

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

Кафедра математичного моделювання та штучного інтелекту (№ 304)

**ЗАТВЕРДЖУЮ**

Керівник проектної групи



(підпис)

Д. І. Чумаченко

(ініціали та прізвище)

« 30 » 08 2023 р.

**РОБОЧА ПРОГРАМА ОBOB'ЯЗКОВОЇ  
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

**Об'єктно-орієнтоване програмування (мова C#)**

(назва навчальної дисципліни)

Галузь знань: 12 "Інформаційні технології "  
(шифр і найменування галузі знань)

Спеціальність: 122 "Комп'ютерні науки"  
(код та найменування напрямку підготовки)

Освітня програма: "Інтелектуальні системи та технології"  
(найменування освітньої програми)

Форма навчання: денна

Рівень вищої освіти: перший (бакалаврський)

Харків 2023 рік

Робоча програма навчальної дисципліни

«Об'єктно-орієнтоване програмування (мова C#)»

(назва навчальної дисципліни)

для студентів за спеціальністю 122 «Комп'ютерні науки»

освітніми програмами "Інтелектуальні системи та технології"

«27» серпня 2023 р. – 13 с.

РОЗРОБНИК ПРОГРАМИ:

Скоб Ю. О., проф. кафедри 304, д. т. н., доцент

(прізвище та ініціали, посада, наукова ступень та вчене звання)



(підпис)

Робочу програму розглянуто на засіданні випускової кафедри Математичного моделювання та штучного інтелекту» (№ 304)

(назва кафедри)

Протокол № 1 від «30» серпня 2023 р.

Завідувач  
кафедри № 304

д.т.н., професор

(науковий ступінь і вчене звання)



(підпис)

А. Г. Чухрай

(ініціали та прізвище)

## 1. Опис навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни	
		Денна форма навчання	
Кількість кредитів – 4,0	<b>Галузь знань</b> <u>12 "Інформаційні технології"</u> <small>(шифр та найменування)</small>  <b>Спеціальність</b> <u>122 "Комп'ютерні науки"</u> <small>(код та найменування)</small>  <b>Освітня програма</b> <u>"Інтелектуальні системи та технології"</u> <small>(найменування)</small>  <b>Рівень вищої освіти:</b> <u>перший (бакалаврський)</u>	Обов'язкова	
Кількість модулів – 1		<b>Навчальний рік</b>	
Кількість змістових модулів – 2			
Індивідуальне науково-дослідне завдання – розрахункова робота «Програмування на основі інтерфейсів»		<b>Семестр</b>	
Загальна кількість годин – 120 денна – 64/56		3-й	
		<b>Лекції</b>	
Кількість тижневих годин для денної форми навчання: аудиторних – 4,0 самостійної роботи студента - 3,5		32 год.	
		<b>Практичні, семінарські</b>	
		32 год.	
		<b>Лабораторні</b>	
	–		
	<b>Самостійна робота</b>		
56 год.			
<b>Індивідуальна робота</b>			
–			
<b>Вид контролю</b>			
модульний контроль, іспит			

### Примітка

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

для денної форми навчання – 64/56.

## 2. Мета та завдання навчальної дисципліни

### Мета навчальної дисципліни:

підготовка спеціалістів з інформаційних систем до виконання робіт з розроблення програмного забезпечення з використанням принципів та методів об'єктно-орієнтованого програмування; формування у студентів уявлення про основні принципи об'єктно-орієнтованої моделі побудови програмного забезпечення, вмінь об'єктно-орієнтованого мовно-незалежного проектування і програмування сучасних додатків на базі технології клієнт-сервер з використанням компонентно-об'єктної моделі та бібліотеки шаблонів.

### Завдання навчальної дисципліни:

вивчення засобів розробки програмного забезпечення для роботи під керівництвом ОС Windows з широким використанням можливостей об'єктно-орієнтованого програмування, правила будовання програмних засобів в середовищах візуального програмування,

налагодження програм та розв'язання типових задач; вивчення принципів об'єктно-орієнтованого програмування (інкапсуляція, спадкування і поліморфізм); оволодіння навичками розробки і реалізації типових додатків з архітектурою клієнт-сервер, які знаходяться на локальному комп'ютері або в мережі; засвоєння основ застосування під час розробки об'єктів бібліотеку шаблонів, яка полегшує процес розробки програмного забезпечення і підвищує рівень безпечності коду.

