

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 pentru unitatea de curs**

# **Programarea Java. Teza anuală**

**Specialitatea: 444.1 Informatica (științe exacte)**

***Studii cu frecvența la zi***

**Bălți, 2015**

Curriculumul a fost discutat la ședința Catedrei de  
matematică și informatică

Procesul verbal nr. 9 din 7 mai 2015

Șeful catedrei dr. conf. univ. \_\_\_\_\_

Eugeniu Plohotniuc

Curriculumul a fost aprobat la ședința Consiliului

Facultății de Științe Reale, Economice și ale Mediului

Procesul verbal nr. 15 din \_\_\_ iunie 2015

Decanul facultății, dr. hab., prof. univ. \_\_\_\_\_

Pavel Topala

## Informații de identificare a disciplinei

**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 disciplinei:** *Programarea Java. Teza anuală.*

### Administrarea unității de curs:

Codul unității de curs	Credite ECTS	Total ore	Repartizarea orelor				Forma de evaluare	Limba de predare
			Prel.	Sem.	Lab.	L. ind		
S.05.O.56	6	180	45	-	45	90	Ex. scris	română

**Anul de studii și semestrul în care se studiază disciplina:** anul III, semestrul V

**Regimul disciplinei:** obligatorie

**Categoria formativă:** unitate de curs de specializare.

### Informații referitoare la cadrele didactice



Titularul cursului - *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](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 disciplinei

Disciplina "Programarea Java" 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. Java este o tehnologie inovatoare lansată de compania Sun Microsystems în 1995, care a avut un impact remarcabil asupra întregii comunități a dezvoltatorilor de software, impunându-se prin calități deosebite cum ar fi simplitate, robustețe și nu în ultimul rând portabilitate.

În cadrul disciplinei se studiază noțiuni de: platforma Java, mașina virtuală Java, serializare, excepții, applet, servlet. Viitorii programatori vor obține cunoștințe referitor la lucrul cu diverse medii de programare în limbajul Java (precum IDE Eclipse, IDE NetBeans și IDE IntelliJIdea) și clase Java.

## Integrarea în programul de studii

Studierea unității de curs „*Programarea Java*” se sprijină pe cunoștințele, capacitățile și competențele dezvoltate în cadrul disciplinei „*Bazele programării*”, „*Programarea obiect orientată*”, „*Baze de date*” și „*Programarea Web*”, studiate la ciclul I. Finalitățile și conținutul unității de curs sînt corelate cu finalitățile și conținuturile unităților de curs menționate mai sus.

## **Prealabile/precondițiile**

1. *Algoritmi și programare*: Tipuri de date simple și structurate. Instrucțiunile simple și compuse (condiționale și repetitive).
2. *Programarea obiect orientată*: Clase. Moștenire. Polimorfism. Obiecte.
3. *Baze de date*: SGBD, proiectarea bazelor de date, secvențe SQL.
4. *Programarea Web*: HTML, CSS, formulare, client, server.

## **Competențele dezvoltate în cadrul disciplinei**

### *Competențe generice*

1. Competențe cognitive: căutarea, procesarea și analiză critică a informației din diverse surse și de prezentare a informației în diferite forme.
2. Competențe de cooperare și lucru în echipă: utilizarea cunoștințelor și deprinderilor specifice pentru a crea oportunități de realizare a sarcinilor de învățare în colaborare cu colegii.
3. Competențe în domeniul TIC: integrarea TIC în procesul de căutare, prelucrare și asimilare a noilor cunoștințe în cadrul disciplinei.

### *Competențe specifice*

1. Competențe cognitive: de cunoaștere și înțelegere a principalelor noțiuni legate de platforma Java și noțiuni de programare obiect orientată (moștenire, polimorfism, încapsulare).
2. Competențe de aplicare: de proiectare și elaborare a aplicațiilor de tip consolă, desktop, appleturi utilizând posibilitățile platformei Java.
3. Competențe de analiză: a posibilităților tehnologiei Java în proiectarea și elaborarea aplicațiilor ce îndeplinesc condițiilor problemelor din viața reală.

