

**MINISTERUL EDUCAȚIEI AL REPUBLICII MOLDOVA
UNIVERSITATEA DE STAT „ALECU RUSSO” DIN BĂLȚI
FACULTATEA DE ȘTIINȚE REALE, ECONOMICE ȘI ALE MEDIULUI
CATEDRA DE MATEMATICĂ ȘI INFORMATICĂ**

CURRICULUM

pentru unitatea de curs

„PROGRAMAREA APLICAȚIILOR PE DISPOZITIVE MOBILE”

pentru specialitatea Informatică

Ciclul I, studii superioare de licență, învățământ cu frecvență

Titularul unității de curs:
lector universitar Adela Gorea

BALȚI, 2016

Curriculumul a fost discutat la ședința Catedrei de matematică și informatică

Procesul verbal nr. 1 din 29.08.2016

Șeful catedrei dr. conf. univ. E. Plohotniuc _____

Curriculumul a fost aprobat la ședința Consiliului Facultății de Științe Reale, Economice și ale Mediului

Procesul verbal nr. 5 din 20.10.2016

Decanul facultății, dr. hab., prof. univ. P. Topala _____

Informații de identificare a unității de curs

Facultatea: Științe Reale, Economice și ale Mediului

Catedra: matematică și informatică

Domeniul general de studiu: 14 Științe ale Educației

Domeniul de formare profesională la ciclul I: Educație și formarea profesorilor ciclul I, licență

Denumirea specialității: 141.02. Informatica

Denumirea unității de curs: Programarea aplicațiilor pe dispozitive mobile

Administrarea unității de curs:

Codul unității de curs	Credite ECTS	Total ore	Repartizarea orelor				Forma de evaluare	Limba de predare
			Prelegeri	Seminare	Laboratoare	Lucrul individual		
S. 06.A.143	4	120	30	–	30	60	Examen	Rom

Statutul: Unitate de curs la libera alegere

Localizarea sălilor: curs – aula 224, laboratoare – aula 158, 141a

Informații referitoare la cadrele didactice

Titularul cursului – *Adela Gorea*, lector universitar, absolventa Universității de Stat „A. Russo” din Bălți, specialitatea „Informatică”. A efectuat studiile de master la specializarea „Tehnologii informaționale și comunicaționale în învățământ”.

E-mail: adelaosadta@gmail.com

Orele de consultații – joi: 14.00 -15.30. Consultațiile se oferă atât în regim „față-în-față”, cât și prin utilizarea poștei electronice, Skype. Numele în Skype – [adela_gorea](#).

Integrarea unității de curs în programul de studii

Cursul „Programarea aplicațiilor pe dispozitive mobile” este un curs de specializare care permite dezvoltarea competențelor de programare. Scopul de bază este de a învăța studenții să dezvolte aplicații pentru SO Android cu ajutorul limbajului de programare Java. Pe parcursul studierii cursului, studenții vor dobândi noi cunoștințe atât despre partea de bază a unei aplicații Android, cât și despre mediul de dezvoltare, realizând o aplicație cu diverse componente: pornind de la controale simple care alcătuiesc interfața grafică de la TextView, Button până la ListView și GridView. Studenții vor realiza interfețe grafice cu utilizatorul, vor trata evenimente, vor defini și utiliza meniuri și ferestre de dialog, vor analiza modalitățile de stocare persistentă a datelor, inclusiv în baze de date SQLite, vor analiza modul de accesare la rețea și modul de prelucrare a fișierelor XML și JSON.

Cursul este destinat studenților de la specialitățile „Informatica (profil pedagogic)” studii superioare de licență a Facultății de Științe Reale, Economice și ale Mediului. Este o disciplină la libera alegere pentru viitorii specialiști în informatică.

Studierea disciplinei „Programarea aplicațiilor pe dispozitive mobile” se sprijină pe cunoștințele, capacitățile și competențele dezvoltate în cadrul disciplinelor „Bazele programării”, „Programarea orientată pe obiect”, „Programarea Java”, „Programare Web” și „Gestiunea informației”. Scopurile și conținutul cursului sunt corelate cu scopurile și conținutul cursurilor menționate mai sus.

