

**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 logică**

**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

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

**Denumire unității de curs:** Programarea logică.

### 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.03.A.120             | 5            | 150       | 30                  | -       | 45        | 75            | Examen            | Română           |

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

**Regimul unității de curs:** obligatorie (la libera 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 „Alec 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](mailto: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 logică*” este un curs facultativ 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. La baza limbajului de programare logică stau elementele logicii. Procesul deducerii unei concluzii în baza unor condiții prealabile se numește deducție logică. Prologul este mai adaptat pentru rezolvarea problemelor inteligenței artificiale. Limbajul Prolog este destul de bine adaptat și la soluționarea altor probleme, fapt care îl determină să fie destul de atragător și de perspectivă. În Prolog pot fi realizate diverse probleme.

Limbajul de programare Prolog ne atrage prin simplitatea sa. Un program scris în Prolog este ușor de citit. Datorită formalismelor universale ale frazelor lui Horn, textul programului scris în Prolog, în comparație cu programele scrise în alte limbaje, este mai puțin supus influenței particularităților ce țin de gestionarea calculelor în calculator. În cursul dat vor fi studiate aspecte programării logice în baza limbajului de programare Prolog.

## **Integrarea în programul de studii**

Studierea unității de curs „*Programarea logică*” se sprijină pe cunoștințele, capacitățile și competențele dezvoltate în cadrul unității de curs „*Bazele programării*” și „*Programarea structurată*” studiate în ciclul I de licență. Finalitățile și conținutul unității de curs sînt corelate cu finalitățile și conținuturile unităților de curs menționate mai sus.

## Competențe prealabile

1. *Bazele programării*: Tipuri de date simple și structurate, expresii.
2. *Programarea structurată*: Instrucțiunile simple și compuse (condiționale și repetitive). Funcții. Proceduri. Recursivitate.

## 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. definească conceptul programare logică;
2. cunoască destinația și modul de lucru cu predicate;
3. posede deprinderi de lucru cu entitățile limbajului Prolog;
4. utilizeze corect listele;
5. inducă ideea utilizării corecte a programării logice în funcție de necesitățile aplicațiilor;
6. explice corect modul de programare logică;
7. explice corect diferența între programarea logică și alte paradigme de programare (funcțională, imperativă);

## Conținuturi

### a) Prelegeri

| Nr. d/o   | Subiectele de studiu  | Nr. de ore |
|---|---|------------|
| 1   | Programare și limbaje de programare. Programare imperativă vs. programare declarativă.                          | 2          |
| 2   | Recursivitate   | 2          |
| <b><i>Limbaajul de programare logică - Prolog</i></b> |   |            |
| 3   | Elemente fundamentale ale limbajului Prolog. Fapte și reguli Prolog. Întrebări. Strategia de control în Prolog. | 2          |
| 4   | Variabile și propoziții compuse. Variabile anonime. Reguli de definire a potrivirilor. Model de flux. Exemple.  | 2          |
| 5   | Programul Prolog. Domenii predefinite. Întrebări. Predicate cu aritate multiplă.                                | 2          |
| 6   | Simbolul IF (Prolog) și instrucțiunea IF (alte limbaje). Expresii aritmetice și comparații.                     | 2          |
| 7   | <b>Lucrarea de control nr. 1</b>  | <b>2</b>   |
| 8   | Controlarea backtracking-ului în Prolog. Predicatele fail și ! (cut). Utilizarea lui !.                         | 2          |
| 9   | Tipuri de tăieturi. Predicatul "not". Liste Prolog  | 2          |
| 10  | Recursivitate. Exemple de tratare a backtracking-ului.  | 2          |

|              |   |           |
|--------------|---|-----------|
|              | Găsirea tuturor soluțiilor în același timp.   |           |
| 11           | Exemple de predicate Prolog. Predicate nedeterminate.   | 2         |
| 12           | Liste eterogene. Arbori. Backtracking cu ciclări.<br>Exemple de proceduri recursive. Cadrul stivei.     | 2         |
| 13           | Optimizarea prin recursivitate de coadă. Utilizarea tăieturii pentru păstrarea recursivității de coadă. | 2         |
| 14           | <b>Lucrarea de control nr. 2</b>  | <b>2</b>  |
| 15           | <b>Recapitulare</b>   | <b>2</b>  |
| <b>Total</b> |   | <b>30</b> |

