

Aprobat în ședința Consiliului profesoral al
Facultății Informatică, Inginerie și Design,
proces-verbal nr. 4 din 15 noiembrie 2024
Președinte CP al FIID
Mitev L., dr., conf. univ.

Aprobat în ședința Catedrei Tehnologii
Informaționale, proces-verbal nr.4 din
07 noiembrie 2024
Șef Catedră Tehnologii Informaționale

Bujac C., dr., lector univ.

Universitatea Liberă Internațională din Moldova
Facultatea Informatică, Inginerie și Design
Catedra Tehnologii Informaționale
Program de licență la specialitatea 0613.4 Informatică

CAIET DE SARCINI PENTRU EXAMENUL INTEGRAT DE LICENȚĂ
anul universitar 2024-2025

Probele examenului de licență:

1. Probă orală - *Prezentarea și susținerea tezei/proiectului de licență.*
2. Probă scrisă „*Cunoștințe fundamentale și de specialitate în informatică*”.

1. Probă orală - Prezentarea și susținerea tezei/proiectului de licență

Teza/proiectul de licență constituie o lucrare cu caracter teoretico-empiric, întocmită în baza activităților realizate în cadrul stagiilor de practică prevăzute în planul de studii la anul absolvent: III - pentru studenții care-și fac studii cu frecvență, și IV - cu frecvență redusă.

Conținutul tezei de licență urmează să reflecte formarea finalităților de studii.

Finalități de cunoștințe:

- stabilirea cadrului teoretic al cercetării în domeniu;
- posedarea cunoștințelor despre metodologia și metodele de cercetare științifică în domeniul profesional;

Finalități de abilități:

- elaborarea designului experimental al tezei/proiectul de licență, utilizând metode și tehnologii moderne, demonstrând competențe de explorare a informației științifice corespunzătoare, respectând principiile etice, cu citarea surselor, evitarea plagiatului și denaturării rezultatelor;
- selectarea și aplicarea perspectivelor demersului practic în raport cu persoana, grupul și situația, respectând principiile etice și deontologia profesională și în conformitate cu conținutul problemei și comanda socială;
- generalizarea rezultatelor cercetării experimentale, elaborarea de concluzii și recomandări practice;
- posedarea capacității de muncă în echipă și de organizare a relațiilor interpersonale.

Finalități de competențe:

- promovarea imaginii unui specialist în domeniul științelor informaticii aplicate, caracterizat de profesionalism în activitatea de cercetare și intervenție, de autenticitate, toleranță socială, spirit creativ și inovativ, responsabilitate personală, deschidere spre schimbare, autonomie a deciziilor;

- împărtășirea de atitudini empaticе și de asertivitate, de respect față de valorile general-umane, multiculturale, etico-profesionale și aplicarea a acestora în relațiile cu ambianța socială și în cadrul profesiei;
- operaționalizare a cunoștințelor teoretice din domeniu și de metodologie a cercetării, făcând distincție dintre abordările teoretice științifice, științifico-populare sau cu caracter speculativ, utilizând tehnicile de informare și de activitate independentă în procesul de dezvoltare continuă profesională, autodezvoltare prin autocunoaștere, autoevaluare și gândire critică, cunoscând o limbă de circulație internațională și tehnologiile informaționale moderne.

Teza/proiectul de licență va cuprinde o componenta de fundamentare teoretică și o componenta aplicativă care poate consta într-una din următoarele variante:

- microcercetare (colectarea, analiza și interpretarea de date obținute prin aplicarea unor metode precum: observarea, chestionarul, interviul, studiul de caz, analiza documentelor etc.);
- componentă de cercetare-dezvoltare (instrumente, mijloace, proiecte/programe etc. elaborate și testate în practică);
- experiment de testare a impactului unor teorii, practici, strategii, metode, procedee, tehnici, strategii, instrumente etc. asupra unor elemente/ domenii/ componente/ laturi fenomenului cercetat.

Criterii de evaluare a tezei/proiectului de licență

Evaluarea tezei/proiectului de licență se realizează în cadrul Comisiei de Licență. La evaluare se ține cont de:

- 1) prezentarea designului cercetării în introducere (consecutivitatea și corectitudinea planificării activității de cercetare, stabilirea problemei, obiectului, scopului, obiectivelor investigației empirice, formularea ipotezelor, constituirea eșantionului și selectarea metodelor de cercetare, planificarea activităților investigative);
- 2) capacitatea de a realiza un studiu bibliografico-științific, selectând și prezentând cele mai relevante investigații din domeniu care argumentează ipotezele experimentale);
- 3) prezentarea rezultatelor cercetării experimentale: capacitatea de a generaliza informația, a o analiza, clasifica și a sintetiza datele relevante, a le explica și a elabora concluzii);
- 4) capacitatea studentului de a prezenta lucrarea, inclusiv prin utilizarea mijloacelor electronice, de argumentarea și ilustrarea rezultatelor cercetării întreprinse.