Prin conținutul său și activitățile de învățare a studenților, cursul „Programarea aplicațiilor pentru diverse platforme” contribuie la dezvoltarea mai multor competențe generice, necesare profesorului de informatică:

- capacitatea de analiză și sinteză;
- deprinderi de gestiune a informației (extragerea și analiza informației din diverse surse);
- capacitatea de a lucra în echipă;
- atașamentul la valorile etice;
- capacitatea de a aplica cunoștințele în practică;
- capacitatea de a genera idei noi;
- capacitatea de a lucra independent.

Competențe prealabile:

- Utilizarea eficientă a resurselor sistemelor de calcul, de operare și ale Internetului.
- Dezvoltarea de componente pentru produse software, folosind structuri de date, algoritmi, tehnici și limbaje de programare evolute.

Competențele formate/dezvoltate în cadrul unității de curs:

În cadrul studierii unității de curs la studenți vor fi dezvoltate următoarele competențe CP – competențe profesionale:

CP1. Operarea cu fundamentele științifice ale matematicii, informaticii și ale științelor educației și utilizarea acestor noțiuni în comunicarea profesională.

CP2. Elaborarea modelelor pentru descrierea fenomenelor și proceselor reale.

CP3. Proiectarea, elaborarea și analiza algoritmilor pentru rezolvarea problemelor.

CP4. Programarea, dezvoltarea și mentenanța aplicațiilor informatice în limbaje de nivel înalt.

CT – competențe transversale:

CT2. Desfășurarea eficientă și eficace a activităților organizate în echipă.

CT3. Identificarea oportunităților de formare continuă și valorificarea eficientă a resurselor și tehnicilor de învățare pentru propria dezvoltare.

Finalitățile unității de curs

La finalizarea studierii cursului studentul va fi capabil:

- să proiecteze o aplicație mobilă;

- să cunoască destinația claselor esențiale;
- să posede deprinderi de lucru cu clase, metode abstracte și interfețe;
- să trateze evenimente în cadrul mobile;
- să determine funcționalitatea aplicației cu baze de date;
- să deprindă logica dezvoltării și testării funcționalității programelor.

Structura unității de curs

Conținuturi

Nr. d/o	Subiectele predate	Nr. de ore
1.	Platforme mobile. Concepții de baza Android. Arhitectura sistemului de operare Android	2
2.	Mediu de programare a aplicațiilor	2
3.	Clase uzuale Android. Instrumente alternative pentru dezvoltarea aplicațiilor Android	2
4.	Resurse, activități.	2
5.	Mesaje. Clasa Intent. Mesaje de tip PendingIntent	2
6.	Interfața grafică și tratarea evenimentelor	2
7.	Meniuri și bara de acțiune. Meniuri de opțiuni. Meniuri contextuale. Meniuri contextuale. Meniuri în bara de acțiune și navigarea standard.	2
8.	Evaluare curentă.	2
9.	Fragmente și controale din bibliotecile suport. Clasa ViewPager.	2
10.	Stocarea persistentă a datelor. Fișiere XML compilate. Fișiere ca fluxuri de date.	2
11.	Accesul la rețea. Controalele WebView și WebViewFragment. Comunicația prin protocolul HTTP.	2
12.	Prelucrarea fișierelor XML și JSON. Prelucrarea fișierelor XML cu biblioteca SAX. Clase pentru prelucrarea fișierelor JSON	2
13.	Furnizori de conținut și receptori de mesaje.	2
14.	Calitatea aplicațiilor Android. Securitatea aplicațiilor Android	2
15.	Evaluare curentă.	2
Total		30