*b) Laborator*

| Nr. d/o | Tematica lecțiilor de laborator                           | Nr. de ore |
|---------|---|------------|
| 1       | Fapte. Întrebări. Termeni.                                | 2          |
| 2       | Reguli. Reguli recursive.                                 | 2          |
| 3       | Operatori în Prolog.                                      | 2          |
| 4       | Expresii aritmetice. Evaluarea expresiilor.               | 2          |
| 5       | Reprezentarea și unificarea listelor.                     | 2          |
| 6       | Terminarea prelucrării când lista e vidă.                 | 2          |
| 7       | Terminarea prelucrării când elementul specificat e găsit. | 2          |
| 8       | Terminarea prelucrării când țelul specificat e atins.     | 2          |
| 9       | <b>Sarcini individuale nr. 1</b>                          | <b>2</b>   |
| 10      | Argumentul adițional pentru fixarea datelor intermediare. | 2          |
| 11      | Predicatul tăierea.                                       | 2          |
| 12      | Tăierea verde.  | 2          |
| 13      | Tăierea roșie.  | 2          |
| 14      | Predicatul fail/0 și conjuncția fail,!                    | 2          |
| 15      | Bucle repetitive.   | 2          |
| 16      | Negația prin eşec.  | 2          |
| 17      | Capcanele regulilor implicite.                            | 2          |
| 18      | Modificarea dinamică a programului.                       | 2          |

|              |   |           |
|--------------|---|-----------|
| 19           | Tehnici de sortare a secvențelor.           | 2         |
| 20           | Tratarea mulțimilor.                        | 2         |
| 21           | <b>Sarcini individuale nr. 2</b>            | <b>2</b>  |
| 22           | <b><i>Susținerea proiectului Prolog</i></b> | <b>2</b>  |
| 23           | Recapitulare                                | 1         |
| <b>Total</b> |   | <b>45</b> |

### **Activități de lucru independent**

Proiectul va presupune elaborarea unei aplicații Prolog care rezolvă o problemă din viața reală. 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 prezentate 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 într-o ședință aparte.

Criterii de evaluare:

- Corectitudinea rezolvării problemei prin elaborarea aplicației  
Scala;
- Relevanța și valoarea comentariilor;
- Exactitate (logică, ortografică) a raportului prezentat;

**Termenul limită (deadline) de prezentare a sarcinii – perechea a 22-a (laborator)**



## Resursele informaționale la unitatea de curs

### A. Literatura de bază

1. COTELEA, V. *Programare logică*. Editura „Nestor”, Chișinău, 2000, 394 p.
2. МАРКОВ, В. А. *Современное логическое программирование на языке Visual Prolog 7.5*, ДМК, Москва, 2016, 544 стр.

### B. Literatura suplimentară

3. MASALAGIU, C. D. *Fundamentele logice ale Informaticii*, Editura Universității „Al. I. Cuza”, Iași, 2004, 408 p.
4. CAZACU, C., SLABU, V. *Logică Matematică*, Editura „Ștefan Lupașcu”, Iași, 1999, 288 p.

### C. Resurse Internet

5. Online compiler and IDE [online], [vizitat 09.10.2016] Disponibil:  
< <http://ideone.com/> >
6. Documentația SWI Prolog [online], [vizitat 09.10.2016] Disponibil:  
< <http://www.swi-prolog.org/> >

### D. Tutoriale video *Programarea logică*

7. Tutorial video programarea logică în limba rusă [online] [vizitat 09.10.2016] Disponibil:

< <https://www.youtube.com/playlist?list=PLDrmKwRSNx7JhMchOf54epeoG3vO7jM7z>>

## Evaluarea

Cunoștințele, capacitățile și competențele studenților vor fi evaluate:

1. La prelegeri (**PR**):
  - 1.1. *Lucrare de control scrisă nr. 1*: perechea a 7-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 9-a (**SI1**)
  - 2.2. *Sarcini individuale nr. 2*: perechea a 21-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 logică” 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 „Alecu 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” .

