Міністерство освіти і науки України

Харківський національний університет імені В.Н. Каразіна

Кафедра інформаційних технологій в фізико-енергетичних системах

"ЗАТВЕРДЖУЮ"
Директор
навчально-наукового інституту
комп'ютерної фізики та енергетики

Ірина ГАРЯЧЕВСЬКА

2024 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ІНТЕГРОВАНІ СЕРЕДОВИЩА РОЗРОБКИ

рівень вищої освіти	перший (бакалаврський) рівень
галузь знань	10 Природничі науки
спеціальність	105 Прикладна фізика та наноматеріали
освітня програма	Прикладна фізика енергетичних систем
	Прикладна фізика нетрадиційної енергетики
	Комп'ютерна фізика
вид дисципліни	обов'язкова
ННІ	комп'ютерної фізики та енергетики

2024 / 2025 навчальний рік

2

Програму рекомендовано до затвердження Вченою радою навчально-наукового інституту комп'ютерної фізики та енергетики

"25" грудня 2024 року, протокол № 12-2/24

РОЗРОБНИКИ ПРОГРАМИ:

Протектор Денис Олегович, PhD, доцент кафедри інформаційних технологій в фізико-енергетичних системах.

Програму схвалено на засіданні кафедри інформаційних технологій в фізико-енергетичних системах

Протокол від "24" грудня 2024 року № 7

Завідувач кафедри інформаційних технологій в фізико-енергетичних системах

(підпис) Руслан СУХОВ

Програму погоджено з гарантом освітньо-професійної програми (керівником проектної групи) Комп'ютерна фізика

Гарант освітньо-професійної програми (керівник проектної групи)

(підпис) Світлана РОГОВА

Програму погоджено з гарантом освітньо-професійної програми (керівником проектної групи) Прикладна фізика енергетичних систем

Гарант освітньо-професійної програми (керівник проектної групи)

(підпис) Денис ПРОТЕКТОР

Програму погоджено з гарантом освітньо-професійної програми (керівником проектної групи) Прикладна фізика нетрадиційної енергетики

Гарант освітньо-професійної програми (керівник проектної групи)

(підпис) Ілля МАРУЩЕНКО

Протокол від "25" грудня 2024 року № 12-2/24

Голова методичної комісії навчально-наукового інституту комп'ютерної фізики та енергетики

(підпис) Денис ПРОТЕКТОР

ВСТУП

Програма навчальної дисципліни “Інтегровані середовища розробки” складена відповідно до освітньо-професійної (освітньо-наукової) програми підготовки бакалаврів

спеціальність: 105 Прикладна фізика та наноматеріали

освітня програма: «Прикладна фізика енергетичних систем»
«Прикладна фізика нетрадиційної енергетики»
«Комп’ютерна фізика»

1. Опис навчальної дисципліни

1.1. Мета викладання навчальної дисципліни

Метою викладання навчальної дисципліни «Інтегровані середовища розробки» є формування базових знань та навичок, необхідних для створення додатків з графічним інтерфейсом користувача.

Предметом вивчення навчальної дисципліни є мови програмування C++, MATLAB та Python, знання яких є необхідним інструментом для створення сучасних додатків з графічним інтерфейсом користувача в різних галузях науки.

1.2. Основні завдання вивчення дисципліни

Основними завданнями вивчення дисципліни «Інтегровані середовища розробки» є оволодіння студентами основними навичками побудови додатків з графічним інтерфейсом користувача у середовищах Microsoft Visual Studio, MATLAB та PyCharm, з використанням мов програмування C++, MATLAB та Python відповідно.

1.3. Кількість кредитів 4

1.4. Загальна кількість годин 120

1.5. Характеристика навчальної дисципліни	
Обов’язкова	
Денна форма навчання	Заочна (дистанційна) форма навчання
Рік підготовки	
3-й	-й
Семестр	
6-й	-й
Лекції	
16 год.	год.
Практичні, семінарські заняття	
год.	год.
Лабораторні заняття	
32 год.	год.
Самостійна робота	
72 год.	год.

у тому числі індивідуальні завдання
год.