Temetica lecțiilor de laborator

Nr. d/o	Tematica	Nr. de ore
1.	Aplicații Android. Android development Tools. Android SDK.	2
2.	Medii de dezvoltare. IDE Eclipse. IDE Android Studio.	2
3.	Definirea și referirea resurselor. Șiruri de caractere. Liste de valori. Resurse grafice.	2
4.	Clasele View și ViewGroup. Butoane. Bara de progres. Componente vizuale de tip listă.	4

5.	Meniuri. Bara de acțiune Action Bar.	4
6.	Grafica bibimensională și tridimensională.	2
7.	Ferestre de dialog și de informare. Clasa DialogFragment. Ferestre de tip PopupWindow. Mesaje informative.	2
8.	Baze de date. Utilizarea clasei SQLiteOpenHelper. Utilizarea comenzilor SQL	2
9.	Clasa Thread și interfața Runnable. Clasa Handler. Clasa Timer. Clasa LoaderManager.	4
10.	Multimedia și animații	2
11.	Servicii și senzori. Implementarea serviciilor Android. Clasa IntentService	2
12.	Determinarea poziției geografice și utilizarea Google Maps.	2
13.	Managementul informațiilor personale și telefonie. Gestiunea contactelor. Mesagerie și telefonie. Evenimente. Notificări.	2
Total		30

Consemnele pentru sarcinile individuale și de grup

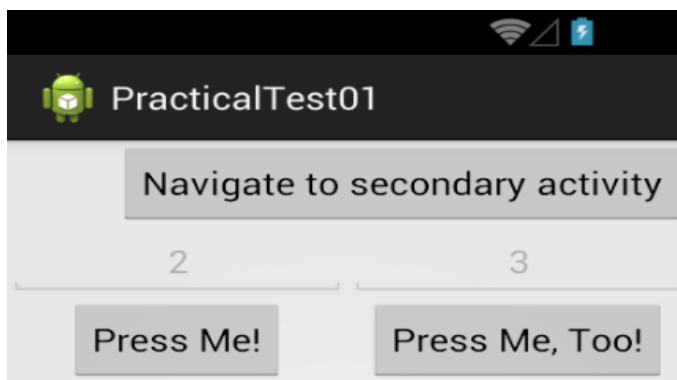
Sarcina nr. 1 pentru lucrul independent

A.1. În contul personal, să se creeze un depozit denumit *PracticalTest01*. 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 cadrul sistemului de versionare a codului sursă; un fișier *LICENSE* care să descrie condițiile de utilizare a aplicației Android (la alegere).

A.2. Să se descarce pe discul local conținutul depozitului (la distanță) *PracticalTest01*.

A.3. În mediul integrat de dezvoltare preferat (Eclipse / Android Studio), să se creeze un proiect corespunzător unei aplicații Android, având următoarea specificație: denumirea aplicației Android și a proiectului este *PracticalTest01*; denumirea pachetului corespunzător aplicației Android este *Nume/Prenume.practicaltest01*; nivelul minim de API este cel corespunzător versiunii Jelly Bean (4.1); denumirea activității principale a aplicației Android este *PracticalTest01MainActivity*; denumirea fișierului ce descrie interfața grafică a activității principale a aplicației Android este *activity_practical_test01_main.xml*.

A.4. Să se creeze o interfață grafică care conține două câmpuri text, needitabile, dispuse pe o linie, unul lângă celălalt ocupând jumătate din suprafața containerului din care fac parte, având dedesubt câte un buton centrat relativ la ele. În cadrul câmpurilor text, se va afișa o cifră reprezentând numărul de apăsări al butonului (inițial, valoarea va fi 0).



B.1. Să se asocieze butoanelor un ascultător, astfel încât pentru fiecare eveniment de tip apăsare a butonului, valoarea din câmpul text corespunzător să fie incrementată.

B.2. Forțați sistemul de operare Android să nu salveze / restaureze în mod automat starea câmpurilor text în momentul în care aceasta este distrusă din cauza asigurării necesarului de resurse. Implementați un mecanism pentru salvarea / restaurarea valorii din câmpurile text, dacă sistemul de operare Android distruge activitatea spre a asigura necesarul de resurse. Simulați comportamentul de mai sus.