2. Probă scrisă „Cunoștințe fundamentale și de specialitate în tehnologia informației”

Proba scrisă conține subiecte teoretice și aplicative, pe baza următoarelor discipline:

- I. Disciplina fundamentală „Programarea Java”
- II. Disciplina de specialitate „Programarea Python”

SUBIECTELE PENTRU EXAMENUL SCRIS DE LICENȚĂ
anul universitar 2024-2025

la specialitatea
0613.4 INFORMATICĂ

Disciplina fundamentală
Programarea Java

1. Istoria succintă a limbajului de programare Java, avantajele limbajului de programare Java
2. Structura unui program Java, funcția „main”, clase statice, tipul „void”
3. Tipuri de date primitive în limbajul de programare Java
4. Metode (funcții) de prelucrare a numerelor întregi și reale în limbajul de programare Java
5. Descrierea și inițializarea variabilelor în limbajul de programare Java
6. Afișarea informației pe ecran și citirea datelor de la tastatură în limbajul de programare Java
7. Operatori aritmetici, operatori de atribuire, de incrementare și de decrementare în Java
8. Operatori relaționali, operatori logici și operatorul condițional în limbajul Java
9. Operatori la nivel de bit în limbajul de programare Java, prioritatea operațiilor, utilizarea parantezelor pentru a modifica ordinea de efectuare a lor
10. Instrucțiuni condiționale în limbajul de programare Java
11. Cicluri în limbajul de programare Java
12. Tablouri unidimensionale în limbajul de programare Java: declararea, inițializarea, etc.
13. Prelucrarea tablourilor unidimensionale în limbajul de programare Java (determinarea sumei/ produsului, elementului maxim/ minim, parcurgerea, etc.)
14. Tablouri bidimensionale în limbajul de programare Java: declararea, inițializarea, etc.
15. Tablouri bidimensionale pătrate, tablouri bidimensionale în care fiecare rând are număr diferit de coloane
16. Prelucrarea tablourilor bidimensionale în limbajul de programare Java (determinarea sumei/ produsului, elementului maxim/ minim, parcurgerea, etc.)
17. Tablouri multidimensionale (tridimensionale ș.a.) în limbajul de programare Java
18. Proprietatea „length”, forma specială a ciclului „for” pentru tablouri (similar for-each din alte limbaje).
19. Sortarea tablourilor în limbajul de programare Java
20. Șiruri de caractere (Clasele String, StringBuffer și StringBuilder) în limbajul de programare Java: declararea, inițializarea, modificarea etc., deosebiri între clasele „String”, „StringBuffer” și „StringBuilder
21. Principalele metode folosite la prelucrarea șirurilor de caractere în limbajul Java
22. Avantajele programării orientate pe obiecte
23. Clase: declararea, operatori de protecție, set-teri și get-teri
24. Crearea obiectelor unei clase, operatorul „new”, Garbage Collector; metoda „finalize()”
25. Utilizarea claselor care corespund tipurilor de date primitive
26. Metode (funcții) în limbajul de programare Java, transmiterea parametrilor metodelor, întoarcerea valorilor de către metode, variabile locale, metode recursive
27. Metode supraîncărcate și suprascrise
28. Elemente statice ale clasei; specificatorul „final”
29. Constructori ai claselor, constructori cu parametri, operatorul „this”
30. Moștenirea în Java, avantajele și tipurile de moștenire
31. Operatorul „super”, cele două forme ale sale
32. Clase abstracte: declararea, utilizarea, metode abstracte
33. Interfețe: declararea, utilizarea
34. Deosebiri și asemănări dintre clase abstracte și interfețe

35. Pachete, avantajele polimorfismului, exemple
36. Utilizarea listelor, clasele „ArrayList”, „LinkedList” și altele
37. Colecții, mulțimi, dicționare, clasele „HashSet”, „TreeSet” și altele
38. Lucrul cu comparatorul colecțiilor
39. Lucrul cu fișierele, utilizarea serializării
40. Lucrul cu datele calendaristice, clasele „Calendar” și „GregorianCalendar”
41. Noțiunea de excepție, tipuri de excepții, prelucrarea excepțiilor (instrucțiunile „try”, „catch” și „finally”)
42. Generarea excepțiilor proprii, instrucțiunea „throw” și „throws”

