

**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

Programarea orientată pe obiect II

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

**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

Eugeniu 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

Ina 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: Programarea orientată pe obiect II

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.04.A.129	5	150	30	-	45	75	Examen	Română

Anul de studii și semestrul în care se studiază: anul II, sem. IV

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

Unitatea de curs „*Programarea orientată pe obiect II*” este un curs de specializare care permite aprofundarea cunoștințelor în programare. Programarea orientată pe obiect este unul din cei mai importanți pași făcuți în evoluția limbajelor de programare spre o mai puternică abstractizare în implementarea programelor. Ea a apărut din necesitatea exprimării problemei într-un mod mai natural ființei umane. Astfel unitățile care alcătuiesc un program se apropie mai mult de modul nostru de a gândi decât modul de lucru al calculatorului.

Unitatea de curs „*Programarea orientată pe obiect II*” este continuarea disciplinei „*Programarea orientată pe obiect I*”. În cursul dat vor fi studiate posibilitățile de modelare a claselor cu mijloacele limbajului UML, principiile de programare SOLID, noțiune de pattern de programare, cât și paternul de programare MVC.

Integrarea în programul de studii

Studierea unității de curs „*Programarea orientată pe obiect II*” se sprijină pe cunoștințele, capacitățile și competențele dezvoltate în cadrul unității de curs „*Programarea obiect orientată I*”, „*Bazele programării I și II*”, „*Gestiunea informației*” și „*Programarea structurată*”, studiate la ciclul I. Finalitățile și conținutul unității de curs sunt corelate cu finalitățile și conținuturile unităților de curs menționate mai sus.

Competențe prealabile

- a) *Bazele programării*: Tipuri de date simple și structurate. Instrucțiunile simple și compuse (condiționale și repetitive).
- b) *Programarea obiect orientată I*: Clase. Moștenire. Polimorfism. Obiecte.
- c) *Gestiunea informației*: SGBD, proiectarea bazelor de date.
- d) *Programarea structurată*: subprogram, parametri, apel de subprogram

Competențele dezvoltate în cadrul unității de curs

Competențe profesionale:

CP1. Operarea cu fundamentele științifice ale informaticii și matematicii 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.

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.

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 disciplinei studentul va fi capabil să:

1. posedă deprinderi de lucru cu clase, metode abstracte și interfețe;
2. analizeze eficiența utilizării diferitor patterne de programare;
3. explice modul de implementare a patternului de programare MVC.
4. proiecteze aplicații soft conform principiului de programare SOLID.

Conținuturi

a) Prelegeri

Nr. d/o	Subiectele de studiu	Nr. de ore
1	Paradigme de programare. Principii de bază ale POO: abstractizare, încapsulare, moștenire și polimorfism.	2
2	Metode și proprietăți. Constructori și destructori. Constante. Instanțe ale clasei.	2
3	Crearea modelului conceptual. Diagrame UML.	2
4	Domeniu de vizibilitate: public, protected și private. Încapsularea. Utilizarea setter-ilor și getter-ilor.	2
5	Clase abstracte și interfețe. Moștenire multiplă.	2
6	Clase și metode statice. Clase finale. Metode virtuale.	2
7	Lucrarea de control nr. 1	2
8	Prelucrarea excepțiilor.	2
9	Clonarea și transmiterea obiectelor prin referință. Serializare. Deserializare.	2
10	Principii de programare SOLID.	2
11	Patterni de programare. Clasificare.	2
12	Patternul MVC. Componente și arhitectură.	2
13	Implementarea MVC.	2
14	Lucrarea de control nr. 2	2
15	Recapitulare	2
Total		30

b) Laborator

Nr. d/o	Tematica lecțiilor de laborator	Nr. de ore
1	Paradigme de programare.	2
2	Principii de bază ale POO: abstractizare, încapsulare. Metode și proprietăți. Constructori și destructori.	2
3	Principii de bază ale POO: moștenire	2
4	Principii de bază ale POO: polimorfism.	2
5	Diagrame de clase.	2
6	Diagramele de obiecte. Diagrame de pachete.	2
7	Sarcini individuale nr. 1	2
8	Diagramă de structură compozită. Diagramele de secvență.	2
9	Domeniu de vizibilitate: public, protected și private.	2
10	Încapsularea. Utilizarea setter-ilor și getter-ilor.	2
11	Clase abstracte și interfețe.	2
12	Moștenire multiplă.	2
13	Sarcini individuale nr. 2	2
14	Clase și metode statice.	2
15	Clase finale.	2
16	Metode virtuale.	2
17	Prelucrarea excepțiilor.	2
18	Clonarea și transmiterea obiectelor prin referință.	2
19	Serializare. Deserializare.	2
20	Pattern de programare.	2
21	Implementarea MVC.	2
22	Susținerea proiectului POO II	2
23	Recapitulare	1
Total		45

Activități de lucru independent

Elaborarea unei aplicații bazate pe noțiunile programării orientate pe obiecte 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 ținând cont de principiile programării orientate pe obiecte;
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 22-a (laborator)

Resursele informaționale la unitatea de curs

A. Literatura de bază

1. ГОМА, X. *UML Проектирование систем реального времени, распределенных и параллельных приложений*. ДМК, Москва, 2016, 700 стр.
2. WEISFELD, M. *The Object-Oriented Thought Process*. Addison-Wesley Professional, 2013.336 p.

B. Literatura suplimentară

3. LARMAN, C. *Applying UML and Patterns*, Addison Wesley, Boston, 2002, 615 p.
4. GAMMA, E., HELM, R., JOHNSON, R., VISSIDES, J., *Design Patterns, Elements of Reusable Object-Oriented Software*, Addison Wesley, Boston, 1998, 431 p.
5. ШИЛДТ, Г. *Java 8. Полное руководство*. Oracle Press. 2015, 1377 стр.

C. Resurse Internet

6. Situl oficial Argo UML [online], [vizitat 05.05.2015] Disponibil:
< <http://argouml.tigris.org/> >

Evaluarea

Cunoștințele, capacitățile și competențele studenților vor fi evaluate:

1. La prelegeri (**PR**):
 - 1.1. *Lucrare de control scrisă nr. 1*: perechea a 6-a (**LC1**).
 - 1.2. *Lucrare de control scrisă nr. 2*: perechea a 14-a (**LC2**)
2. În cadrul lecțiilor de laborator (**LLab**):
 - 2.1. *Sarcini individuale nr. 1*: perechea a 6-a (**SI1**)
 - 2.2. *Sarcini individuale nr. 2*: perechea a 13-a (**SI2**)
 - 2.3. *Proiect*: perechea a 22-a (**Pro**)
3. La examenul final, conform orarului întocmit de decanat (**Ex**).
Nota finală la disciplina „Programarea orientată pe obiect IP” se calculează conform formulelor:

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

$$N_{\text{finală}} = 0,6 \times N_{\text{evaluarea_curentă}} + 0,4 \times N_{\text{examen}}$$

Unde **PR**=(**LC1**+**LC2**)/2 și **LLab**=(**SI1**+**SI2**+**Pro**)/3

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” .

