

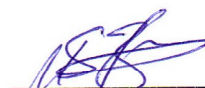
6

Міністерство освіти і науки України
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Кафедра комп'ютерних систем, мереж і кібербезпеки (№ 503)
(назва кафедри)

ЗАТВЕРДЖУЮ

Голова НМК


(підпис)

М.С. Зряхов
(ініціали та прізвище)

« 30 » серпня 2019 р.

**РОБОЧА ПРОГРАМА ОBOB'ЯЗКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Технології програмування
(назва навчальної дисципліни)

Галузь знань:	12 «Інформаційні технології» (шифр і найменування галузі знань)
Спеціальність:	123 «Комп'ютерна інженерія» (код та найменування спеціальності)
Освітня програма:	«Комп'ютерні системи та мережі», «Системне програмування», «Програмовні мобільні системи та інтернет речей» (найменування освітньої програми)
Спеціальність:	125 «Кібербезпека» (код та найменування спеціальності)
Освітня програма:	«Безпека інформаційних і комунікаційних систем», «Кібербезпека індустріальних систем» (найменування освітньої програми)
Форма навчання:	денна
Рівень вищої освіти:	перший (бакалаврський)

Харків 2019 рік

Робоча програма

«Технології програмування»

(назва дисципліни)

для студентів за спеціальністю

123 «Комп'ютерна інженерія»

(код та найменування спеціальності)

спеціалізаціями

«Комп'ютерні системи та мережі», «Системне програмування»,

«Програмовні мобільні системи та інтернет речей»

(назви освітніх програм)

(за цією спеціальністю також: 125 Кібербезпека, освітні програми «Безпека інформаційних і комунікаційних систем», «Кібербезпека індустріальних систем»)

« 22 » серпня 2019 р. , – 11 с.

Розробник: Бабешко Євген Васильович, старший викладач, к.т.н.

(прізвище та ініціали, посада, науковий ступінь та вчене звання)

(підпис)

Робочу програму розглянуто на засіданні кафедри

комп'ютерних систем, мереж і

(назва кафедри)

кібербезпеки

Протокол № 1 від « 30 » серпня 2019 року

Завідувач кафедри

комп'ютерних систем, мереж і кібербезпеки

(назва кафедри)

Д.т.н., професор

(науковий ступінь та вчене звання)

(підпис)

В.С. Харченко

(ініціали та прізвище)

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, спеціалізація, рівень вищої освіти	Характеристика навчальної дисципліни
		Денна форма навчання
Кількість кредитів: 4.5	Галузь знань: 12 «Інформаційні технології»	Цикл професійної підготовки
Модулів – 2	Спеціальність: 123 «Комп'ютерна інженерія» Освітні програми: «Комп'ютерні системи та мережі», «Системне програмування», «Програмовні мобільні системи та інтернет речей»	Навчальний рік 2019/2020
Змістових модулів – 2		Семестр
Індивідуальне науково- дослідне завдання: немає		
Загальна кількість годин – 64 ¹⁾ /135		
Тижневих годин для денної форми навчання: аудиторних – 4 самостійної роботи студента – 9	Спеціальність: 125 «Кібербезпека» Освітні програми: «Безпека інформаційних і комунікаційних систем»», «Кібербезпека індустріальних систем»	3
		Лекції ¹⁾
		32 години
		Практичні ¹⁾
		0 годин
		Лабораторні ¹⁾
		32 годин
		Самостійна робота
		71 година
		Вид контролю
		Іспит

Співвідношення кількості годин аудиторних занять до самостійної роботи становить 64/71.

¹⁾ Аудиторне навантаження може бути зменшене або збільшене на одну годину в залежності від розкладу занять.

2. Мета та завдання навчальної дисципліни

Мета вивчення: (ОК31) надання необхідних знань з об'єктно-орієнтованого програмування, а також формування твердих практичних навичок щодо розроблення програмного забезпечення з використанням об'єктно-орієнтованого підходу., а також засвоєння основних положень об'єктного принципу при створенні комп'ютерних програм, вивчення мови програмування високого рівня C# та стандартних бібліотек класів .NET Framework Class Library.

