



Academia de Studii Economice
Departamentul de Informatică și Cibernetică Economică

Calea Dorobanți, 15-17, Sector 1, București, 010552 (camera 2314)

Tel.: +40 21 319 19 00, ext. 319, 336, Fax: +40 21 311 20 66

www.dice.ase.ro

**Tematica de concurs pentru postul de Conferențiar universitar,
poziția 62, an univ. 2023-2024, semestrul I,**

Disciplinele: Calitate și testare software; Procesare paralelă; Programare orientată obiect; Securitatea codului sursă în programare - în limba engleză.

Calitate și testare software

1. Principii privind scrierea codului sursă - Clean Code (SOLID, DRY, KISS, YAGNI, WET, convenții de nume, definirea și implementarea funcțiilor)
2. Design Patterns: Singleton, Simple Factory, Factory Method, Abstract Factory, Builder, Adapter, Decorator, Facade, Flyweight, Chain of Responsibility, Command, Observer, State, Strategy, Memento
3. Gestiunea versiunilor codului sursă - Git, SVN
4. Concepte privind testarea unitară (Unit Testing)
5. Utilizarea framework-ului JUnit (versiunile 3, 4 și 5)
6. Concepte privind calitatea software - metrici, indicatori, instrumente, testare automată (testare aplicații Web cu platforma Selenium)

Bibliografie:

1. I. Ivan, C. Boja, S. Capisizu, M. Popa, *Managementul calității aplicațiilor informatice*, ASE, 2006, România
2. I. Sommerville, *Software Engineering. 9th ed.*, Addison-Wesley, 2011, Statele Unite ale Americii
3. S. Pressman, *Software Engineering: A Practitioner's Approach. 7th ed.*, McGraw-Hill, 2009, Statele Unite ale Americii
4. P. Pocatilu, *Costurile testării software*, ASE, 2004, România
5. S. McConnell, *Code complete, 2nd ed.*, Microsoft Press, 2004, Statele Unite ale Americii
6. Scott Chacon, Ben Straub, *Pro Git, 2nd edition*, Apress, 2014, <http://git-scm.com/book/en/v2>, Statele Unite ale Americii
7. Robert C. Martin, *Clean Code, A Handbook of Agile Software Craftsmanship*, Prentice Hall, 2009, Statele Unite ale Americii
8. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995, https://books.google.ro/books?id=6oHuKQe3TjQC&printsec=frontcover&source=gbs_ge_summary_r&hl=ro#v=onepage&q&f=false, Statele Unite ale Americii
9. Andy Hunt, Dave Thomas, *Pragmatic Unit Testing in Java with JUnit*, The Pragmatic Programmers, 2004, Statele Unite ale Americii



Academia de Studii Economice
Departamentul de Informatică și Cibernetică Economică

Calea Dorobanți, 15-17, Sector 1, București, 010552 (camera 2314)

Tel.: +40 21 319 19 00, ext. 319, 336, Fax: +40 21 311 20 66

www.dice.ase.ro

10. Mădălina Zurini, Alin Zamfiroiu, *Calitate și testare software. Studii de caz*, ASE, București, 2017, România

Procesare paralelă:

1. Concepte specifice Procesării paralele. Comparatie cu programarea secvențială, asincronă și concurentă.
2. Avantajele și dezavantajele procesării paralele. Legea lui Amdahl.
3. Procesare paralelă pe mai multe fire de execuție în Java.
4. Conceptul de fir de execuție.
5. Crearea firelor de execuție în Java. Runnable vs Thread vs Callable
6. Orchestrarea firelor de execuție în Java prin utilizarea semafoarelor și a barierelor
7. Probleme ale procesării paralele: race condition, deadlock, livelock, cache coherence, cache false sharing. Exemplificare în Java
8. Atomic, Volatile și synchronized in Java.
9. Avantajele și dezavantajele orchestrării firelor de execuție prin synchronized.
10. Tipuri de colecții Java în contextul procesării paralele prin mai multe fire de execuție
11. Managementul firelor de execuție în Java prin ExecutorService
12. Java Parallel Streams
13. Algoritmi specifici procesării paralele. Exemplificare pentru sortare și căutare de subșiruri de caractere

Bibliografie:

