

FIȘA DISCIPLINEI

Medii de proiectare și programare

Anul universitar

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai
1.2. Facultatea	de Matematică și Informatică
1.3. Departamentul	de Matematică și Informatică al Liniei Maghiare
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Licență
1.6. Programul de studii / Calificarea	Informatică
1.7. Forma de învățământ	cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei		Medii de proiectare și programare				Codul disciplinei		MLM5013			
2.2. Titularul activităților de curs			Lect. dr. SULYOK Csaba								
2.3. Titularul activităților de seminar			Lect. dr. SULYOK Csaba								
2.4. Anul de studiu		2	2.5. Semestrul		4	2.6. Tipul de evaluare		E	2.7. Regimul disciplinei		DS

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

3.1. Număr de ore pe săptămână	5	din care: 3.2. curs	2	3.3. seminar/ laborator/proiect	3
3.4. Total ore din planul de învățământ	70	din care: 3.5. curs	28	3.6 seminar/laborator/proiect	42
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					10
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					10
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					31
Tutoriat (consiliere profesională)					-
Examinări					4
Alte activități					-
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				55	
3.8. Total ore pe semestru				125	
3.9. Numărul de credite				5	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	- Introducere în programare și algoritmi - Metode avansate de programare - Baze de date
4.2. de competențe	- Cunoștințe de bază despre limbajul Java - Cunoștințe de bază despre programare web

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	- Sală de curs dotat cu videoproiector - Conexiune stabilă de rețea
5.2. de desfășurare a seminarului/ laboratorului	- Sală de laborator echipată cu calculatoare și rețea wireless fiabilă - Sursă de alimentare pentru laptopuri

6.1. Competențele specifice acumulate¹

Competențe profesionale/esențiale	• C1.1 Descrierea adecvată a paradigmelor de programare și a mecanismelor de limbaj specifice, precum și identificarea diferenței dintre aspectele de ordin semantic și sintactic • C1.3 Elaborarea codurilor sursă adecvate și testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificații de proiectare date • C2.1 Identificarea metodelor de dezvoltare adecvate pentru sistemele software • C2.3 Aplicarea metodelor, mecanismelor de specificare și a mediilor de dezvoltare pentru dezvoltarea aplicațiilor IT
Competențe transversale	• CT1 Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională • CT2 Gestionarea eficientă a activităților organizate într-un grup interdisciplinar și dezvoltarea abilității empatică pentru comunicarea interpersonală, relația cu diferite grupuri și cooperarea • CT3 Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională

6.2. Rezultatele învățării

Cunoștințe	Studentul cunoaște: • Caracteristicile modelelor de proiectare enterprise; • Diferența dintre limbajele și modelele de nivel scăzut și de nivel înalt; • Posibilitățile oferite de un framework de tip enterprise.
Aptitudini	Studentul este capabil să: • implementeze un proiect respectând principiile unei arhitecturi multi-layer; • asigure gestionarea corespunzătoare a erorilor; • separe în mod corespunzător codul sursă în funcție de rol; • automatizeze fragmentele de cod repetitive utilizând un framework.
Responsabilități și autonomie	Studentul are capacitatea de a lucra independent pentru: • a înțelege, a concepe și a implementa un proiect de complexitate medie; • a aplica regulile de lucru într-un mod organizat și eficient; • a respecta principiile și standardele de etică profesională; • a aplica unele metode de lucru care sunt utilizate și în industrie.

¹ Se poate opta pentru competențe sau pentru rezultatele învățării, respectiv pentru ambele. În cazul în care se alege o singură variantă, se va șterge tabelul aferent celeilalte opțiuni, iar opțiunea păstrată va fi numerotată cu 6.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	• Aprofundarea și sistematizarea cunoștințelor legate de proiectarea și implementarea sistemelor software complexe • Dobândirea de cunoștințe legate de dezvoltarea sistemelor distribuite • Dezvoltarea arhitecturilor software multilayer, proiectarea arhitecturilor orientate spre servicii
7.2 Obiectivele specifice	• Descrierea modelului MVC, distincție între clienții thin și rich • Înțelegerea modelelor de proiectare și a tehnologiilor avansate legate de persistență: ORM (Object Relational Mapping), JPA (Java Persistence API), specificații, framework-uri relevante. • Descrierea framework-urilor Enterprise (Java EE, Spring) • Descrierea modelelor de proiectare Inversion of Control și Dependency Injection

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Introducere • Framework-uri enterprise • Modele de arhitecturi enterprise	Expuneri, explicații, exemple, studii de caz	
2. Arhitecturi multilayer • Analiză de domeniu • Three-tier architectures • Gestionarea erorilor și logare	Expuneri, explicații, exemple, studii de caz	
3. Arhitecturi multilayer și acces la date • Bazele persistenței: JDBC • Modelul Abstract DAO Factory • Connection pooling	Expuneri, explicații, exemple, studii de caz	
4. Aplicații web bazate pe Java • Containere web Java: Tomcat, Servlet API	Expuneri, explicații, exemple, studii de caz	
5. Modelul Model-View-Controller • Descrierea modelului MVC, distincție între clienții thin și rich • Implementare MVC în Java: JSP, Thymeleaf	Expuneri, explicații, exemple, studii de caz	
6. Componente Java SE • Fișiere de configurare: Properties • Profilare • Internaționalizare (i18n)	Expuneri, explicații, exemple, studii de caz	
7. Reflection • Reflection and Java Reflection API • Probleme de securitate	Expuneri, explicații, exemple, studii de caz	
8. Containere IoC • Clarificarea diferențelor dintre librărie și framework • Modelul de proiectare Dependency Injection • Container Spring IoC: bean, object, factory	Expuneri, explicații, exemple, studii de caz	
9. Cadre Enterprise: Spring Boot • Properties, profiles • Integrare cu sistemul build	Expuneri, explicații, exemple, studii de caz	
10. Dezvoltarea unei strat de afișare cu un cadru enterprise • Spring Web și Spring Web MVC	Expuneri, explicații, exemple, studii de caz	
11. REST API • Introducere în API-uri de tip RESTful • Modelul Mapper/DTO • Validare de date	Expuneri, explicații, exemple, studii de caz	
12. Strat peste nivelul de persistență • Object-Relational Mapping (ORM)	Expuneri, explicații, exemple, studii de caz	