Завдання: (ОК31) придбання студентами необхідних знань та вмінь в сфері проектування програмного забезпечення на основі відповідного набору абстрактних типів даних, вирішення прикладних задач з використанням стандартних типів даних і типів даних власної розробки, отримання навиків використання ключових концепцій об'єктно-орієнтованого підходу, а також

- вивчення основних методологічних особливостей об'єктно-орієнтованого програмування;
- вивчення базових конструкцій мови C#;
- використання типів даних мови C# та платформи .NET;
- освоєння принципів проектування програм з використанням об'єктної декомпозиції.

Програмні компетентності. Дисципліна має допомогти сформувати у студентів такі компетентності:

- (ЗК1) здатність до абстрактного мислення, аналізу і синтезу;
- (ЗК2) здатність вчитися і оволодівати сучасними знаннями;
- (ЗК3) здатність застосовувати знання у практичних ситуаціях;
- (ЗК4) здатність спілкуватися державною мовою як усно, так і письмово;
- (ЗК7) вміння виявляти, ставити та вирішувати проблеми;
- (ФК2) здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення;
- (ФК3) здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

Програмні результати навчання. У результаті вивчення дисципліни студенти мають досягти такі програмні результати навчання:

- (ПРН18) використовувати інформаційні технології та для ефективного спілкування на професійному та соціальному рівнях;
- (ПРН21) якісно виконувати роботу та досягати поставленої мети з дотриманням вимог професійної етики.

Крім того, студенти повинні бути здатними до рішення задач з використанням принципів об'єктно-орієнтованого програмування.

Міждисциплінарні зв'язки: Дисципліна базується на знаннях, отриманих під час вивчення дисциплін у циклі загальної і професійної підготовки, передбачених навчальним планом спеціальності. Матеріал дисципліни базується на знаннях, отриманих під час вивчення дисциплін (ОК9) Дискретна математика, (ОК36) Навчальна практика.

Матеріал, засвоєний під час вивчення цієї дисципліни, є базою для дисципліни (ВБ1.14) Технології програмування.

3. Програма навчальної дисципліни

Семестр 3

Модуль 1

Змістовий модуль 1

Тема 1. Вступ до навчальної дисципліни

Передумови та необхідність переходу до об'єктно-орієнтованого програмування. Декомпозиція, абстракція та ієрархія як базові принципи подолання складності, закладені в об'єктно-орієнтованому підході.

Загальні поняття та зв'язок об'єктно-орієнтованого аналізу, проектування та програмування. Огляд об'єктно-орієнтованих мов програмування.

Тема 2. Архітектура платформи .NET та програмування для .NET Framework

Архітектура платформи .NET та мови програмування для .NET Framework. Основи програмування мовою C#: змінні та типи даних (величини та посилання), структура програми та точка входу (метод Main). Клас як тип даних. Базовий клас System.Object.

Організація введення та виводу даних в консольних програмах та засоби управління консоллю (базові можливості зміни властивостей). Приклад організації введення числових даних з контролем помилок.

Загальна характеристика стандартної бібліотеки класів та просторів імен.

Тема 3. Синтаксис мови C#. Основні операції, оператори та типи даних C#

Реалізація класів в програмі: опис класів, поля даних, методи (функції члени класу). Ініціалізація полів даних та помилки, що виникають при використанні неініційованих даних.

Зони видимості та спеціфікатори доступу public та private. Огляд принципу інкапсуляції даних.

Передача параметрів (даних та посилань) в методи та дані, що повертаються. Перевантаження методів для розширення функціональності класів. Управляючи оператори, оператори та вирази в мові C#: синтаксис та приклади використання.

Тема 4. Класи та об'єкти. Перетворення типів даних

Перерахування, структури та клас string. Рядкові константи та їх подання в програмі. Коди символів та спрощення запису рядків-констант за допомогою символу @.

Створення об'єктів (екземплярів класу) оператором new. Статичні члени класу. Константні члени класу.

Поняття базового класу та спадкування характеристик між класами. Приведення типів: оператори as та is. Боксування (упаковка).