## Finalitățile disciplinei

La finalizarea studierii disciplinei studentul va fi capabil

1. să definească conceptul de Mașină Virtuală Java.
2. să cunoască destinația și modul de lucru cu clasele esențiale în Java
3. să posede deprinderi de lucru cu clase, metode abstracte și interfețe;
4. să utilizeze corect componentele AWT la programarea vizuală
5. să posede deprinderi de lucru cu biblioteca Swing
6. să creeze și utilizeze corect applet-urile
7. să explice corect modul de conectare la o bază de date cu interfața JDBC
8. să efectueze corect secvențe SQL

## Calendarul unității de curs

*Prelegeri (Lecții teoretice)*

Prelegerea	Conținuturi	Nr. de ore
1	Momente istorice tehnologia Java. Platforma Java. Lexicul limbajului Java. Operații. Instrucțiuni.	2
2	IDE de lucru cu tehnologia Java. IDE Eclipse. IDE NetBeans. IDE IntelliJIDEA.	2
3	Metode. Clase. Constructori. Obiecte. Clase interioare. Moștenire.	2
4	Polimorfism. Încapsulare. Clase abstracte. Interfețe. Modificatori. Pachete.	2
5	Clasa JFrame. Componente vizuale. Poziționarea componentelor. Meniuri.	2
6	Cutii de dialog. Evenimente.	2
7	<b>Lucrare de control nr. 1</b>	<b>2</b>
8	Masive. Colecții generice. Interfața Iterator. Tratarea excepțiilor.	2
9	Clasa String. Clasa StringTokenizer. Parametrii metodei main().	2
10	Clase înfășurătoare. Lucru cu numere mari.	2
11	Fluxuri de date. Fire de execuție.	2
12	Expresii regulate.	2

13	XML. Parsare SAX și DOM.	2
14	Serializarea obiectelor. Programarea în rețea.	2
15	<b>Lucrare de control nr. 2</b>	<b>2</b>
16	Generalități de lucru cu baze de date. JDBC. Conectare la o bază de date.	2
17	Efectuare de secvențe SQL.	2
18	Contextul grafic de desenare. Clasa Canvas.	2
19	Appleturi. Interfața grafică.	2
20	Servlets. Pagini JSP. Formularele web. Transmiterea datelor cu metodele GET și POST.	2
21	Sesiuni și cookies. Forward și redirect. Filtre web. Dependency injection.	2
22	<b>Lucrare de control nr. 3</b>	<b>2</b>
23	<b>Recapitulare</b>	<b>1</b>
<b>Total</b>		<b>45</b>

#### *Lecții de laborator*

<b>Lecția</b>	<b>Tematica lecțiilor de laborator</b>	<b>Nr. de ore</b>
1	Platforma Java. Prima aplicație Java. Lexicul limbajului Java. Operații. Instrucțiuni.	2
2	IDE Eclipse	2
3	IDE NetBeans	2
4	IDE IntelliJIDEA	2
5	Metode. Clase. Constructori. Obiecte. Clase interioare. Moștenire.	2
6	Polimorfism. Încapsulare. Clase abstracte. Interfețe. Modificatori. Pachete.	2
7	Clasa JFrame. Componente vizuale. Poziționarea componentelor. Meniuri.	2
8	Cutii de dialog. Evenimente.	2
9	<b>Sarcini individuale nr. 1</b>	<b>2</b>
10	Masive. Colecții generice. Interfața Iterator. Tratarea excepțiilor.	2
11	Clasa String. Clasa StringTokenizer. Parametrii metodei main().	2
12	Clase înfășurătoare. Lucru cu numere mari.	2
13	Fluxuri de date. Fire de execuție.	2
14	Expresii regulate.	2
15	Generalități de lucru cu baze de date. JDBC.	2

	Conectare la o bază de date.	
16	Efectuare de secvențe SQL.	2
17	Contextul grafic de desenare. Clasa Canvas.	2
18	Appleturi. Interfața grafică.	2
19	Servlets. Pagini JSP. Formularele web. Transmiterea datelor cu metodele GET și POST.	2
20	Sesiuni și cookies. Forward și redirect. Filtre web. Dependency injection.	2
21	<b>Sarcini individuale nr. 2</b>	<b>2</b>
22	<b><i>Susținerea proiectului Java</i></b>	<b>2</b>
23	Recapitulare	1
<b>Total</b>		<b>45</b>

### **Sarcină pentru lucrul la proiect**

Proiectul va presupune elaborarea unei aplicații Java care rezolvă o problemă din viața reală. Aplicația trebuie să aibă o interfața grafică. Ea poate fi desktop sau Web. 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 Java;
- Relevanța și valoarea comentariilor;
- Exactitate (logică, ortografică) a raportului prezentat;