Згідно з вимогами освітньо-професійної програми студенти повинні досягти таких **загальних компетентностей**:

- Здатність застосовувати знання у практичних ситуаціях (ЗК 2)
- Знання та розуміння предметної області та професійної діяльності (ЗК 3).
- Здатність до абстрактного мислення, аналізу та синтезу на відповідних рівнях (ЗК 4).
- Здатність вчитися і оволодівати сучасними знаннями (ЗК 6).
- Базові знання науково-методичних основ і стандартів в області інформаційних технологій, уміння застосовувати їх під час розробки та інтеграції систем, продуктів і сервісів інформаційних технологій (ЗК 17).
- Навички використання інформаційних і комунікаційних технологій (ЗК 21).

Згідно з вимогами освітньо-професійної програми студенти повинні досягти таких **фахових компетентностей**:

- Ґрунтовна підготовка в області програмування, володіння алгоритмічним мисленням, методами програмної інженерії для реалізації програмного забезпечення з урахуванням вимог до його якості, надійності, виробничих характеристик (ФК 2).

### **Програмні результати навчання.**

Відповідно до освітньої програми студент повинен досягти наступних програмних результатів:

- Знання сучасних методів побудови та аналізу ефективних алгоритмів і вміння їх реалізувати в конкретних застосуваннях (ПРН 4).
- Знання принципів структурного програмування, сучасних процедурно-орієнтованих мов, основних структур даних і вміння їх застосовувати під час програмної реалізації алгоритмів професійних завдань (ПРН 9).
- Здатність до об'єктно-орієнтованого мислення, знання об'єктно-орієнтованих мов програмування та уміння застосовувати об'єктно-орієнтований підхід під час проектуванні складних програмних систем (ПРН 10).
- Знання сучасних технологій та інструментальних засобів розробки програмних систем, уміння їх застосовувати на всіх етапах життєвого циклу (ПРН 11).

### **Міждисциплінарні зв'язки:**

для вивчення дисципліни «Об'єктно-орієнтоване програмування (мова C#)» необхідно володіти запасом знань таких дисциплін, як «Програмування та алгоритмічні мови», «Методи обчислень», а в подальшому знання з дисципліни «Об'єктно-орієнтоване програмування (мова C#)» стануть основою для вивчення таких дисциплін, як «Операційні системи», «Організація баз даних», «Ознайомча практика», «Обчислювальна геометрія та комп'ютерна графіка», «Організація та обробка електронної інформації», «Паралельні та розподілені обчислювання», «Платформи корпоративних інформаційних систем» та ін.

### 3. Програма навчальної дисципліни

#### Модуль 1.

#### Змістовний модуль 1. Структурне і процедурне програмування мовою C#.

##### Тема 1. Предмет вивчення і задачі дисципліни.

Предмет і об'єкт дослідження, цілі і задачі дисципліни «Об'єктно-орієнтоване програмування (мова C#)». Основні історичні етапи розвитку інформатики та програмування. Ознайомлення з інтегрованим середовищем програмування Visual Studio. Основні принципи структурного програмування. Складання алгоритму рішення задачі шляхом її декомпозиції на основі спадної і/або висхідної концепцій. Приклади побудови блок-схем.

##### Тема 2. Стандартні типи даних. Вирази.

Алфавіт мови Visual C#. Структура програми мовою Visual C#. Коментарі. Вбудовані типи даних. Декларація та використання констант і змінних в програмі. Перетворення типів. Процедури найпростішого безформатного введення даних з клавіатури і виведення на екран. Формати введення/виведення даних.

Визначення виразу в Visual C#. Арифметичні, логічні та символічні вирази. Основні операції в виразах і їх пріоритет. Оператор присвоювання. Ініціалізація даних в програмі. Директиви компілятора.

