

FIȘA DISCIPLINEI

1. Date despre program

| | |
|---------------------------------------|---|
| 1.1 Instituția de învățământ superior | Universitatea Ovidius din Constanța |
| 1.2 Facultatea | Matematică și Informatică |
| 1.3 Departamentul | Matematică și Informatică |
| 1.4 Domeniul de studii: | Informatică |
| 1.5 Ciclul de studii: | Master |
| 1.6 Programul de studii: | Medii virtuale multi-modale distribuite |

2. Date despre disciplină

| | | | | | | |
|--|---------------------------------------|---------------|---|------------------------------------|---|--|
| 2.1 Denumirea disciplinei | Modele si framework-uri de proiectare | | | | | |
| 2.2. Codul disciplinei | FML.MVMMD.I.2.07 | | | | | |
| 2.2 Titularul activitatilor de curs | Conf.univ.dr. Puchianu Crenguța | | | | | |
| 2.3 Titularul activitatilor de seminar | Conf.univ.dr. Puchianu Crenguța | | | | | |
| 2.5 Anul de studiu | 1 | 2.6 Semestrul | 2 | 2.7 Tipul de evaluare ¹ | E | 2.8 Regimul disciplinei ^{2,3} DC / DO |

3. Timpul total estimat (ore pe semestru al activităților didactice)

| | | | | | |
|--|----|--------------------|----|-----------------------|--------------|
| 3.1 Numarul de ore pe saptamana | 4 | din care: 3.2 curs | 2 | 3.3 seminar/laborator | 2 |
| 3.4 Total ore din planul de invatamant | 56 | din care: 3.5 curs | 28 | 3.6 seminar/laborator | 28 |
| <i>Distributia fondului de timp</i> | | | | | <i>[ore]</i> |
| Studiul dupa manual, suport de curs, bibliografie si notite | | | | | 30 |
| Documentare suplimentara in biblioteca, pe platforme electronice de specialitate si pe teren | | | | | 30 |
| Pregatire seminarii/laboratoare, teme, referate, portofolii si eseuri | | | | | 20 |
| Tutoriat | | | | | 24 |
| Examinari | | | | | 15 |
| Alte activitati | | | | | 0 |
| 3.7 Total ore studiu individual | | | | | 119 |
| 3.8 Total ore pe semestru | | | | | 175 |
| 3.9 Numarul de credite | | | | | 7 |

4. Precondiții (acolo unde este cazul)

| | |
|-------------------|---|
| 4.1 de curriculum | Proiectarea orientata spre obiecte, Dezvoltarea aplicatiilor software |
| 4.2 de competente | C2. Dezvoltarea si intretinerea sistemelor software |

5. Condiții (acolo unde este cazul)

| | |
|--|---------------------------------------|
| 5.1 de desfășurare a cursului | Sala de curs disponibila |
| 5.2 de desfășurare a seminarului/laboratorului | Sala de seminar/laborator disponibila |

6. Competențele specifice acumulate

| | |
|-------------------------|---|
| Competențe profesionale | Utilizarea modelelor de proiectare adecvate in proiectarea arhitecturii unui sistem software. Identificarea conceptelor, modelelor si tehnicilor utilizate in framework-urile de proiectare. Dezvoltarea unui proiect și elaborarea documentației aferente. |
| Competențe transversale | Executarea unor sarcini profesionale complexe, în condiții de autonomie și de independență profesională, implicând detectarea și rezolvarea problemelor conexe apărute în proiectarea arhitecturilor orientate spre obiecte folosind modele de proiectare sau un framework de proiectare. Desfășurarea eficientă a activităților organizate într-un grup inter-disciplinar și dezvoltarea capacităților empatice de comunicare inter-personală, de relaționare și colaborare cu grupuri diverse. |

7. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

| | |
|---------------------------------------|---|
| 7.1 Obiectivul general al disciplinei | Familiarizarea cu modelele de proiectare pentru aplicarea lor in dezvoltarea sistemelor software. |
|---------------------------------------|---|

¹ Forma de evaluare: E - examen, C - colocviu

² Categoria formativa a disciplinei: DF - fundamentala; DS - de specialitate; DC - complementara

³ Categoria de optionalitate a disciplinei: DI -impusa; DO - optionala;DF - facultativa

| | |
|---------------------------|--|
| 7.2 Obiectivele specifice | <p>Identificarea problemelor de proiectare ce pot fi rezolvate cu modele de proiectare.</p> <p>Proiectarea orientată spre obiect a sistemelor software folosind modele de proiectare.</p> <p>Aplicarea corectă a modelelor.</p> <p>Folosirea framework-ului de proiectare Spring</p> |
|---------------------------|--|