Disciplinele de specialitate **Programarea Python**

1. Avantajele limbajului de programare Python, comentarii, identificatori și cuvinte cheie
2. Tipuri de date în limbajul de programare Python
3. Funcții (metode) de prelucrare a numerelor întregi și reale în limbajul de programare Python
4. Afișarea informației pe ecran și citirea datelor de la tastatură în limbajul de programare Python
5. Declararea, inițializarea și eliminarea variabilelor în limbajul de programare Python
6. Operatori aritmetici și de atribuire în limbajul de programare Python
7. Operatori relaționali (de comparare) și operatori logici în limbajul de programare Python
8. Operatori la nivel de bit în limbajul de programare Python
9. Instrucțiuni condiționale în limbajul de programare Python
10. Cicluri în limbajul de programare Python
11. Funcții (metode) în limbajul de programare Python, întoarcerea valorilor de către funcții, funcții cu parametri implicați, funcții cu număr variabil de parametri, funcția „len”
12. Transmiterea parametrilor către funcții, variabile locale și globale, funcții recursive
13. Tablouri unidimensionale în limbajul de programare Python: declararea, inițializarea, etc.
14. Prelucrarea tablourilor unidimensionale în limbajul de programare Python (determinarea sumei/ produsului, elementului maxim/ minim, parcurgerea, etc.)
15. Tablouri bidimensionale în limbajul de programare Python: declararea, inițializarea, etc.
16. Prelucrarea tablourilor bidimensionale în limbajul de programare Python (determinarea sumei/ produsului, elementului maxim/ minim, parcurgerea, etc.)
17. Șiruri de caractere în limbajul de programare Python: declararea, inițializarea, modificarea etc.
18. Principalele metode folosite la prelucrarea șirurilor de caractere în limbajul de programare Python
19. Liste în limbajul de programare Python
20. Tupl-uri (tuple) în limbajul de programare Python
21. Mulțimi în limbajul de programare Python
22. Dicționare în limbajul de programare Python
23. Clase în Python: declararea, modificarea dreptului de acces la membrii clasei, crearea obiectelor
24. Constructori și metode ale claselor în Python
25. Supraîncărcarea operatorilor în Python
26. Moștenire claselor, suprascrierea funcțiilor, apelarea constructorului superclasei
27. Lucrul cu fișierele în limbajul de programare Python
28. Noțiunea de excepții, tipuri de excepții, prelucrarea excepțiilor

Bibliografie selectivă

1. Barry Paul. Head First Python: A Brain-Friendly Guide. O'Reilly, 2016, 624 p.
2. Beazley David, Jones Brian K. Python Cookbook, III Edition. O'Reilly Media, 2013, Online, <http://chimera.labs.oreilly.com/books/12300000000393/>.
3. Campbell Jennifer, Gries Paul, Montojo Jason, Wilson Greg. Practical Programming. An Introduction to Computer Science Using Python, Pragmatic Bookshelf. 2009, 350 p.
4. Conținutul lucrărilor de laborator:
<https://www.dropbox.com/home/Lucrari%20de%20laborator/2020-21?preview=BPJ+lab+RO.docx>.
5. Elkner Jeffrey, Downey Allen B., Meyers Chris et al. Practical Programming in Python. Green Tea Press, 2015, 288 p.
6. Ghilic-Micu Bogdan și alții. Bazele programării calculatoarelor. Suport de seminar. București: Editura ASE, 2013, 132 p.
7. Ghilic-Micu Bogdan și alții. Bazele programării calculatoarelor. Suport de curs. București, Editura ASE, 2013, 200 p.
8. Jackson C. Learning to Program Using Python. 2013, 258 p.
<https://docs.google.com/file/d/0B8IUCMSuNpl7MnpaQ3hhN2R0Z1k/edit>.
9. Matthes Eric. Python Crash Course: A Hands-On, Project-Based Introduction to Programming. O'Reilly, 2015, 560 p.
10. Moise G., Constantinescu Z., Vlădoiu M., Dumitru M. Networking și Securitate. Ploiești: Editura Universitatii Petrol-Gaze, 2015, 260p.
11. Percival Harry. Test-Driven Development with Python. O'Reilly Media, 2014, <http://chimera.labs.oreilly.com/books/12340000000754/>
12. Ноутон П., Шилдт Г. Java 2. Наиболее полное руководство. Санкт- Петербург: БХВ-Петербург, 2007, 1050 с.
13. Шилдт Герберт. Java 8. Руководство для начинающих. VI издание. Санкт- Петербург, Вильямс, 2015, 720 с.

Link-uri utile:

1. Interactive Java Tutorial. www.learnjavaonline.org
2. Fundamental programming concepts. www.codecademy.com/learn/learn-java
3. Interactive Python Tutorial. www.learnpython.org
4. Python for Beginners. www.python.org
5. Python Tutorial. www.computer-pdf.com/programming/802-tutorial-python-tutorial.html
6. <http://pythonbooks.revolunet.com/>
7. <https://learnpythonthehardway.org/>
8. <http://inventwithpython.com/>
9. <http://www.onlineprogrammingbooks.com/python/>