1.6. Заплановані результати навчання

Згідно з вимогами стандарту вищої освіти за спеціальністю 105 «Прикладна фізика та наноматеріали» галузі знань 10 «Природничі науки» для першого (бакалаврського) рівня вищої освіти студенти повинні досягти таких результатів навчання:

В результаті вивчення навчальної дисципліни студент повинен знати:

- основні принципи побудови додатків з графічним інтерфейсом користувача у середовищах Microsoft Visual Studio, MATLAB та PyCharm, з використанням мов програмування C++, MATLAB та Python.

В результаті вивчення навчальної дисципліни студент повинен вміти:

- створювати додатки з графічним інтерфейсом користувача у середовищах Microsoft Visual Studio, MATLAB та PyCharm, з використанням мов програмування C++, MATLAB та Python;
- застосовувати ефективні технології, інструменти та методи експериментального дослідження властивостей речовин і матеріалів, включаючи наноматеріали, при розв’язанні практичних проблем прикладної фізики;
- застосовувати фізичні, математичні та комп'ютерні моделі для дослідження фізичних явищ, розробки приладів і наукоємних технологій;
- вибирати ефективні методи та інструментальні засоби проведення досліджень у галузі прикладної фізики;
- класифікувати, аналізувати та інтерпретувати науковотехнічну інформацію в галузі прикладної фізики;
- презентувати результати досліджень і розробок фахівцям і нефахівцям, аргументувати власну позицію;
- проектувати та розробляти складові енергетичних комплексів;
- здійснювати математичне та комп’ютерне моделювання фізичних процесів в енергетичних комплексах;
- розробляти програмне та програмно-апаратне забезпечення в фізико-енергетичних системах.

2. Тематичний план навчальної дисципліни

Розділ 1. Можливості пакета прикладних програм MATLAB.

Тема 1. Вступ в MATLAB. Основи роботи в середовищі GUIDE.

Тема 2. Створення додатку з графічним інтерфейсом в середовищі GUIDE.

Тема 3. Створення додатку з графічним інтерфейсом мовою програмування MATLAB.

Розділ 2. Можливості інтегрованого середовища розробки PyCharm.

Тема 4. Вступ в Python. Основи роботи в середовищі PyCharm.

Тема 5. Вивчення бібліотек для роботи з графічним інтерфейсом користувача в Python.

Тема 6. Створення додатку з графічним інтерфейсом мовою програмування Python.

Розділ 3. Можливості інтегрованого середовища розробки Microsoft Visual Studio.

Тема 7. Вступ в Microsoft Visual Studio. Панель інструментів.

Тема 8 . Створення додатку з графічним інтерфейсом мовою програмування C++.

3. Структура навчальної дисципліни

Назви розділів і тем	Кількість годин											
	денна форма						заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб	сем	с.р.		л	п	лаб	сем	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
Розділ 1. *Можливості пакета прикладних програм MATLAB*												
Тема 1. Вступ в MATLAB. Основи роботи в середовищі GUIDE.	15	2		4		9						
Тема 2. Створення додатку з графічним інтерфейсом в середовищі GUIDE.	15	2		4		9						
Тема 3. Створення додатку з графічним інтерфейсом мовою програмування MATLAB.	15	2		4		9						
Розділ 2. *Можливості інтегрованого середовища розробки PyCharm*												
Тема 4. Вступ в Python. Основи роботи в середовищі PyCharm.	15	2		4		9						
Тема 5. Вивчення бібліотек для роботи з графічним інтерфейсом користувача в Python.	15	2		4		9						
Тема 6. Створення додатку з графічним інтерфейсом мовою програмування Python.	15	2		4		9						
Розділ 3. *Можливості інтегрованого середовища розробки Microsoft Visual Studio*												
Тема 7. Вступ в Microsoft Visual Studio. Панель інструментів.	15	2		4		9						
Тема 8 . Створення додатку з графічним інтерфейсом мовою програмування C++.	15	2		4		9						
Усього годин	120	16		32		72						

4. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Вступ в MATLAB. Основи роботи в середовищі GUIDE.	4
2	Створення додатку з графічним інтерфейсом в середовищі GUIDE.	4
3	Створення додатку з графічним інтерфейсом мовою програмування MATLAB.	4
4	Вступ в Python. Основи роботи в середовищі PyCharm.	4
5	Вивчення бібліотек для роботи з графічним інтерфейсом користувача в Python.	4
6	Створення додатку з графічним інтерфейсом мовою програмування Python.	4
7	Вступ в Microsoft Visual Studio. Панель інструментів.	4
8	Створення додатку з графічним інтерфейсом мовою програмування C++.	4
Усього годин		32

5. Завдання для самостійної роботи

№ з/п	Назва теми	Кількість годин
1	Оволодіти навичками роботи в середовищі GUIDE.	9
2	Навчитися створенню додатків з графічним інтерфейсом в середовищі GUIDE.	9
3	Навчитися створенню додатків з графічним інтерфейсом мовою програмування MATLAB.	9
4	Оволодіти навичками роботи в середовищі PyCharm.	9
5	Ознайомитися з бібліотеками для роботи з графічним інтерфейсом користувача в Python.	9
6	Навчитися створенню додатків з графічним інтерфейсом мовою програмування Python.	9
7	Навчитися працювати з панеллю інструментів Microsoft Visual Studio.	9
8	Навчитися створенню додатків з графічним інтерфейсом мовою програмування C++.	9
Усього годин		72

6. Індивідуальні завдання

Не передбачено.

7. Методи навчання

Методи навчання, що застосовуються при викладанні навчальної дисципліни «Інтегровані середовища розробки»:

1) Пояснювально-ілюстративний метод. Викладання лекційного матеріалу дисципліни у вигляді презентацій за допомогою мультимедійного обладнання.

2) Репродуктивний метод. Відтворення студентами набутих теоретичних знань при виконанні лабораторних робіт.

3) Дослідницький метод. Виконання студентами самостійних робіт.

8. Методи контрою

1) Написання контрольної роботи. За навчальним планом передбачено дві контрольні роботи. Максимальна кількість балів, яку може отримати студент за написання однієї контрольної роботи, складає 10 балів. Написання контрольної роботи полягає в складанні алгоритму роботи програми у вигляді блок-схеми з подальшою реалізацією алгоритму роботи програми однією з мов програмування C++, MATLAB або Python. Нарахування балів здійснюється за наступними критеріями:

10 балів нараховується, коли:

- блок-схема та алгоритм роботи програми складені логічно правильно;
- правильно оформлені вхідні та вихідні дані;
- немає помилок у використанні структурних елементів блок-схеми та алгоритму роботи програми;
- студент без помилок читає блок-схему та алгоритм роботи програми;
- початковий код програми не має синтаксичних та логічних помилок;
- студент вільно орієнтується в початковому коді програми.

8 балів нараховується, коли:

- блок-схема та алгоритм роботи програми складені логічно правильно, але були допущені 1-2 помилки або 2-3 недоліка;
- початковий код програми складено логічно правильно, але були допущені 1-2 синтаксичні помилки або 2-3 недоліка.

6 балів нараховується, коли:

- допущені помилки в алгоритмі роботи програми;
- неправильно використовуються структурні елементи блок-схеми;
- у поясненні алгоритму роботи програми та блок-схеми студент відчував труднощі, які були виправлені за допомогою викладача;
- початковий код програми має синтаксичні та логічні помилки;
- у поясненні початкового коду програми студент відчував труднощі, які були виправлені за допомогою викладача.

4 бали нараховується, коли:

- допущені істотні помилки в оформленні алгоритму роботи програми та блок-схеми;
- студент не володіє основними правилами оформлення алгоритму роботи програми та блок-схеми;
- допущені грубі помилки в алгоритмі роботи програми, які студент не може виправити навіть за допомогою додаткових запитань викладача;
- початковий код програми має істотні синтаксичні та логічні помилки, які студент не може виправити навіть за допомогою додаткових запитань викладача.