##### Тема 3. Розгалуження алгоритму.

Умовний оператор `if` (укорочена та повна форми). Мітки й оператори переходу `goto`. Розгалуження обчислювального процесу з використанням оператора `switch`.

##### Тема 4. Оператори циклічної обробки даних. Обробка масивів. Файли даних.

Циклічні оператори (`for`, `foreach`, `while`, `do while`). Організація циклів з передумовою та умовою в кінці тіла циклу. Приклади циклічних алгоритмів. Оператори виходу із циклу (`break`, `continue`). Застосування різних видів циклу під час знаходження суми безкінечного ряду.

Визначення масивів. Способи декларації масивів (в операторах об'яви типу). Ініціалізація масиву. Розташування елементів у пам'яті ЕОМ. Обробка одновимірних і багатовимірних масивів. Типові алгоритми пошуку екстремумів в масивах. Впорядкування масивів.

Файлові типи даних. Стандартні файли і пристрої. Стандартні процедури для роботи з файлами. Файли послідовної організації і доступу. Текстові файли. Файли прямого доступу.

##### Тема 5. Стандартні функції. Функції користувача.

Доцільність використання функцій. Декларація стандартних функцій в програмі. Принцип локалізації імен. Формальні та фактичні параметри, правила їх співвідношення. Механізм передачі параметрів. Оператори виклику функцій. Приклади використання стандартних функцій. Глобальні функції.

Декларація і визначення функцій користувача в програмі. Спільні риси та відмінності стандартних функцій і функцій користувача. Правила виклику функцій. Повернення результату роботи функцій. Механізм передачі параметрів. Приклади написання функцій користувача.

##### Тема 6. Похідні типи даних (структури).

Необхідність існування похідних типів (структур). Декларація похідних типів даних в програмі. Доступ до елементів похідних типів. Захист елементів похідних типів. Використання похідних типів в інженерних задачах.

## **Модульний контроль**

### **Змістовний модуль 2. Об'єктно-орієнтоване програмування.**

#### **Тема 7. Принцип інкапсуляції. Класи.**

Доцільність використання класів. Структура класу. Принципи об'єктно-зорієнтованого програмування (інкапсуляція, успадкування і поліморфізм). Декларація і визначення класів. Інтерфейси. Визначення і декларація типу. Поняття об'єкту. Доступ до методів класу. Властивості і їх визначення та поведінка. Сектори доступу до елементів класу. Принцип інкапсуляції.

#### **Тема 8. Принцип успадкування класу.**

Успадкування методів та змінних класів. Побудова дерев успадкування. Принцип повторного використання коду. Доступ до членів батьківського класу.

#### **Тема 9. Принцип поліморфізму.**

Визначення поліморфної поведінки об'єктів. Конкретний і класичний поліморфізм. Використання поліморфізму для написання компактного коду. Поліморфізм на основі базового класу. Використання абстрактних класів. Поліморфізм на основі інтерфейсів. Поліморфізм на основі виклику методів.

#### **Тема 10. Програмування на основі інтерфейсів.**

Розробка консольних застосунків мовою Visual C# для обчислювання виразів різного типу із застосуванням інтерфейсного програмування. Створення ієрархій інтерфейсів, що реалізують обробку виразів. Програмування класів, які реалізують інтерфейси. Створення об'єктів класу, який підтримує інтерфейси. Використання об'єктів інтерфейсів за допомогою оператора `is`. Використання об'єктів інтерфейсів за допомогою оператора `as`.

#### **Тема 11. Внутрішні класи.**

Визначення внутрішніх (вкладених) класів. Доступ до членів внутрішніх класів. Принципи використання внутрішніх класів.

#### **Тема 12. Перевантаження операцій.**

Синтаксис перевантаження операторів. Перевантаження унарних операторів (інкремент, декремент). Перевантаження бінарних операторів. Аргументи та результати, що повертаються.

#### **Тема 13. Масиви та колекції.**

Правила оголошення масивів, створення об'єктів класу `System.Array`. Використання методів і властивостей об'єктів масивів. Ініціалізація елементів масивів і використання одновимірних і багатовимірних масивів C#. Вивчення функціонування оператора `foreach` для обробки колекцій.