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



## Exemple de teme pentru proiect

1. Numărarea cuvintelor dintr-un fișier txt scris în limba română cu diacritice (cel puțin 1000 cuvinte). Separatori de cuvinte sînt: spațiile libere, semnele de punctuație, simbolul "-".
2. Este nevoie de a automatiza evidența iepurilor din crescătorie. Se cunosc cîți iepuri sînt la începutul fiecărei luni, cîți au murit și cîți s-au născut în cursul fiecărei luni.
3. Determinarea greutății ideale a unei persoane cunoscînd înălțimea, vîrsta și sexul persoanei. Formulele de calcul sînt:  $G_{\text{masculin}} = 50 + 0.75 * (\text{înălțime} - 150) + (\text{vîrsta} - 20)/4$ ,  $G_{\text{feminin}} = G_{\text{masculin}} - 10$ , unde înălțimea este exprimată în cm și vîrsta în ani. Sexul indică separat.
4. O broscuță se deplasează efectuînd cîte o săritură de lungime  $p$  cm la fiecare secundă. După fiecare  $n$  secunde broscuța devine mai obosită, iar lungimea săriturii pe care o face se înjumătățește. Este nevoie de a scrie un program care citind lungimea inițială a săriturii,  $p$ , numărul de secunde după care broscuța își înjumătățește saltul,  $n$ , precum și durata totală a deplasării broscuței  $T$  (exprimată în secunde) și care să determine distanța totală pe care a parcurs-o broscuța. Distanța totală determinată va fi afișată pe ecran cu două zecimale. Restricții:  $p, n, T$  sînt numere naturale;  $p, n, T < 30000; T/n < 16$  Exemplu Pentru  $n=10, p=20$  și  $T=33$ , distanța totală pe care se deplasează broscuța este 357.50 cm.
5. O bază de date conține un tabel cu cîteva cîmpuri. Se cere să se transforme acest tabel într-un fișier XML sau fișier txt. Numele bazei de date și cel al fișierelor de ieșire se va indica prin interfață.
6. Este nevoie de a transforma o suma de bani dintr-o valută în alta. În acest scop se va utiliza cursul de cumparare și cel de vînzare pentru valutele corespunzătoare. Se lucrează cu valutele: leul moldovenesc, dolarul american, euro, leul românesc, rubla rusă și hrivna ucraineană.
7. Trebuie de calculat suma de bani care va fi după ce a fost depusă la o bancă la un depozit, în valută națională, dolar american, euro. Depozitul permite capitalizare, uneori sînt condiții de retragere și completare. Ținînd cont de aceste date să se calculeze sumele la finele termenului.
8. Trebuie de calculat suma care este necesara pentru achitarea creditului bancar cu toate taxele aferente. Se va lua în calcul toate particularitățile și mai multe tipuri de credite.

## ***Resursele informaționale la disciplină***

### **A. Literatura de bază**

1. Frășinaru Cr., Curs practic de Java, Iași, 462 p.
2. Sorin T., Huțanu V., Bazele programării în Java. Editura L&S InfoMat, 2005, 296 p.
3. Флэнеган Д., Java в примерах. Справочник. Учебное пособие к книге «Java. Справочник» Второе издание, Символ+, Москва, 2003, 664 стр.

### **B. Literatura suplimentară**

4. Файн Як., Программирование на Java для детей, родителей, бабушек и дедушек, Москва, 231 стр.
5. Машнин Т. Современные Java технологии на практике, «БХВ-Петербург», Санкт-Петербург, 2010, 560 стр. ISBN 978-5-9775-0561-1.
6. Lemay L., Cadenhead R., Java 2 fără profesor. Editura Teora, București., 2001, 552 p.

### **C. Resurse Internet**

7. Situl oficial IDE IntelliJIDEA [online] data 05.05.2015 Disponibil pe internet < [www.jetbrains.com](http://www.jetbrains.com) >
8. Situl oficial IDE Eclipse [online] data 05.05.2015 Disponibil pe internet < [www.eclipse.org](http://www.eclipse.org) >
9. Situl oficial IDE NetBeans [online] data 05.05.2015 Disponibil pe internet < [www.netbeans.org](http://www.netbeans.org) >
10. Documentația oficială Java [online] data 05.05.2015 Disponibil pe internet < [www.docs.oracle.com](http://www.docs.oracle.com) >

### **D. Tutoriale video Java**

11. Tutorial video limbajul Java în limba română [online] data 05.05.2015 Disponibil pe internet < [www.tutorialevideo.info/categories/java](http://www.tutorialevideo.info/categories/java) >