Ce se întâmplă în situația în care utilizatorul apasă butonul *Back*? Cum ar fi putut fi rezolvată problema de mai sus în aceste condiții?

C.1. Să se implementeze o nouă activitate (`PracticalTest01SecondaryActivity`, având interfața grafică în fișierul `activity_practical_test01_secondary.xml`) în contextul aceleiași aplicații Android, care să poată fi invocată doar prin intermediul unei intenții. Aceasta va conține un câmp text, care va afișa de câte ori au fost accesate cele două butoane (din activitatea principală) în total. Totodată, aceasta va avea și două butoane, *Ok* și *Cancel*, prin care se va reveni la activitatea din care a fost invocată, însă cu rezultate diferite.

C.2. În activitatea principală, să se creeze un buton prin care se poate invoca cealaltă activitate din aplicația Android, afișând la revenirea din ea (într-un control grafic de tip `Toast`) rezultatul care a fost primit.

D.1. Să se implementeze un serviciu de tip *started* denumit `PracticalTest01Service` care primește prin intenția cu care e invocat numerele din câmpurile text și propagă la nivelul sistemului de operare Android un mesaj cu difuzare, la fiecare 10 secunde ce conține data și ora la care a fost transmis, media aritmetică și media geometrică numerelor. Definiți trei acțiuni diferite, care vor fi asociate aleator intenției folosite pentru propagarea mesajului cu difuzare.

Să se pornească serviciul atunci când suma valorilor din cele două câmpuri text depășește un prag, definit la nivelul codului sursă sub formă de constantă. Serviciul este oprit (nu mai transmite mesaje cu difuzare) atunci când aplicația Android este distrusă.

D.2. Să se implementeze un ascultător pentru mesajele de difuzare. Procesarea unui mesaj cu difuzare implică jurnalizarea sa în consolă, folosind o anumită etichetă.

D.3. Să se încarce rezolvările în depozitul `PracticalTest01` de pe contul personal, folosind un mesaj sugestiv.

Sarcina nr. 2 pentru lucrul independent

Fiecare student va realiza un proiect. Proiectul presupune elaborarea unei aplicații pentru platforma Android (tematica la libera alegere) care rezolvă o problemă din viața reală. Aplicația trebuie să aibă o interfața grafică din două sau trei Activity și comunicarea între componente. Aplicația va conține și o descriere a problemei soluționate și a modului în care a fost rezolvată. Codul aplicației va avea comentarii explicative. Descrierea aplicației va fi prezentată într-un raport editat într-un document Word pe 3-4

pagini format A4, font #12, 1.5 intervale. Activitatea va fi evaluată atât de către colegi cât și de către titularul disciplinei perechea a 14-a.

Criterii de evaluare:

- Corectitudinea rezolvării problemei prin elaborarea aplicației pentru platforma Android;
- Relevanța și valoarea comentariilor;
- Exactitatea (logică, ortografică) raportului prezentat.

Termenul limită (deadline) de prezentare a sarcinii – perechea a 14-a (lucrare de laborator).

Evaluare

Cunoștințele, capacitățile și competențele studenților vor fi evaluate:

- În cadrul orelor practice/lucrărilor de laborator (conform calendarului disciplinei);
- Prin realizarea a 2 sarcini de lucru independent (conform calendarului disciplinei);
- La examenul final (conform orarului întocmit de decanat).

Nota finală la disciplina „Programarea aplicațiilor pe dispozitive mobile” se calculează conform formulei: $N_f = 0,6 \times n_c + 0,4 \times n_e$,

unde N_f – nota finală; n_c – media curentă, n_e – nota de la examen.

$$n_c = (m_{lab} + m_{teorie} + s_{i_1} + s_{i_2}) / 4$$

unde n_c – media curentă, m_{lab} - media notelor pentru lucrările de laborator, m_{teorie} – media notelor pentru evaluările curente, s_{i_1} și s_{i_2} – note pentru sarcinile de lucru independent.