Тема 5. Взаємодія об'єктів в програмі, механізми оброблення помилок та позаштатних ситуацій

Поля, що зберігають дані, як елементи, що відображують стан об'єктів, та методи, як механізми зміни стану (підсумок принципу інкапсуляції).

Взаємодія об'єктів в процесі функціонування.

Позаштатні ситуації, їх оброблення та створення для передачі інформації про виникнення помилки до метода, який викликав функцію, що привела до виникнення

позаштатної ситуації. Оператори checked та unchecked. Можливість створення своїх класів-позаштатних ситуацій.

Тема 6. Ініціалізація стану об'єкта, призначення та використання конструкторів

Властивість яз засіб доступу до даних. Різновиди властивостей. Поля readonly.

Конструктори та їх призначення. Конструктори за замовчуванням. Параметри конструктора та перевантаження конструкторів. Закриті та статичні конструктори. Конструктори для структур.

Тема 7. Механізми керування пам'яттю та іншими ресурсами

Механізми виділення пам'яті під об'єкти та видалення об'єктів з пам'яті. Garbage collector: оптимізація його роботи та програмне управління вивільненням пам'яті.

Явне вивільнення ресурсів та слабкі посилання. Метод Finalize.

Модуль 2

Змістовий модуль 2

Тема 8. Спадкування. Різновиди спадкування та їх застосування при розробленні програм

Спадкування реалізації та спадкування інтерфейсу. Віртуальні методи. Закриття методів. Виклик базових версій методів. Абстрактні за закриті класи та методи (ключові слова abstract та sealed). Конструктори в похідних класах.

Огляд модифікаторів для класів, методів та полів.

Тема 9. Інтерфейси та їх використання

Визначення та реалізація інтерфейсів. Похідні інтерфейси. Приклади використання інтерфейсів (IClonable, IComparable, IDisposable). Зауваження стосовно множинного спадкування.

Спадкування, поліморфізм та інкапсуляція як базові аспекти ООП – підведення підсумків стосовно концепцій об'єктно-орієнтованого підходу.

Тема 10. Робота з файлами та каталогами. Серіалізація та десеріалізація стану об'єктів

Простір імен System.IO та потоки. Читання та запис файлів (FileStream, об'єкти - Reader та -Writer).

Робота с файлами та каталогами на диску (File, FileInfo, Directory, DirectoryInfo, FileSystemWatcher).

Приклад асинхронної роботи з файлами. Поняття серіалізації. Приклад бінарної серіалізації: атрибут [Serializable], інтерфейс IFormatter та клас BinaryFormatter.

Тема 11. Використання масивів та колекцій для упорядкування даних в програмі

Групування об'єктів. Масиви в C#: лінійні, багатовимірні та ламані.

Поняття колекції та перераховувача. Можливості та приклади використання класів ArrayList, Stack та Queue.

Тема 12. Використання колекція та узагальнених колекцій для спрощення реалізації складних алгоритмів оброблення даних

Можливості та реалізація колекції SortedList. Приклад використання SortedList.

Словники та хеш-таблиці. Реалізація словників в .NET.

Тема 13. Оброблення символічних та строкових даних з використанням стандартних класів .NET

Оброблення символів та рядків. Перетворення рядків в інші типи даних (методи Parse() та ToString()) та форматування (метод Format()). Можливості пошуку, порівняння та модифікації рядків. Можливості класу StringBuilder.

Регулярні вирази: створення (клас System.Text.RegularExpressions.Regex), типи виразів (RegexOptions) та методи для оброблення рядків.

Тема 14. Делегати та події. Застосування технології Windows Forms для створення графічного інтерфейсу користувача

Делегати та події як засіб взаємодії та передачі даних між об'єктами.

Поняття Windows Forms. Створення простої форми та її властивості. Методи та події форми. Реакція форми на події.

Тема 15. Огляд елементів управління Windows Forms

Елементи управління та їх класифікація. Наповнення форми елементами управління та визначення функціональності для них.

Короткий огляд стандартних діалогових вікон.

Тема 16. Моделювання програмного забезпечення за об'єктно-орієнтованими принципами

Моделі життєвого циклу програмного забезпечення. Microsoft Solutions Framework як основа організації процесів створення програм: етапи та завдання що на них вирішуються.

