *)descriptionWithLocale:(id)locale;

+ (NSTimeInterval)timeIntervalSinceReferenceDate;

@end
*/

#import <Cocoa/Cocoa.h>
#import <XCTest/XCTest.h>

#import "Chrono.h"

@interface ChronoTests : XCTestCase {
    Chrono *chrono;
}
@end

@implementation ChronoTests

- (void)setUp {
    [super setUp];
    chrono = [[Chrono alloc] init];
}

- (void)tearDown {
    [super tearDown];
}

- (void)testTotalDuration_shouldReturnOneSecond {
    // Given
    [chrono start];
    sleep(1);
    [chrono stop];
```

```
    // When
    NSNumber *d = [chrono totalDuration];
    NSTimeInterval duration = [d doubleValue];

    // Expect
    XCTAssertEqualWithAccuracy(duration, 1.0f, 0.01f);
}

- (void)testLapDurations_shouldReturnNoDurations {
    // Given

    // When
    NSArray *lapDurations = [chrono lapDurations];

    // Expect
    XCTAssertEqual([lapDurations count], 0);
}

- (void)testLapDurations_shouldReturnTwoTimesOfOneSecond {
    // Given
    [chrono start];
    sleep(1);
    [chrono lap];
    sleep(1);
    [chrono lap];
    [chrono stop];

    // When
    NSArray *lapDurations = [chrono lapDurations];
    NSTimeInterval firstLapDuration = [[lapDurations firstObject] doubleValue];
    NSTimeInterval secondLapDuration = [[lapDurations lastObject] doubleValue];

    // Expect
    XCTAssertEqual([lapDurations count], 2);
    XCTAssertEqualWithAccuracy(firstLapDuration, 1.0f, 0.01f);
    XCTAssertEqualWithAccuracy(secondLapDuration, 1.0f, 0.01f);
}

- (void)testPause_shouldReturnOneSecond {
    // Given
    [chrono start];
    sleep(1);
    [chrono pause];
    sleep(1);
    [chrono resume];
    sleep(1);
    [chrono stop];

    // When
    NSNumber *d = [chrono totalDuration];
    NSTimeInterval duration = [d doubleValue];

    // Expect
    XCTAssertEqualWithAccuracy(duration, 2.0f, 0.01f);
}

@end
```