

FIȘA DISCIPLINEI

Denumirea disciplinei :		Analiza și proiectarea algoritmilor			
Codul disciplinei:		390453080612FO20			
Domeniul:		Calculatoare și Tehnologia Informației			
Specializarea:		Tehnologia Informației			
Catedra:		Calculatoare și Automatizări			
Facultatea:		Inginerie			
Universitatea:		„Lucian Blaga” din Sibiu			
Anul de studiu:	2	Semestrul	3	Tipul de evaluare finală	E
Regimul disciplinei (DI=obligatorie/ DO=opțională/DF=liber aleasă):			DI	Numărul de credite:	5
Categorica formativă a disciplinei (DF=fundamentală.; DI=ingineresci; DS=specialitate; DC=complementară)					DF
Total ore din planul de învățământ	70			Total ore pe semestru:	70
Titularul disciplinei: Dr. Arpad GELLERT					

Numărul total de ore (pe semestru) din planul de învățământ					
Total ore/ semestru	C	S	L	P	Total
	3	-	2	-	5

Obiective:	<ul style="list-style-type: none"> • Dobândirea cunoștințelor necesare înțelegerii, analizei, adaptării și implementării de algoritmi; • Însușirea cunoștințelor necesare identificării algoritmului adecvat pentru o anumită problemă; • Familiarizarea cu limbajul de programare Java.
Competențe specifice disciplinei	<p>1. Cunoaștere și înțelegere:</p> <ul style="list-style-type: none"> • Însușirea principalelor concepte relative la analiza și proiectarea algoritmilor; • Înțelegerea aprofundată a metodelor de analiză a complexității algoritmilor; • Cunoașterea și înțelegerea principalelor tehnici de programare; • Familiarizarea cu limbajul Java respectiv programarea orientată obiect. <p>2. Explicare și interpretare:</p> <ul style="list-style-type: none"> • Dezvoltarea unor deprinderi de analiză și proiectare a algoritmilor; • Explicarea conceptelor de programare orientată obiect; • Abordarea conceptelor specifice limbajului Java. <p>3. Instrumental – aplicative</p> <ul style="list-style-type: none"> • Exersarea principalelor tehnici de analiză și proiectare a algoritmilor; • Utilizarea mediilor de dezvoltare Java; • Implementarea și depanarea programelor Java. <p>4. Atitudinale:</p> <ul style="list-style-type: none"> • Capacitatea de implementare corectă și eficientă a algoritmilor; • Cunoașterea principalelor tehnici de analiză și proiectare a algoritmilor; • Capacitatea studenților de a alege algoritmul adecvat pentru o anumită problemă.

Conținutul tematic (descriptori)	TEMATICA CURSURILOR		
	Nr. crt.	Denumirea temei	Nr. ore
	1.	Structura lexicală a limbajului Java. Tipuri de date primitive și referință. Conversii. Operatori și instrucțiuni. Variabile și constante. Tablouri și matrici	3
	2.	Tratarea excepțiilor Operații I/O	3
	3.	Crearea claselor. Constructori. Moștenirea Interfețe: utilizarea interfetelor Comparator, Comparable și Cloneable	3
	4.	Interfețe grafice. Layout manager Fire de execuție	3
	5.	Colecții de date Java: Vector, ArrayList, LinkedList, HashSet, TreeSet, LinkedHashSet, HashMap, TreeMap, LinkedHashMap	3
	6.	Introducere în problematica complexității algoritmilor Notații asimptotice Recurențe	3
	7.	Algoritmi de căutare: căutare secvențială, căutare binară Algoritmi de sortare: sortare prin inserție, sortare prin selecție, Bubblesort, Quicksort, Mergesort	3
	8.	Tabele de dispersie: funcții de dispersie, dispersie cu înlănțuire, adresare deschisă	3
	9.	Arbori binari: inserarea unui nod, ștergerea unui nod, traversarea în inordine, preordine și postordine, căutarea recursivă și iterativă	3
	10.	Heap-uri: inserarea unui nod, reconstituire heap, extragerea rădăcinii, construire heap, algoritmul Heapsort	3
	11.	Grafuri: parcurgerea în lățime, parcurgerea în adâncime, algoritmul Dijkstra Căutare în spațiul stărilor: algoritmul A*	3
	12.	Divide et impera: problema turnurilor din Hanoi, etc. Greedy: problema fracționară a rucsacului, coduri Huffman, etc.	3
	13.	Programare dinamică: cel mai lung subșir crescător, cel mai scurt drum al unui cal între două poziții pe tabla de șah, etc. Backtracking: problema celor n regine, problema turelor, etc.	3
	14.	Algoritmi genetici: determinarea maximului unei funcții. Rețele neuronale: algoritmul de învățare Backpropagation	3
	TEMATICA LABORATOARELOR		
	1.	Compilarea și rularea programelor Java: aplicația „Hello World”. Tablouri: căutarea minimului și a maximului într-un tablou, generarea și afișarea unui tablou cu numerele lui Fibonacci, sortare folosind clasa Arrays. Matrici: adunarea și înmulțirea a două matrici.	2
	2.	Tratarea excepțiilor. Operații I/O: afișarea la consolă, citirea de la tastatură, citirea din fișier, scrierea în fișier.	2
	3.	Crearea claselor, constructori, moștenirea: implementarea unor aplicații care implică definirea unor clase și derivarea acestora. Interfețe: implementarea unor aplicații care implică definirea unor interfete și utilizarea acestora, respectiv, folosirea interfetelor Comparator, Comparable și Cloneable.	2