0 балів нараховується, коли:

- студент показує повне незнання алгоритмічних конструкцій і структурних елементів блок-схеми;
- студент показує повне незнання синтаксису мови програмування.

За семестр студент може набрати максимум 20 балів за виконання 2 контрольних робіт.

2) Написання тестової роботи. Курс «Інтегровані середовища розробки» передбачає написання однієї тестової роботи за пройденим лекційним матеріалом семестру. Максимальна кількість балів за тестову роботу складає 14 балів.

3) Виконання лабораторних робіт. В навчальній дисципліні передбачено виконання 8 лабораторних робіт. За виконання кожної лабораторної роботи студент отримує від 0 до 2 балів. Нарахування балів здійснюється за оформлений звіт, початковий код програми, відповіді на питання по коду програми, проходження тестів без збоїв в роботі програми. За виконання усіх лабораторних робіт студент за семестр може набрати максимум 16 балів.

4) Захист самостійних робіт. В навчальній дисципліні передбачено розробку трьох незалежних додатків з графічним інтерфейсом користувача, написаних на мовах програмування C++, MATLAB та Python. Захист кожної самостійної роботи проводиться у вигляді презентації розробленого додатка. Кожна робота оцінюється в 10 балів. Нарахування балів здійснюється за наступними критеріями:

- алгоритм роботи розробленого додатка (2 бали);
- графічний інтерфейс розробленого додатка (2 бали);
- функціонування розробленого додатка (2 бали);
- презентація розробленого додатка (2 бали);
- відповіді на запитання по коду розробленого додатка (2 бали).

За семестр студент може набрати максимум 30 балів за виконання 3 самостійних робіт.

5) Написання залікової роботи. Максимальна кількість балів, яку студент може отримати за написання залікової роботи складає 20 балів.

9. Схема нарахування балів

Підсумковий семестровий контроль в формі заліку.

Мінімальна кількість балів для допуску до підсумкового контролю – 30 балів.

Критерії оцінювання навчальних досягнень

Назва роботи	**Оцінка**
Лабораторна робота № 1	2
Лабораторна робота № 2	2
Лабораторна робота № 3	2
Лабораторна робота № 4	2
Лабораторна робота № 5	2
Лабораторна робота № 6	2
Лабораторна робота № 7	2
Лабораторна робота № 8	2
Самостійна робота № 1	10
Самостійна робота № 2	10
Самостійна робота № 3	10
Контрольна робота № 1	10
Контрольна робота № 2	10
Тестова робота	14
Залікова робота	20
Сума	**100**

Шкала оцінювання

Сума балів за всі види навчальної діяльності протягом семестру	Оцінка	
	для чотирирівневої шкали оцінювання	для дворівневої шкали оцінювання
90 – 100	відмінно	зараховано
70-89	добре	
50-69	задовільно	
1-49	незадовільно	не зараховано

10. Рекомендована література

Основна література

1) *Bjarne Stroustrup*. Programming: Principles and Practice Using C++. 2nd Edition. (2014)
2) *Scott Meyers*. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14. (2014)
3) *Amos Gilat*. MATLAB: An Introduction with Applications. 5th Edition. (2014)
4) *Stormy Attaway* Ph.D. MATLAB: A Practical Introduction to Programming and Problem Solving. 5th Edition. (2018)
5) *Mark Lutz*. Learning Python. 5th Edition. (2013)
6) *Eric Matthes*. Python Crash Course. 2nd Edition: A Hands-On, Project-Based Introduction to Programming. (2019)

Допоміжна література

1) *Deitel P., Deitel H.* C++ How to Program, 10th Edition. (2016).
2) *Stanley B. Lippman*. C++ Primer: Completely Rewritten for the New C++11 Standard. (2012).

11. Посилання на інформаційні ресурси в Інтернеті, відео-лекції, інше методичне забезпечення

1) https://docs.python.org/3/
2) https://www.mathworks.com/help/matlab/
3) https://learn.microsoft.com/en-us/cpp/cpp/?view=msvc-170
4) https://learn.microsoft.com/en-us/cpp/dotnet/dotnet-programming-with-cpp-cli-visual-cpp?view=msvc-170