#### **Тема 14. Виключення.**

Визначення виключень. Правила обробки виключень. Структура, призначення і правила використання блоків `try / catch / finally`. Об'єкти виключень. Виклики стандартних, призначених для користувача і неспеціфікованих виключень. делегатів, як засоби програмування подій на мові C #; освоєння правил оголошення делегатів і створення екземплярів делегатів

#### **Тема 15. Делегати.**

Визначення делегатів. Правила використання делегатів. Делегатів, як засоби програмування подій мовою C#. Правил оголошення делегатів і створення екземплярів делегатів.

## Тема 16. Події.

Визначення події Windows. Правила використання подій. Утворення методів обробки подій. Використання майстрів інтегрованого середовища для програмування функцій-обробників подій. Стандартні події Windows. Події, визначені користувачем.

### Модульний контроль.

## 4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
л		п	лаб	інд	с.р.	
1	2	3	4	5	6	7
<b>Модуль 1.</b>						
<b>Змістовний модуль 1. Структурне і процедурне програмування мовою C#</b>						
Тема 1. Вступ до навчальної дисципліни	2	2				
Тема 2. Стандартні типи даних. Вирази	9	2	3			4
Тема 3. Оператори розгалуження алгоритму	9	2	3			4
Тема 4. Оператори циклічної обробки даних. Обробка масивів. Файли даних.	8	2	2			4
Тема 5. Стандартні функції. Функції користувача.	8	2	2			4
Тема 6. Похідні типи даних (структури).	8	2	2			4
Модульний контроль.	4					4
Разом за змістовим модулем 1	<b>48</b>	<b>12</b>	<b>12</b>			<b>24</b>
<b>Змістовний модуль 2. Об'єктно-орієнтоване програмування</b>						
Тема 7. Принцип інкапсуляції. Класи.	7	2	2			3
Тема 8. Принцип успадкування класу.	7	2	2			3
Тема 9. Принцип поліморфізму.	7	2	2			3
Тема 10. Програмування інтерфейсів.	7	2	2			3
Тема 11. Внутрішні класи.	7	2	2			3
Тема 12. Перевантаження операцій.	7	2	2			3
Тема 13. Масиви та колекції.	7	2	2			3
Тема 14. Виключення.	7	2	2			3
Тема 15. Делегати.	7	2	2			3
Тема 15. Події.	7	2	2			3
Модульний контроль.	2					2
Разом за змістовим модулем 2	<b>72</b>	<b>20</b>	<b>20</b>			<b>32</b>
Разом за модулем 1	<b>120</b>	<b>32</b>	<b>32</b>			<b>56</b>
Контрольний захід – екзамен						
<b>Усього годин навчальної дисципліни</b>	<b>120</b>	<b>32</b>	<b>32</b>			<b>56</b>

## 5. Теми семінарських занять

№ з/п	Назва теми	Кількість годин
1		
2		
...		

### 6. Теми практичних занять

№ з/п	Назва теми	Кількість годин
1.	Вирази	3
2.	Умовні оператори	3
3.	Оператори циклу	2
4.	Функції	2
5.	Похідні типи (структури)	2
6.	Принцип інкапсуляції. Класи	2
7.	Принцип успадкування класу	2
8.	Принцип поліморфізму	2
9.	Програмування інтерфейсів	2
10.	Внутрішні класи	2
11.	Перевантаження операцій	2
12.	Масиви та колекції	2
13.	Виключення	2
14.	Делегати	2
15.	Події	2
	<b>Разом</b>	<b>32</b>

### 7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1		
2		
...		

### 8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1.	Вирази	4
2.	Умовні оператори	4
3.	Оператори циклу	4
4.	Функції	4
5.	Похідні типи (структури)	4
6.	<i>Модульний контроль</i>	4
7.	Принцип інкапсуляції. Класи	3
8.	Принцип успадкування класу	3
9.	Принцип поліморфізму	3
10.	Програмування на основі інтерфейсів	3
11.	Внутрішні класи	3
12.	Перевантаження операцій	3
13.	Масиви та колекції	3
14.	Виключення	3
15.	Делегати	3
16.	Події	3
17.	<i>Модульний контроль</i>	2
	<b>Разом</b>	<b>56</b>