	4.	Interfețe grafice: proiectarea unei interfețe grafice pentru introducerea studenților în liste, tabele, etc.	2
	5.	Fire de execuție. Colecții de date Java: Vector, ArrayList, LinkedList, HashSet, TreeSet, LinkedHashSet, HashMap, TreeMap, LinkedHashMap.	2
	6.	Algoritmi de căutare: căutare secvențială, căutare binară. Algoritmi de sortare: sortare prin inserție, sortare prin selecție, Bubblesort, Quicksort, Mergesort.	2
	7.	Tabele de dispersie: funcții de dispersie, dispersie cu înlănțuire, adresare deschisă. Arbori binari: inserarea unui nod, ștergerea unui nod, traversarea în inordine, preordine și postordine, căutarea recursivă și iterativă.	2
	8.	Heap-uri: inserarea unui nod, reconstituire heap, extragerea rădăcinii, construire heap, algoritmul Heapsort.	2
	9.	Grafuri: parcurgerea în lățime, parcurgerea în adâncime, algoritmul Dijkstra. Căutare în spațiul stărilor: algoritmul A*	2
	10.	Divide et impera: problema turnurilor din Hanoi, etc.	2
	11.	Greedy: problema fracționară a rucsacului, coduri Huffman	2
	12.	Programare dinamică: cel mai lung subșir crescător, cel mai scurt drum al unui cal între două poziții pe tabla de șah, etc. Backtracking: problema celor n regine, problema turelor, etc.	2
	13.	Algoritmi genetici: determinarea maximului unei funcții.	2
	14.	Rețele neuronale: implementarea unei rețele neuronale cu un strat ascuns care, pe baza algoritmului Backpropagation, să învețe cifrele.	2

Metode de predare / seminarizare	Expunerea, conversația, problematizarea, studii de caz, teme de casă.
----------------------------------	---

Stabilirea notei finale (procentaje)	- răspunsurile la examen/colocviu (evaluare finală)	50%
	- teste pe parcursul semestrului	20%
	- răspunsurile la colocviul de laborator	25%
	- teme	5%
	- TOTAL	100%

Evaluarea finală va fi realizată printr-un examen scris care va consta într-un subiect descriptiv respectiv rezolvarea a 5-6 probleme cu un pronunțat caracter aplicativ (probleme).

Cerințe minime pentru nota 5

- Capacitatea de interpretare și înțelegere a algoritmilor (descriși în pseudocod sau implementați în Java).
- Abilitatea de proiectare a algoritmilor pentru rezolvarea diferitelor probleme.
- Stăpânirea conceptelor de bază specifice limbajului Java (programare orientată obiect, colecții de date).

Cerințe pentru nota 10

- Capacitatea de analiză a algoritmilor din punct de vedere al complexității.
- Capacitatea de selecție a algoritmului potrivit pentru rezolvarea unei anumite probleme.
- Abilitatea de adaptare a unui algoritm general pentru rezolvarea unei anumite probleme.

TOTAL ore studiu individual (pe semestru) = 98

Bibliografia	<p>Minimală obligatorie: Arpad GELLERT – Analiza și proiectarea algoritmilor, http://webspaces.ulbsibiu.ro/arpad.gellert/html/Algoritmi.pdf</p> <p>Complementară:</p> <ol style="list-style-type: none"> 1. Knuth D., <i>Arta programării calculatoarelor</i>, Vol. 1 – <i>Algoritmi fundamentali</i>, Teora, 2000. 2. Knuth D., <i>Arta programării calculatoarelor</i>, Vol. 3 – <i>Sortare și căutare</i>, Teora, 2002. 3. Cormen T., Leiserson C., Rivest R., <i>Introducere în algoritmi</i>, Agora, 2000. 4. Giumale C., <i>Introducere în analiza algoritmilor</i>, Polirom, 2004. 5. Waite M., Lafore R., <i>Structuri de date și algoritmi în Java</i>, Teora, 2001. 6. Logofătu D., <i>Algoritmi fundamentali în Java</i>, Polirom, 2007. 7. Cristea V., Athanasiu I., Kalisz E., Iorga V., <i>Tehnici de programare</i>, Teora 1998. 8. Mitchell T., <i>Machine Learning</i>, McGraw-Hill, 1997. 9. Roberts S., Heller P., Ernest M., <i>Complete Java 2 Certification</i>, Second Edition, SYBEX, USA, 2000. 10. Chan M., Griffith S., Iasi A., <i>Java 1001 secrete pentru programatori</i>, Teora, 2000. 11. Tanasă Ș., Andrei Ș., Olaru C., <i>Java de la 0 la expert</i>, Polirom, 2007. 12. Gordon R., <i>Java Native Interface</i>, Prentice Hall, USA, 1998.
	<p>Lista materialelor didactice utilizate în procesul de predare: Cursul, disponibil la adresa http://webspaces.ulbsibiu.ro/arpad.gellert/html/Algoritmi.pdf, este prezentat cu videoproiectorul. Aplicațiile prezentate și problemele propuse se implementează la laborator în Java Builder 9.</p>

Coordonator de Disciplină	Grad didactic, titlul, prenume, numele	Semnătura
	Dr. Arpad GELLERT	