Examenul final se susține în scris.

Principiile de lucru în cadrul disciplinei:

1. O parte din sarcinile de învățare vor fi propuse pentru realizare în grupe mici prin cooperare. Deși activitatea de învățare va fi una colectivă, notele pentru realizarea sarcinilor vor fi individuale. Prezentarea sarcinilor realizate va fi însoțită de o evaluare reciprocă a membrilor subgrupului pentru a identifica aportul fiecărui membru în rezultatul final.
2. Calendarul cursului (termenii-limită de prezentare a sarcinilor propuse spre rezolvare, momentele de evaluare etc.) este corelat cu calendarele la alte discipline din semestru. De aceea prezentarea sarcinilor după termenul-limită indicat în calendar nu este salutăată, iar studenții care amână frecvent prezentarea sarcinilor își formează o imagine nefavorabilă.
3. Nu este salutăată întârzierea la ore.
4. Este salutăată poziția activă a studentului, care studiază din propria inițiativă noi conținuturi, propune soluții (aplicații, instrumente Web), formulează întrebări în cadrul prelegerilor și a orelor practice.
5. În cadrul disciplinei o atenție sporită va fi oferită respectării principiilor *etice*. Prezentarea unor soluții a sarcinilor, preluate de la colegi sau din alte surse, preluarea informațiilor din diverse surse, fără a face trimitere la sursă, va fi considerată *plagiat* și va fi sancționată prin note de „1”.

Resursele informaționale la disciplină

A. Literatura de bază

1. Pocatilu P., Ivan I., Visoiu A., Alecu F., Zamfiroiu A., Iancu B. Programarea aplicatiilor Android. Editura ASE, 2015, ISBN:978-606-505-856-9, 709 p.
2. Колисниченко Д. Программирование для Android. Издательство БХВ-Петербург, 2014, ISBN 978-5-9775-0770-7, 261с.
3. Пол Дейтел, Харви Дейтел, Android для программистов: создаем приложения. Издательство Питер, 2013, ISBN:978-5-459-01646-8, 557с.

B. Literatura suplimentară

4. Медникс З., Дорнин Л. Программирование под Android. Издательство Питер, 2013, 456 с.
5. Рето Майер, Программирование приложений для планшетных компьютеров и смартфонов, Издательство Эксмо, 2013, 590 с.
6. Голощапов Алексей Google Android: системные компоненты и сетевые коммуникации Издательство БХВ-Петербург, 2012, ISBN:978-5-9775-0666-3, 361 с.

C. Resurse Internet

7. Situl oficial IDE Eclipse [online] data 05.03.2015 Disponibil pe internet < www.eclipse.org >
8. Situl oficial IDE Android Studio [online] data 05.03.2015 Disponibil pe internet < www.developer.android.com/studio/index.html>
9. Documentația oficială Android [online] data 05.03.2015 Disponibil pe internet < www.developer.android.com>

D. Tutoriale video Programare Android

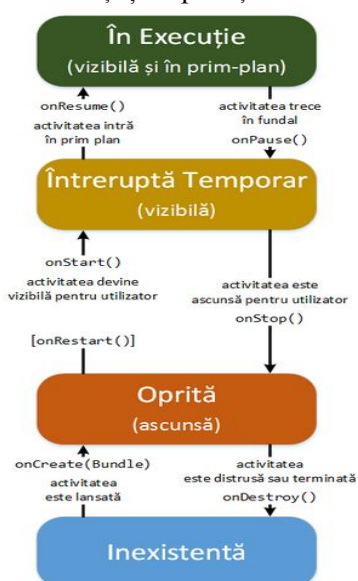
10. Tutorial video programarea Android în limba română [online] data 05.03.2015 Disponibil pe internet <<http://www.androidro.ro/tutoriale-android/>>
11. Tutorial video programarea Android în limba rusă [online] data 05.03.2015 Disponibil pe internet <www.intuit.ru>
12. Tutorial video programarea Android în limba rusă [online] data 05.03.2015 Disponibil pe internet <www.habrahabr.ru/post/230559/>
13. Tutorial video programarea Android în limba rusă [online] data 05.03.2015 Disponibil pe internet <www.youtube.com/playlist?list=PLyfVjOYzujungWKGPAOz0iYXTUbP1WW27p>
14. Tutorial video programarea Android în limba rusă [online] data 05.03.2015 Disponibil pe internet <www.fandroid.info/razrabotka-v-android-studio-prilozhenij-dlya-android-s-nulya-na-yazyke-java-videouroki/>