Моделювання програмних систем на базі принципів об'єктно-орієнтованого аналізу. UML як інструмент моделювання програмних систем. Поняття model-driven development та перспективні технології проектування програм. Unit-тестування.

Тема 17. Застосування шаблонів проектування при розробленні програмних систем

Поняття шаблонів проектування. Призначення та сфера використання шаблонів проектування. Стислий огляд шаблонів та прикладів їх використання: Factory, Singleton, Builder, Prototype, Adapter, Bridge тощо. Підходи до використання шаблонів та джерела інформації про шаблони. Зв'язок з подальшими дисциплінами.

4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин				
	Денна форма				
	Усього	У тому числі			
		л	п	лаб.	с. р.
1	2	3	4	5	6
Модуль 1					
Змістовий модуль 1					
1. Вступ до навчальної дисципліни	1	1			
2. Архітектура платформи .NET та програмування для .NET Framework	8	2		2	4
3. Синтаксис мови C#. Основні операції, оператори та типи даних C#	8	2		2	4
4. Класи та об'єкти. Перетворення типів даних	8	2		2	4
5. Взаємодія об'єктів в програмі,	8	2		2	4

механізми оброблення помилок та позаштатних ситуацій					
6. Ініціалізація стану об'єкта, призначення та використання конструкторів	8	2		2	4
1	2	3	4	5	6
7. Механізми керування пам'яттю та іншими ресурсами	8	2		2	4
Модульний контроль	1	1			
Разом за змістовим модулем 1	50	14		12	24
Модуль 2					
Змістовий модуль 2					
8. Спадкування. Різновиди спадкування та їх застосування при розробленні програм	8	2		2	4
9. Інтерфейси та їх використання	7	1		2	4
10. Робота з файлами та каталогами. Серіалізація та десеріалізація стану об'єктів	7	1		2	4
11. Використання масивів та колекцій для упорядкування даних в програмі	8	2		2	4
12. Використання колекцій та узагальнених колекцій для спрощення реалізації складних алгоритмів оброблення даних	7	1		2	4
13. Оброблення символьних та строкових даних з використанням стандартних класів .NET	8	2		2	4
14. Делегати та події. Застосування технології Windows Forms для створення графічного інтерфейсу користувача	8	2		2	4
15. Огляд елементів управління Windows Forms	8	2		2	4
16. Моделювання програмного забезпечення за об'єктно-орієнтованими принципами	11	2		2	7
17. Застосування шаблонів проектування при розробленні програмних систем	12	2		2	8
Модульний контроль	1	1			
Разом за змістовим модулем 2	85	18		20	47
Усього годин за дисципліною	135	32		32	71

5. Теми семінарських занять

№ з/п	Назва теми	Кількість годин
1	<i>Не передбачено</i>	
	Разом	

6. Теми практичних занять

№ з/п	Назва теми	Кількість годин
1	<i>Не передбачено</i>	
	Разом	

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Створення бібліотек класів і консольних додатків	4
2	Абстрактні сутності та зв'язки між ними	4
3	Спадкування та поліморфізм	4
4	Перевантаження методів та операторів. Інтерфейси	4
5	Колекції	4
6	Робота з файлами та серіалізація	4
7	Регулярні вирази. Unit-тестування	8
	Разом	32

8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Відпрацювання матеріалів лекційних занять за темою 1	2
2	Відпрацювання матеріалів лекційних занять за темою 2	4
3	Відпрацювання матеріалів лекційних занять за темою 3	4
4	Відпрацювання матеріалів лекційних занять за темою 4	4
5	Відпрацювання матеріалів лекційних занять за темою 5	4
6	Відпрацювання матеріалів лекційних занять за темою 6	4
7	Відпрацювання матеріалів лекційних занять за темою 7	4
8	Відпрацювання матеріалів лекційних занять за темою 8	4
9	Відпрацювання матеріалів лекційних занять за темою 9	4
10	Відпрацювання матеріалів лекційних занять за темою 10	4
11	Відпрацювання матеріалів лекційних занять за темою 11	4
12	Відпрацювання матеріалів лекційних занять за темою 12	4
13	Відпрацювання матеріалів лекційних занять за темою 13	4
14	Відпрацювання матеріалів лекційних занять за темою 14	4
15	Відпрацювання матеріалів лекційних занять за темою 15	4
16	Відпрацювання матеріалів лекційних занять за темою 16	5
17	Відпрацювання матеріалів лекційних занять за темою 17	8
	Разом	71

