

**Ministerul Educației al Republicii Moldova
Universitatea de Stat „Alec Russo” din Bălți
Facultatea de Științe Reale, Economice și ale Mediului
Catedra de matematică și informatică**

Curriculum

la unitatea de curs

Ingineria programării

Ciclul I, studii superioare de licență

Codul și denumirea domeniului general de studiu: 44 Științe exacte

Codul și denumirea specialității: 444.1 Informatica

Forma de învățământ: cu frecvență redusă

**Autor: Mircea PETIC,
dr., conf. univ.**

Bălți, 2017

Discutat și aprobat la ședința Catedrei de matematică și informatică
Procesul-verbal nr. 15 din 09.06.2017

Șeful Catedrei de matematică și informatică

E. Plohotniuc conf. univ., dr. Eugeniu PLOHOTNIUC

Discutat și aprobat la ședința Consiliului Facultății de Științe Reale,
Economice și ale Mediului.

Procesul-verbal nr. 15 din 27.06.2017

Decanul Facultății de Științe Reale, Economice și ale Mediului

I. Ciobanu conf. univ. dr. Ina CIOBANU



Informații de identificare a unității de curs

Facultatea: Științe Reale, Economice și ale Mediului

Catedra de matematică și informatică

Domeniului general de studii: 44 Științe exacte

Domeniul de formare profesională: 444 Informatica

Denumirea specialității: 444.1 Informatica

Denumirea unității de curs: Ingineria programării

Administrarea unității de curs

Codul unității de curs	Credite ECTS	Total ore	Repartizarea orelor				Forma de evaluare	Limba de predare
			Prelegeri	Seminar	Laborator	Lucrul indiv.		
S.06.A.136	5	150	12	-	18	30	Examen	Română

Anul de studii și semestrul în care se studiază: anul III, sem. VI

Regimul unității de curs: obligatorie (la liberă alegere)

Categoria formativă: unitate de curs de specializare.

Informații referitoare la cadrele didactice



Mircea Petic, dr. în informatică, conferențiar universitar. Absolvent al Universității de Stat „A. Russo” din Bălți, specialitatea „Informatica și limba engleză aplicată”. A efectuat stagii în diverse universități, inclusiv peste hotare, unde s-a specializat în domeniul informaticii, utilizarea tehnologiei informației și a comunicațiilor în învățământ, procesarea limbajului natural, programarea obiect orientată, programarea logică și funcțională, ingineria

programării, managementul proiectelor.

E-mail: petic.mircea@yahoo.com

Orele de consultații - miercuri: 15.00 -17.00. Consultațiile se oferă atât în regim „față-în-față”, cât și prin utilizarea poștei electronice, YahooMessenger (petic.mircea) și Skype sau rețeaua socială facebook. Numele în Skype – mir_cescu.

Descrierea unității de curs

Unității de curs „*Ingineria programării*” este un curs de specializare care permite aprofundarea cunoștințelor în programare. Realizarea unui sistem informatic reprezintă o activitate complexă și de durată, ce antrenează mari resurse materiale, umane și de timp. IT-ul este doar o industrie printre alte industrii care întreprind proiecte. Asemenea altor domenii, proiectele IT au specificul său metodologic. Este nevoie de cunoștințe de inginerie software pentru a livra valoare în mod predictibil, pentru a oferi informații reale atât echipei, cât și managementului companiei și pentru a putea lua decizii bazate pe rezultate și metrici concrete. Până la urmă, „*Ingineria programării*” este o disciplină care își propune să crească gradul de siguranță al produselor software în condițiile unui mediu în continuă schimbare.

În cadrul unității de curs se studiază noțiuni de: proiect, ingineria programării, ingineria cerințelor, limbajele de modelare, testarea software, drepturile de autor. Viitorii programatori vor obține cunoștințe referitor la diverse modele de dezvoltare software, vor căpăta deprinderi de lucru cu limbajul de modelare UML.

Integrarea în programul de studii

Studierea unității de curs „*Ingineria programării*” se sprijină pe cunoștințele, capacitățile și competențele dezvoltate în cadrul unității de curs „*Programarea Java*”, „*Programarea obiect orientată*”, „*Arhitectura calculatoarelor și sisteme de operare*”, „*Gestiunea informației*”, „*Programarea Web*”, „*Rețele de calculatoare*”, studiată la ciclul I. Finalitățile și conținutul unității de curs sînt corelate cu finalitățile și conținuturile unităților de curs menționate mai sus.

Competențe prealabile

- a) *Algoritmi și programare*: Tipuri de date simple și structurate. Instrucțiunile simple și compuse (condiționale și repetitive).
- b) *Programarea obiect orientată*: Clase. Moștenire. Polimorfism. Obiecte.
- c) *Arhitectura calculatoarelor și sisteme de operare*: componentele calculatoarelor, sistemele de operare Windows, Linux.
- d) *Gestiunea informației*: SGBD, proiectarea bazelor de date.
- e) *Programarea Web*: client, server, site.

Competențele dezvoltate în cadrul unității de curs

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.

CP6. Prelucrarea datelor, analiza și interpretarea lor.

Competențe transversale:

CT1. Aplicarea regulilor de muncă riguroasă și eficientă, manifestarea unei atitudini responsabile față de domeniul profesional, pentru valorificarea optimă și creativă a propriului potențial în situații specifice, cu respectarea principiilor și a normelor de etică profesională.