8. Conținuturi

| 8.1 Curs | Metode de învățământ | Mijloace de învățământ |
|--|--|------------------------|
| <ol style="list-style-type: none"> 1. <i>Introducere</i>. Modele de proiectare. Clasificare 2. <i>Modele de proiectare fundamentale</i>: Delegation, Interface, Marker Interface, Proxy 3. <i>Modele de creare</i>: Factory Method, Abstract Factory, Builder, Prototype, Singleton, Object Pool. 4. <i>Modele de impartire</i>: Layered Initialization, Filter, Composite. 5. <i>Modele structurale</i>: Adapter, Iterator, Bridge, Façade, Flyweight, Dynamic Linkage, Virtual Proxy, Decorator, Cache Management. 6. <i>Modele comportamentale</i>: Command, Mediator, Observer, State, Null Object, Strategy, Template Method, Visitor. 7. <i>Modele de concurență</i>: Read/Write Lock, Lock Object, Producer-Consumer, Two-Phase Termination, Future, Single Thread Execution 8. <i>Framework-uri orientate spre obiecte</i>. Definiții, clasificări, avantaje și dezavantajele folosirii framework-urilor. Principii de construire a framework-urilor. 9. <i>Proiectarea sistemelor software folosind framework-ul Tapestry</i>. | <p>Prelegere cu sintetizarea și esențializarea informațiilor</p> <p>Metode de predare-învățare interactive</p> <p>Dialogul</p> <p>Problematizarea</p> <p>Conversația</p> <p>Metode care contribuie la dezvoltarea gândirii critice</p> <p>Interacțiunea, problematizarea, argumentarea</p> | |
| <ol style="list-style-type: none"> 1. Bibliografie 2. M. Grand, Patterns in Java, A Catalog of Reusable Design Patterns Illustrated with UML, vol. 1 și 2, 2/e, Editura John Wiley & Sons, 2002 3. C. Larman, <i>Applying UML and Patterns - An Introduction to Object-Oriented Analysis and Design and the Unified Process</i>, 2/e, Prentice Hall, 2002. 4. Gamma, Erich; Richard Helm, Ralph Johnson, and John Vlissides (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. 5. Eric Freeman, Elisabeth Freeman, Bert Bates, Kathy Sierra, Head First Design Patterns, O'Reilly Media, 2004. 6. Igor Drobiazko, <i>Tapestry 5: Rapid web application development in Java</i>, 2012 | | |
| 8.2 Seminar/laborator | Metode de învățământ | Mijloace de învățământ |
| <ol style="list-style-type: none"> 1. <i>Modele de proiectare fundamentale</i>: Delegation, Interface, Marker Interface, Proxy 2. <i>Modele de creare</i>: Factory Method, Abstract Factory, Builder, Prototype, Singleton, Object Pool. 3. <i>Modele de impartire</i>: Layered Initialization, Filter, Composite. 4. <i>Modele structurale</i>: Adapter, Iterator, Bridge, Façade, Flyweight, Dynamic Linkage, Virtual Proxy, Decorator, Cache Management. 5. <i>Modele comportamentale</i>: Command, Mediator, Observer, State, Null Object, Strategy, Template Method, Visitor. 6. <i>Modele de concurență</i>: Read/Write Lock, Lock Object, Producer-Consumer, Two-Phase Termination, Future, Single Thread Execution 7. <i>Framework-uri orientate spre obiecte</i>. Definiții, clasificări, avantaje și dezavantajele folosirii framework-urilor. Principii de construire a framework-urilor. 8. <i>Proiectarea sistemelor software folosind framework-ul Tapestry</i>. | <p>Dialogul</p> <p>Problematizarea</p> <p>Conversația</p> <p>Algoritmizare</p> <p>Programare</p> | |

Bibliografie

1. M. Grand, *Patterns in Java, A Catalog of Reusable Design Patterns Illustrated with UML*, vol. 1 si 2, 2/e, Editura John Wiley & Sons, 2002
2. C. Larman, *Applying UML and Patterns - An Introduction to Object-Oriented Analysis and Design and the Unified Process*, 2/e, Prentice Hall, 2002.
3. M. Fontoura, W. Pree, B. Rumpe, *UML Profile for Framework Architectures*, Addison Wesley, 2001
4. Gamma, Erich; Richard Helm, Ralph Johnson, and John Vlissides (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
5. Eric Freeman, Elisabeth Freeman, Bert Bates, Kathy Sierra, *Head First Design Patterns*, O'Reilly Media, 2004.
6. Igor Drobiazko, *Tapestry 5: Rapid web application development in Java*, 2012

9. Coroborarea continuturilor disciplinei cu asteptarile reprezentantilor comunitatii epistemice, asociatiilor profesionale si angajatorilor reprezentativi din domeniul aferent programului

Pregătirea masteranzilor pentru a profesa în domeniul dezvoltării de sisteme software pentru a face față exigențelor firmelor IT sau unor programe de cercetare sau doctorat în știința calculatoarelor.

10. Evaluare

| Tip de activitate | 10.1 Criterii de evaluare | 10.2 Metode de evaluare | 10.3 Pondere din nota finala |
|---|--|-------------------------|------------------------------|
| 10.4 Curs | Participare activa la activitati didactice | Oral | 10% |
| 10.5 Seminar/ laborator | Participare activa | Oral | 10% |
| | Proiect | Oral | 40% |
| | Examen | Oral | 40% |
| 10.6 Standard minim de performanta | | | |
| Aplicarea modelelor de proiectare fundamentale și de creare a obiectelor în proiectarea unei arhitecturi software orientată spre obiecte. | | | |

Data completării

Semnătura titularului de curs

Semnătura titularului de seminar

.....

.....

.....

Data avizării în Departament

Semnătura Directorului de Departament

.....

.....