

## TEST PRACTIC 2 – ELEMENTE DE INFORMATICĂ MOBILĂ

Exemplu, 19 mai 2017

Numele și Prenumele Studentului \_\_\_\_\_  
Cont GitHub \_\_\_\_\_ Grupa \_\_\_\_\_

### Indicații

- Timpul de lucru efectiv este de 90 de minute.
- Punctajul care poate fi obținut pentru acest test practic este de 100 de puncte.
- Acest test practic un subiect practic (o aplicație Android) și un subiect teoretic, fiecare cu mai multe subpuncte. Este recomandat ca în cazul problemei, cerințele să fie tratate în ordinea în care apar.
- Pentru fiecare subiect poate fi obținut punctaj parțial (conform fracțiunilor), însă exclusiv în condițiile unei implementări funcționale.
- După fiecare 30 de minute, cadrul didactic va evalua măsura în care au fost rezolvate exercițiile până la momentul respectiv, punctându-le corespunzător.
- La expirarea timpului de lucru, codul sursă trebuie încărcat pe contul Github. Nu se iau în considerare alte modificări locale. Asigurați-vă că după rezolvarea fiecărui exercițiu, acesta este încărcat pe depozitul la distanță!
- Pentru acest test practic puteți consulta orice fel de resursă, dacă aceasta nu implică colaborarea cu alte persoane. Folosirea oricărui mijloc de comunicare se consideră copiere și se sancționează cu anularea punctajului aferent testului practic!

### SUBIECT PRACTIC

1. [5%] a) În contul Github personal, să se creeze un depozit denumit *PracticalTest02*.

Inițial, acesta trebuie să conțină:

- un fișier `README.md`, în care veți preciza următoarele informații: numele și prenumele, grupa din care faceți parte;
- un fișier `.gitignore` prin intermediul căruia resursele generate ale aplicației Android (clase, fișiere `.ap_`, `.apk`, `.dex`) nu vor fi consemnate în sistemul de versionare a codului sursă;
- un fișier `LICENSE` care să descrie condițiile de utilizare a aplicației Android (la alegere).

b) Să se descarce pe discul local conținutul depozitului (la distanță) *PracticalTest02*.

c) În mediul integrat de dezvoltare Android Studio, să se creeze un proiect corespunzător unei aplicații Android, având următoarea specificație:

- denumirea aplicației Android și a proiectului este *PracticalTest02*;
- denumirea pachetului corespunzător aplicației Android este `ro.pub.cs.systems.eim.practicaltest02`;
- nivelul minim de API este cel corespunzător versiunii Jelly Bean (4.1);
- denumirea activității principale a aplicației Android este `PracticalTest02MainActivity`;
- denumirea fișierului ce descrie interfața grafică a activității principale a aplicației Android este `activity_practical_test02_main.xml`.

Se cere să se implementeze o aplicație client-server, ambele fiind integrate în cadrul aceluiași program Android.

Funcționalitatea oferită vizează furnizarea de informații meteorologice oficiale, așa cum sunt preluate de la serviciul Internet Weather Underground, disponibil la adresa <http://www.wunderground.com/cgi-bin/findweather/getForecast>. Acesta primește parametrul `query` (transmis fie printr-o metodă de tip `GET`, fie printr-o metodă de tip `POST`) care conține orașul despre care se doresc să se obțină informațiile meteorologice.

Serverul va transmite aceste informații meteorologice pe un anumit port, precizat de utilizator. De asemenea, el va gestiona un obiect care va conține datele preluate anterior, referitoare la orașele pentru care au fost realizate solicitări primite de la alți clienți. Datele care pot fi oferite sunt temperatura, viteza vântului, starea generală, presiunea și umiditatea. Solicitarea primită va fi tratată astfel: dacă informația este disponibilă local, ea va fi preluată din obiectul gestionat, altfel se va realiza o conexiune la serviciul Internet la distanță, preluându-se informațiile meteorologice disponibile.

Clientul va trimite o cerere la server folosind parametrii de conectare adresa și port. Ulterior va preciza, pe câte o linie, orașul și informația pe care o solicită. În situația în care dorește să primească toate datele disponibile, va indica parametrul `all`. Mesajul primit poate fi distribuit pe mai multe linii, conținând răspunsul, acesta fiind afișat în cadrul interfeței grafice.

2. [15%] Să se implementeze o interfață grafică minimală pentru aplicația client-server.

3. Să se implementeze funcționalitatea specifică serverului:

- a) [10%] gestiunea obiectului local cu informațiile meteorologice solicitate anterior;
- b) [20%] preluarea informațiilor meteorologice prin accesarea serviciului Internet;
- c) [5%] prelucrarea unei cereri provenite de la client;
- d) [5%] transmiterea unui răspuns către clienți.

4. Să se implementeze funcționalitatea specifică clientului. Utilizatorul va specifica un oraș precum și informația meteorologică solicitată (respectiv parametrul `all` în situația în care dorește să primească toate datele disponibile), aceasta urmând a fi vizualizată în cadrul unui control din cadrul interfeței grafice.

- a) [15%] Să se acceseze funcționalitatea oferită de server prin intermediul unui client, implementat în cadrul aceleiași aplicații Android.
- b) [5%] Să se interogheze funcționalitatea oferită de server prin intermediul unui client care rulează pe mașina fizică (`nc` sau `telnet`). Monitorizați timpul de răspuns în situația în care sunt realizate mai multe cereri (aproximativ) în același timp.

5. [5%] Să se încarce rezolvările în depozitul *PracticalTest02* de pe contul Github personal folosind un mesaj sugestiv. Dacă este necesar, configurați global numele de utilizator și adresa de poștă electronică.

**SUBIECT TEORETIC**

6. [7.5%] Într-un apel telefonic bazat pe protocolul SIP, un user agent transmite unui registration server 2 mesaje de tip REGISTER. Primul răspuns este *Status: 401 Unauthorized* iar al doilea răspuns *Status: 200 OK*. Care este motivul pentru care apar retransmisii? Care este diferența dintre mesajele de tip REGISTER transmise?

.....  
.....  
.....

7. [7.5%] Care sunt controalele grafice care pot fi utilizate în Android pentru a afișa o hartă Google? Care sunt asemănările și diferențele între aceste abordări?

.....  
.....  
.....