12. Tutorial video limbajul Java în limba engleză [online] data 05.05.2015  
Disponibil pe internet < [www.tutorials.jenkov.com/java-nio](http://www.tutorials.jenkov.com/java-nio) >
13. Tutorial video limbajul Java în limba engleză [online] data 05.05.2015  
Disponibil pe internet < [www.java2s.com/Tutorial/Java](http://www.java2s.com/Tutorial/Java) >
14. Tutorial video limbajul Java în limba rusă [online] data 05.05.2015 Disponibil  
pe internet < [www.youtube.com/user/JavaOOP/](http://www.youtube.com/user/JavaOOP/) >

## Evaluare

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 15-a (**LC2**)

1.3. *Lucrare de control scrisă nr. 3*: perechea a 22-a (**LC3**)

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. Teza anuală (**TA**)

4. La examenul final, conform orarului întocmit de decanat (**Ex**).

Nota finală la disciplina „Programare Java” se calculează conform formulelor:

$$N_{\text{evaluarea\_curentă}} = 1/3 \times \text{PR} + 1/3 \times \text{LLab} + 1/3 \times \text{TA}$$

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

Unde  $\text{PR} = (\text{LC1} + \text{LC2} + \text{LC3})/3$  și  $\text{LLab} = (\text{SI1} + \text{SI2} + \text{Pro})/3$

Examenul final se susține scris, care va include un test complex cu diferite tipuri de itemi.

## 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 salutată 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” .



**Exemplu de item pentru testul de la examen**  
la disciplina „Programarea Java. Teza anuală”

**I. Completează spațiile libere cu noțiunea corectă.**

1. Un suport software care trebuie să fie instalat pe calculatoarele care rulează fișierele \_\_\_\_\_ se numește JVM.
2. Dacă dorim să utilizăm unele metode din una sau mai multe clase atunci se vor elabora \_\_\_\_\_.
3. Dacă o clasă conține date membru și mai multe \_\_\_\_\_ atunci așa clasă se numește clasa abstractă.
4. \_\_\_\_\_ returnează o valoare aleatoare în intervalul [0,1).
5. O clasă care conține \_\_\_\_\_ se numește interfață.
6. Șirurile de caractere reținute de obiectele \_\_\_\_\_ pot fi comparate din punct de vedere lexicografic.
7. O situație nedorită în care poate ajunge un program în timpul rulării se numește \_\_\_\_\_.
8. În Java \_\_\_\_\_ sînt abordate pornind de la noțiunea de flux de date.
9. Clasa StringTokenizer se găsește în pachetul \_\_\_\_\_.
10. O \_\_\_\_\_ este un obiect al clasei JFrame.
11. Structura care reține referințele către \_\_\_\_\_ este un obiect al clasei Container.
12. Gestionarul CardLayout se caracterizează prin faptul că, la un moment dat, \_\_\_\_\_.
13. Componentele JPanel sînt de tip \_\_\_\_\_, adică au rolul de a conține pe suprafața lor alte componente.
14. Obiectele clasei JComboBox sînt așa numitele liste, din care utilizatorul poate \_\_\_\_\_.
15. Obiectele clasei JMenu au rolul de a înscrie în meniu anumite opțiuni, care \_\_\_\_\_.

**II. Explică diferența dintre noțiunile:**

- a) *interfață și clasă abstractă*
- b) *Interfața Set și interfața List*
- c) *Pachetele javax.swing.\* și java.awt.\**

**III. Explică esența noțiunilor de încapsulare, excepție, serializare, gestionarul de poziționare.**

- IV. Scrie un program Java care citește de la tastatura valorile a două numere întregi și tipărește la ecran numărul mai mare din ele.
- V. Să se citească din fișierul *in.txt* două valori întregi a și b care sînt scrise separat pe fiecare linie. Programul se va realiza astfel:
- dacă fișierul *in.txt* să se înregistreze în fișierul *out.txt* un mesaj corespunzător;
  - dacă fișierul nu conține numerele întregi să se înregistreze în fișierul *out.txt* un mesaj corespunzător;
  - să se înregistreze în fișierul *out.txt* rezultatul împărțirii a la b în cazul în care ambele valori sînt nenule;
  - dacă este imposibil de a realiza împărțirea să se înregistreze în fișierul *out.txt* eroarea corespunzătoare.

**Baremul**  
**de convertire a punctajului în note**  
**pentru examenul de evaluare a cunoștințelor la disciplina**  
**„Programarea Java. Teza anuală”**  
(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

## Note