## 9. Індивідуальні завдання

*Виконання розрахункової роботи на тему «Програмування на основі інтерфейсів».*

1. Розробити консольну програму мовою Visual C# для обчислювання виразів різного типу із застосуванням інтерфейсного програмування. Створити ієрархію інтерфейсів, що реалізують обробку виразів, причому

- інтерфейс IRead визначає введення вихідних даних;
- інтерфейс IReadWrite розширює інтерфейс IRead можливістю виведення результатів на екран;
- інтерфейс IArithm визначає розрахунок арифметичного виразу;
- інтерфейс IBool визначає розрахунок логічного виразу;
- інтерфейс IExpression комбінує функціональність інтерфейсів і додає можливість розрахунку строкового вираження;

клас *CExpression*

- реалізує інтерфейси IReadWrite, IExpression;
- має конструктор без параметрів;
- реалізує відкриту функцію строкового представлення об'єкту класу *CExpression*.

2. Створити об'єкт класу *CExpression* для заданих вихідних даних. Використовуючи об'єкт класу ввести вихідні дані.

3. Використовуючи об'єкти інтерфейсів IArithm, IBool і IExpression, за допомогою оператора `is` обчислити арифметичне, логічне і строкове вираження. Використовуючи об'єкт інтерфейсу IReadWrite, за допомогою оператора `as` вивести результати розрахунків.

Обсяг роботи – 15 сторінок.

## 10. Методи навчання

1. Пояснювально-ілюстративний (інформаційно-рецептивний) метод та метод проблемного виконання (лекційні заняття).
2. Репродуктивний (лабораторні роботи).
3. Частково-пошуковий (евристичний) та дослідницький (самостійна робота та виконання розрахункової та розрахунково-графічної робіт).
4. Дисципліна «Об'єктно-орієнтоване програмування (мова C#)» передбачає лекційні (в т. ч. з використанням мультимедійного обладнання) і лабораторні заняття під керівництвом викладача та самостійну роботу студента за підручниками і матеріалами, опублікованими кафедрою (методичні посібники і мережеві ресурси), що забезпечує закріплення теоретичних знань, сприяє набуттю практичних навичок і розвитку самостійного наукового мислення. Передбачено регулярні індивідуальні консультації.

## 11. Методи контролю

Визначення рівня засвоєння студентом навчального матеріалу дисципліни здійснюється шляхом проведення поточних (захист лаб. робіт, поточні контрольні з теоретичного матеріалу) і підсумкових контролів (захист змістовного модуля, екзамен).

У завдання поточного контролю входить систематична перевірка розуміння й засвоєння студентом програмного матеріалу, виконання практичних і лабораторних робіт, уміння самостійно проробляти тексти складання конспектів, написання звітів, здатності усно або письмово представляти певний матеріал.

Перед підсумковим контролем ставиться завдання перевірки глибини засвоєння студентом програмного матеріалу дисципліни, логіки й взаємозв'язки між її окремими розділами, здатності творчо використати придбані знання, уміння сформулювати своє відношення до проблеми, що впливає зі змісту дисципліни.

## 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 завдання	Кількість завдань	Сумарна кількість балів
<b>Модуль</b>			
Виконання і захист лабораторних робіт (змістовий модуль 1)	3...5	6	18...30
Модульний контроль	10...15	1	6...10
Виконання і захист лабораторних робіт (змістовий модуль 2)	3...5	9	27...45
Модульний контроль	10...15	1	9...15
<b>Усього за семестр</b>			<b>60...100</b>

**Семестровий контроль (іспит) проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до іспиту. Під час складання семестрового іспиту студент має можливість отримати максимум 100 балів.**

Білет для іспиту/заліку складається з одного теоретичного і двох практичних питань. В першому питанні студент повинен продемонструвати теоретичні знання. У другому питанні – показати навички складання і виконання програми на основі принципів процедурного програмування із застосуванням функцій і файлів. У третьому пункті – продемонструвати знання зі створення і застосування програми з використанням класів і використання графічного інтерфейсу.

