



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 Asistent universitar,
poziția 153, an univ. 2023-2024, semestrul al II-lea**

Discipline: Calitate și testare software; Calitate și testare software (în limba Engleză); Programare Orientată Obiect; Programare Orientată Obiect (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 sursa - Git, SVN
4. Concepte privind testarea unitară (Unit Testing)
5. Utilizarea framework-ului JUnit (versiunile 3, 4 și 5)
6. Concepte privind calitatea software - metrice, indicatori, instrumente, testare automata (testare aplicații Web cu platforma Selenium)

Bibliografie:

1. I. Ivan, C. Boja, S. Capisizu, M. Popa, *Managementul calității aplicațiilor informatice*, Editura 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*, Editura 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,
9. 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



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. Andy Hunt, Dave Thomas, *Pragmatic Unit Testing in Java with JUnit, The Pragmatic Programmers*, 2004, Statele Unite ale Americii
11. Mădălina Zurini, Alin Zamfiroiu, *Calitate și testare software. Studii de caz*, Editura ASE, București, 2017, România

Calitate și testare software (în limba Engleză):

1. Principles of writing source code - Clean Code (SOLID, DRY, KISS, YAGNI, WET, naming conventions, defining and implementing functions)
2. Design Patterns: Singleton, Simple Factory, Factory Method, Abstract Factory, Builder, Adapter, Decorator, Facade, Flyweight, Chain of Responsibility, Command, Observer, State, Strategy, Reminder
3. Source code version management - Git, SVN
4. Unit Testing Concepts
5. Using the JUnit framework (versions 3, 4 and 5)
6. Software quality concepts - metrics, indicators, tools, automated testing (web application testing with the Selenium platform)

Bibliography:

1. I. Ivan, C. Boja, S. Capisizu, M. Popa, *Managementul calității aplicațiilor informatice*, Editura 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*, Editura 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,
9. 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
10. 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

11. Mădălina Zurini, Alin Zamfiroiu, *Calitate și testare software. Studii de caz*, Editura ASE, București, 2017, 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. Herbert Schildt, *C++ manual complet*, Editura Teora, România
5. C++ Standard, ISO, <https://isocpp.org/std/the-standard>, Statele Unite ale Americii

Programare Orientată Obiect (în limba Engleză):

1. Elements of procedural programming: functions, parameter transfer, pointers to data and functions, memory classes.



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

2. The concepts of class, object, constructor, destructor, access methods, the this pointer. Declaring and implementing methods in the classroom and outside the classroom.
3. Assimilating the concepts of copy constructor, = operator overloading, objects with extensions in dynamic memory and name domains (namespace).
4. Conversions between different types of objects (cast operator, = operator and copy constructor), vector of objects, const modifier, typology of static members (static), constant objects, constant pointers to objects and pointers to constant objects.
5. The try-catch mechanism in C++.
6. Overloading operators.
7. Derived classes, inheritance. Polymorphism.
8. Virtual functions, overdefinition, multiple inheritance.
9. RTTI mechanisms, multiple inheritance and dynamic cast.
10. Template functions and classes.
11. Stream-oriented I/O operations. Object serialization/deserialization concepts.
12. Standard Template Library - STL – containers, iterators and algorithms. Class string, map, list, vector, etc.

Bibliography:

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. Herbert Schildt, C++ manual complet, Editura Teora, România
5. C++ Standard, ISO, <https://isocpp.org/std/the-standard>, Statele Unite ale Americii