• Java Persistence API (JPA) și implementările sale (Hibernate)		
13. Strat peste nivelul de persistență • Limbajul de interogare Java Persistence Query Language (JPQL) • Generarea/extragerea stratului de acces la date: Spring Data JPA	Expuneri, explicații, exemple, studii de caz	
14. Repetare	Dialog, exemple de subiecte de examen	

Bibliografie:

1. Károly Simon. *Kenyerünk Java: a Java programozás alapjai*. Presa Universitară Clujeană, 2010.
2. Margit Antal. *Java alapú webtechnológiák*. Scientia, 2009.
3. Christian Bauer, Gavin King, Gary Gregory. *Java Persistence with Hibernate*. Manning Publications, 2015.
4. Madhusudhan Konda. *Just Spring Data Access: Covers JDBC, Hibernate, JPA and JDO*. O'Reilly Media, 2012.
5. Iuliana Cosmina. *Pivotal Certified Professional Spring Developer Exam: A Study Guide*. Apress, 2015.
6. Martin Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2002.

8.2 Seminar / laborator	Metode de predare	Observații
Săpt. 1-3.: Arhitecturi multilayer • proiect multimodul respectând arhitectura multilayer • afișarea datelor stocate în memorie • gestionarea erorilor	Exerciții interactive, evaluarea sarcinilor de laborator	
Săpt. 4-5.: Stratul de acces la date • JDBC • Abstract DAO Factory • Utilizarea fișierelor de configurare	Exerciții interactive, evaluarea sarcinilor de laborator	
Săpt. 6-7.: Aplicație web cu Java • Deployment a unei aplicații web pe un container web • Folosirea API-ului Servlet și al unui motor de șabloane	Exerciții interactive, evaluarea sarcinilor de laborator	
Săpt. 8-9.: Dependency Injection • Introducerea containerului IoC Spring • Spring beans	Exerciții interactive, evaluarea sarcinilor de laborator	
Săpt. 10-11.: Strat de afișare • Introducere Spring Web • DTO/Mapper	Exerciții interactive, evaluarea sarcinilor de laborator	
Săpt. 12-14.: Dezvoltarea unei aplicații multilayer cu un cadru enterprise • Introducere Persistence Java API • Strat modular/comutabil de acces la date • Introducere Spring Data JPA	Exerciții interactive, evaluarea sarcinilor de laborator	

Bibliografie:

1. Spring | Learn: <https://spring.io/learn>
2. Baeldung | Java, Spring and Web Development tutorials: <https://www.baeldung.com>
3. Martin Fowler: Patterns in Enterprise Software: <https://martinfowler.com/articles/enterprisePatterns.html>

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

• Tematica subiectului arată o mare suprapunere a conținutului de subiecte cu subiecte similare predate în marile universități. • Curriculumul se bazează pe activitatea celor mai apreciați autori internaționali, iar literatura recomandată se bazează și pe cele mai relevante lucrări din domeniu. • Subiectele predate în cadrul cursului sunt necesare pentru plasarea în industria soft, companiile așteptând astfel de cunoștințe.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Dobândirea adecvată a cunoștințelor teoretice prezentate în cursuri	Quiz la fiecare curs (Q)	20%
		Examen scris (EV)	30%

10.5 Seminar/laborator	Punerea în practică a cunoștințelor teoretice învățate	Exerciții de laborator (L)	10%
		Examen oral de laborator (LV)	40%
10.6 Standard minim de performanță			
Condițiile minime pentru trecere: ● Rezolvarea corectă a exercițiilor de laborator. ● Rezultatul examenului scris (EV) este de cel puțin 50%. ● Suma rezultatelor (Q+EV+L+LV) este de cel puțin 50%. ● Maxim 1 absență (nejustificată) de la orele de seminar. ● Maxim 1 absență (nejustificată) de la orele de laborator.			

11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)²

Nu se aplică.

Data completării:
29 aprilie 2025

Semnătura titularului de curs
Lect. dr. SÜLYÖK Csaba

Semnătura titularului de seminar
Lect. dr. SÜLYÖK Csaba

Data avizării în departament:
.....

Semnătura directorului de departament
Conf. dr. ANDRÁS Szilárd Károly

² Păstrați doar etichetele care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivesc disciplinei și ștergeți-le pe celelalte, inclusiv eticheta generală pentru Dezvoltare durabilă - dacă nu se aplică. Dacă nicio etichetă nu descrie disciplina, ștergeți-le pe toate și scrieți "Nu se aplică".