Finalitățile unității de curs

La finalizarea studierii unității de curs studentul va fi capabil să

1. determine resurselor necesare realizării proiectului unui sistem software;
2. analizeze atributele măsurabile ale calității produsului software;
3. explice modul de distribuire a forței de muncă pentru elaborarea unui proiect software.
4. proiecteze și să aplice setul de teste care ar verifica produsul software elaborat.

Conținuturi

a) Prelegeri

Nr. d/o	Subiectele de studiu	Nr. de ore
1	Noțiune de proiect. Exemple de proiecte IT. Motivația studierii ingineriei programării. Erori celebre în programare.	2
2	Etapele de dezvoltare a software. Modele de dezvoltare software.	2
3	Ingineria cerințelor. Modelarea arhitecturii programului. Limbajele de modelare.	2
4	Asigurarea calității software. Testarea software. Atributele măsurabile în asigurarea calității programelor.	2
5	Distribuirea forței de muncă în timp. Drepturile de autor.	2
6	Lucrarea de control	2
Total		12

b) Laborator

Nr. d/o	Tematica lecțiilor de laborator	Nr. de ore
1	Modelarea diagramelor UseCase. Diagrame de activități UML. Diagrame de stări.	2
2	Diagramele de clasă UML. Diagramele de obiecte.	2
3	Sarcini individuale nr. 1	2
4	Diagramă de structură composită. Diagramă de activități.	2
5	Diagramele comportamentale UML. Diagramele de secvență UML.	2
6	Sarcini individuale nr. 2	2
7	Diagramă de pachete. Diagrame de deployment UML.	2
8	Sarcini individuale nr. 3	2
9	<i>Susținerea proiectului programării</i> <i>Ingenieria</i>	2
Total		18

Activități de lucru individual

Elaborarea diagramelor UML pentru un proiect de sistem informațional stabilit. Rezultatele vor fi prezentate într-un raport editat într-un document Word pe 2-3 pagini format A4, font #12, 1.5 intervale. Activitatea sa va fi evaluată atât de către colegi cât și de către titularul disciplinei într-o ședință aparte.

Criterii de evaluare:

1. Exactitatea proiectării diagramelor UML;
2. Argumentarea abordării propuse spre implementare;
3. Corectitudinea (logică, ortografică) a raportului prezentat..

Studentul va prezenta raportul însoțit de fișierele sursă care demonstrează editarea diagramelor UML.

Termenul limită (deadline) de prezentare a sarcinii – perechea a 9-a (laborator)

Resursele informaționale la unitatea de curs

A. Literatura de bază

1. TODORAN, E. *Inginerie SoftWare*, Editura MediaMira, Cluj-Napoca, 2001, ISBN: 973-9358-73-X, 240 p.
2. ГЛАСС, Р. *Факты и заблуждения профессионального программирования*, Символ+, Москва, 2008, 232 с.
3. МАРАСКО, Д. *IT проекты. Фронтвые очерки*, Символ+, Москва, 2008, 376 с.

B. Literatura suplimentară

4. SOMMERVILLE, I. *Software Engineering*, Addison Wesley, Boston, 2001, 773 p.
5. LARMAN C. *Applying UML and Patterns*, Addison Wesley, Boston, 2002, 615 p.
6. GAMMA, E. HELM, R., JOHNSON R., VISSIDES J., *Design Patterns, Elements of Reusable Object-Oriented Software*, Addison Wesley, Boston, 1998, 431 p.

C. Resurse Internet

7. Situl oficial Argo UML [on-line], [vizitat 05.05.2015], Disponibil:

< <http://argouml.tigris.org/> >

Evaluarea

Cunoștințele, capacitățile și competențele studenților vor fi evaluate:

1. La prelegeri (**PR**):
 - 1.1. *Lucrarea de control*: perechea a 6-a (**LC1**).
2. În cadrul lecțiilor de laborator (**LLab**):
 - 2.1. *Sarcini individuale nr. 1*: perechea a 3-a (**SI1**)
 - 2.2. *Sarcini individuale nr. 2*: perechea a 6-a (**SI2**)
 - 2.3. *Sarcini individuale nr. 3*: perechea a 8-a (**SI3**)
 - 2.4. *Proiect*: perechea a 9-a (**Pro**)
3. La examenul final, conform orarului întocmit de decanat (**Ex**).
Nota finală la disciplina „Ingineria programării” se calculează conform formulelor:

$$N_{\text{evaluarea_curentă}} = 1/2 \times \text{PR} + 1/2 \times \text{LLab}$$

$$N_{\text{finală}} = 0,5 \times N_{\text{evaluarea_curentă}} + 0,5 \times N_{\text{examen}},$$

Unde **PR=LC1** și **LLab=(SI1+SI2+SI3+Pro)/4**

Examenul final se susține scris, care va include un test complex cu diferite tipuri de itemi. Pentru a fi admis la examen, este obligator ca ambele note (**PR** și **LLab**) să fie cel puțin 5.0.

Recuperarea notelor și susținerea repetată a examenului are loc în datele stabilite de orarul întocmit de decanat de susținere a restanțelor.

Baremul de apreciere

(în baza REGULAMENTULUI

cu privire la evaluarea rezultatelor academice ale studenților în
Universitatea de Stat „Alec Russo” din Bălți)

Procentajul	Nota
100 – 91	10
90 – 81	9
80 – 71	8
70 – 66	7
65 – 61	6
60 – 51	5

50 – 41	4
40 – 31	3
30 – 16	2
15 – 0	1

Principiile de lucru în cadrul disciplinei

1. 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.
2. Nu este salutăată întârzierea la ore.
3. 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.
4. 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” .