9. Індивідуальні завдання

Не передбачено

10. Методи навчання

Проведення аудиторних лекцій, лабораторних робіт, консультацій, а також самостійна робота студентів з використанням відповідних матеріалів (п.14, 15).

11. Методи контролю

Проведення поточного контролю, електронного тестування, підсумковий контроль у вигляді іспиту.

12. Критерії оцінювання та розподіл балів, які отримують студенти

12.1. Розподіл балів, які отримують студенти (кількісні критерії оцінювання)

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовий модуль 1			
Лабораторні роботи	0...10	3	0...30
Тести	0...5	1	0...5
Модульний контроль	0..10	1	0..10
Змістовий модуль 2			
Лабораторні роботи	0...10	4	0...40
Тести	0...5	1	0...5
Модульний контроль	0..10	1	0..10
Усього за семестр			0...100

12.2. Якісні критерії оцінювання

Необхідний обсяг знань для одержання позитивної оцінки

1. Знати ключові принципи об'єктно-орієнтованого програмування
2. Знати можливості та основні положення роботи з мовою програмування C#
3. Знати призначення класів, абстрактних класів, інтерфейсів, делегатів
4. Знати призначення діаграми класів
5. Знати основні шаблони (патерни) проектування

Необхідний обсяг умінь для одержання позитивної оцінки

1. Уміти проектувати об'єктну модель предметної області
2. Уміти створювати програми з консольним та графічним інтерфейсом для розв'язання поставленої задачі
3. Уміти використовувати регулярні вирази для оброблення текстової інформації
4. Уміти використовувати серіалізацію та десеріалізацію

12.3 Критерії оцінювання роботи студента протягом семестру

Задовільно (60-74). Показати мінімум знань та умінь. Захистити не менше 75% від усіх завдань лабораторних занять.

Добре (75-89). Твердо знати мінімум, захистити не менше 90% завдань лабораторних занять.

Відмінно (90-100). Здати всі контрольні точки з оцінкою «відмінно». Досконально знати всі теми та уміти їх застосовувати.

Шкала оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційований залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

13. Методичне забезпечення

Навчально-методичний комплекс дисципліни розміщений у системі управління курсами кафедри комп'ютерних систем, мереж і кібербезпеки.

1. Система управління курсами кафедри комп'ютерних систем, мереж і кібербезпеки [Ел. ресурс]. URL: <https://elearn.csn.khai.edu>

14. Рекомендована література

Базова

1. Коноваленко І.В. Програмування мовою C# 6.0. Тернопіль, ТНТУ, 2016. 227 с.
2. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 2017 896 с.
3. Троелсен Э., Джепикс Ф. Язык программирования C# 6.0 и платформа .NET 4.6. 7-е издание. К.: Вильямс, 2017. 1440 с.

Допоміжна

1. Booch B. Object-Oriented Analysis and Design with Applications. 3rd edition. Addison-Wesley. 720 p.
2. Kimmel P. UML demystified. A self-teaching guide. McGraw-Hill Education. 235 p.
3. Нейгел К., Ивсен Б., Глинн Д., Уотсон К., Скиннер М. C# 5.0 и платформа .NET 4.5 для профессионалов. 2013. 1440 с.

15. Інформаційні ресурси

2. C# Programming. Yellow Book [Ел. ресурс]. URL: <https://www.robmiles.com/c-yellow-book>
3. Stackoverflow на русском. Вопросы с меткой «C#» [Ел. ресурс]. URL: <https://ru.stackoverflow.com/questions/tagged/c#>
4. Система управління курсами кафедри комп'ютерних систем, мереж і кібербезпеки [Ел. ресурс]. URL: <https://elearn.csn.khai.edu>