Exemplu de itemi pentru testul de la examen

la disciplina „Programarea aplicațiilor pe dispozitive mobile”

I. Completează spațiile libere cu noțiunea corectă.

1. Intenție reprezintă o instanță a clasei _____
 2. Componenta unei aplicații Android ce oferă o interfață grafică cu care utilizatorul poate să interacționeze se numește _____
 3. Structura unui instantaneu este formată din: _____
-
4. Enumerați proprietățile principale din cadrul unei aplicații Android: _____
-
5. Enumerați și explicați ciclul de viață al activității:



1. _____
2. _____
3. _____
4. _____

II. Alege varianta corectă.

1. În clasele de tip Activity sau Service, contextul aplicației poate fi obținut printr-un apel de forma:

a. Context context = getApplicationContext();	c. Context = getApplicationContext();
b. Context context = getContext();	d. Context context = findViewById();
2. Tranziția unei activități dintr-o stare în alta este notificată prin intermediul unor metode:

a. onCreate(Bundle);	c. findViewById(int);
b. setContentView(int);	d. onSaveInstanceState();
3. Încărcarea conținutului din obiectul de tip Bundle (în vederea restaurării stării) poate fi realizată:

a. onResume	b. onPause	c. onCreate	d. onStart
-------------	------------	-------------	------------
4. Rezultatul compilării este fișierul cu extensia ...

a. *.java	b. *.apk	c. *.src	d. *.exe
-----------	----------	----------	----------

5. Tipurile de containere pentru colecții de date sunt:

a. ListView; b. ExpandableListView; c. GridView; d. Spinner;

III. Explică diferența dintre noțiunile:

- a) activitate întreruptă temporar și oprită;
- b) clasă și metodă;

IV. Îndeplinește cerințele în baza următorului algoritm:

ViewSwitcher image animation-----

```

    ((ViewSwitcher) findViewById(R.id.switcher)).setOnClickListener(new View.OnClickListener() {
        @Override
  
```

```

public void onClick(View v) {
    ViewSwitcher switcher = (ViewSwitcher) v;
    if (switcher.getDisplayedChild() == 0) {
        switcher.showNext();
    } else {
        switcher.showPrevious();
    }
}
});

```

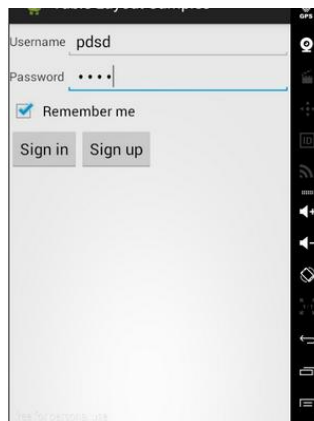
- a. Completează în secvență de algoritm spațiile libere;
- b. Numește clasele prezente în program

- c. Numește obiectele definite în program

- d. Care va fi efectul execuției codului programului alăturat?

V. **Explică esența noțiunilor de *Intenție, activitate*;**

VI. **Definiți algoritmul de generare a unei interfețe similare cu cea prezentată în figura de mai jos.**



PUNCTE ACORDATE

SUBIECTUL	I	II	III	IV	V	VI	Total
PUNCTAJUL	10	5	5	5	10	15	50