1. Oracle, JDK 11 Documentation, <https://docs.oracle.com/en/java/javase/11/>
2. Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, Doug Lea , *Java Concurrency in Practice*, Addison-Wesley Professional; 1st edition (May 9, 2006), ISBN, 978-0321349606
3. Jeff Friesen , *Concurrency in Java*, Apress; 1st ed. edition (November 23, 2015), ISBN 978-1484216996
4. Javier Fernández González, *Mastering Concurrency Programming with Java 9*, Packt Publishing; 2nd Revised edition (July 17, 2017), ISBN 978-1785887949
5. David E. Culler, Anoop Gupta, Jaswinder Pal Singh, *Parallel Computer Architecture: A Hardware/Software Approach*, Morgan Kaufmann Publishers, 1997, Statele Unite ale Americii
6. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, *Introduction to Parallel Computing, Second Edition*, Addison Wesley, 2003, <https://www-users.cs.umn.edu/~karypis/parbook/>, Statele Unite ale Americii



Academia de Studii Economice
Departamentul de Informatică și Cibernetică Economică

Calea Dorobanți, 15-17, Sector 1, București, 010552 (camera 2314)

Tel.: +40 21 319 19 00, ext. 319, 336, Fax: +40 21 311 20 66

www.dice.ase.ro

7. Blaise Barney, *Introduction to Parallel Computing*, 2018, https://computing.llnl.gov/tutorials/parallel_comp/, Statele Unite ale Americii
8. Gheorghe Dodescu, Bogdan Oancea, Mădălina Răceanu, *Procesare paralelă*, Ed. Economica, 2002, România
9. Cosmina Ivan, *Tehnologii și Aplicații în Calculul Paralel și distribuit*, UTPRESS, 2019, <https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/390-5.pdf>, România

Programare orientată obiect:

1. Elemente ale programării procedurale: funcții, transferul parametrilor, pointeri la date și funcții, clase de memorie.
2. Conceptele de clasă, obiect, constructor, destructor, metode de acces, pointerul *this*. Declararea și implementarea metodelor în clasă și în afara clasei.
3. Asimilarea conceptelor de constructor de copiere, supraîncărcare operator =, obiecte cu extensii în memoria dinamică și domenii de nume (namespace).
4. Conversii între diferite tipuri de obiecte (operatorul cast, operatorul = și constructor de copiere), vector de obiecte, modificatorul const, tipologia membrilor statici (static), obiecte constante, pointeri constanți la obiecte și pointeri la obiecte constante.
5. Mecanismul try-catch în C++.
6. Supraîncărcarea operatorilor.
7. Clase derivate, moștenire. Polimorfism.
8. Funcții virtuale, supradefinire, moștenire multiplă.
9. Mecanisme de tip RTTI, moștenire multiplă și dynamic cast.
10. Funcții și clase template.
11. Operații I/O orientate pe stream-uri. Conceptele de serializare/deserializare obiecte.
12. Standard Template Library - STL – containere, iteratori și algoritmi. Clasa string, map, list, vector, etc.

Bibliografie:

1. Ion Smeureanu, Marian Dârdală, *Programarea orientată obiect în limbajul C++*, Editura CISON, 2002, România;
2. Ion Smeureanu, Marian Dârdală, *Programarea în limbajul C/C++*, Editura CISON, 2001, România
3. Bjarne Stroustrup, *The C++ Programming Language, 3rd Edition*, Addison-Wesley, <http://www.research.att.com/~bs/3rd.html>, Statele Unite ale Americii
4. C++ Standard, ISO, <https://isocpp.org/std/the-standard>, Statele Unite ale Americii

Securitatea codului sursă în programare - în limba engleză

1. Securitate de nivel jos;
2. Vulnerabilități ale șirurilor de caractere;



Academia de Studii Economice
Departamentul de Informatică și Cibernetică Economică

Calea Dorobanți, 15-17, Sector 1, București, 010552 (camera 2314)

Tel.: +40 21 319 19 00, ext. 319, 336, Fax: +40 21 311 20 66

www.dice.ase.ro

3. Race Conditions;
4. Vulnerabilități de tip Reverse Engineering;
5. Vulnerabilități ale limbajelor orientate obiect;
6. Vulnerabilități ale limbajelor web,
7. Vulnerabilități ale limbajelor bazate pe mașina virtuală;

Bibliografie:

1. David LeBlanc, Michael Howard, *Writing Secure Code*, Pearson Education, 2002
2. John Viega, Gary R, McGraw, *Building Secure Software: How to Avoid Security Problems The Right Way*, Pearson Education, 2001
3. University of Maryland, *Software Security Course*,
<https://www.coursera.org/learn/software-security>
4. - University of Virginia, *Defense Against the Dark Arts*,
<http://www.cs.virginia.edu/~ww6r/CS4630/syllabus.html>