Складові білету	Складові оцінки	Бали за одне питання	Сумарна кількість балів
Пункт 1.	дано визначення наведеного поняття	5	20
	наведено приклади	5	
	наведено фрагменти тексту	10	
Пункт 2.	складено алгоритм розрахунку (блок-схема)	10	30
	створено проект консольного застосунку	10	
	отримано результати в консольному вікні	10	
Пункт 3.	складено алгоритм розрахунку (блок-схема)	10	50
	створено проект застосунку з графічним інтерфейсом	10	
	отримано результати у вікні-формі	30	
<b>Ітогова оцінка за іспит</b>			<b>100</b>

## 12.2. Якісні критерії оцінювання

Необхідний обсяг знань для одержання позитивної оцінки:

**студент повинен знати:**

- принципи процедурного і об'єктно-орієнтованого програмування;
- як створити проект консольного застосунку;
- як створити проект застосунку з графічним інтерфейсом;
- як використовувати вбудований у середовище інструмент відлагодження застосунку;
- способи декларації, визначення і утворення об'єктів класів і інтерфейсів;
- як утворювати алгоритм розрахунку у вигляді блок-схеми;
- як використовувати принципи захисту елементів класів від несанкційованого доступу;
- як створювати похідні класи і перевизначати віртуальні функції;
- як створювати і використовувати статичні функції класів;
- як неполіморфно і поліморфно викликати функції класів;
- як обробляти події і програмувати обробники подій;
- основні спільні риси та відмінності класів і інтерфейсів;
- як використати можливості діаграми класів.

Необхідний обсяг вмінь для одержання позитивної оцінки:

**повинен вміти:**

- виконувати поставлене завдання у вигляді консольної або графічної аплікації;
- використовувати можливості інтегрованого середовища програмування для відлагодження застосунків;
- алгоритмізувати поставлене завдання у вигляді алгоритму розрахунку (блок-схеми);
- використовувати стандартні простори імен для доступу до стандартних бібліотек класів;
- визначати клас і закривати доступ до елементів класу за допомогою кваліфікаторів доступу;
- використовувати відкриті властивості для доступу до закритих змінних класу;
- декларувати інтерфейси і розширювати можливості класу за допомогою інтерфейсів.

## 12.3 Критерії оцінювання роботи студента протягом семестру.

**Приклад.**

**Задовільно (60-74).** Показати мінімум теоретичних знань та практичних умінь. Виконати всі лабораторні роботи за індивідуальним варіантом із застосуванням консольної аплікації. Захистити всі теми індивідуальних завдань та модулів. Знати як утворити алгоритм розрахунку у вигляді блок-схеми. Виконати розрахункову роботу хоча б у вигляді консольного застосунку і захистити отримані результати розрахунку.

**Добре (75-89).** Показати знання основних теоретичних питань та практичних умінь. Виконати всі лабораторні роботи за індивідуальним варіантом із застосуванням аплікації з графічним інтерфейсом і використанням класів. Захистити всі теми індивідуальних завдань та модулів на оцінку «добре». Знати як утворити алгоритм розрахунку у вигляді блок-схеми і створювати каркас класу за допомогою діаграми класів. Виконати розрахункову роботу у вигляді консольного застосунку але з використанням класів і інтерфейсів, і захистити отримані результати розрахунку.

**Відмінно (90-100).** Показати тверде і досконале знання всіх теоретичних питань та практичних умінь. Виконати всі лабораторні роботи за індивідуальним варіантом із застосуванням аплікації з графічним інтерфейсом і використанням класів. Захистити всі теми індивідуальних завдань та модулів на оцінку «відмінно». Знати як утворити алгоритм розрахунку у вигляді блок-схеми і створювати каркас класу за допомогою діаграми класів.

Вчасно виконати розрахункову роботу у вигляді застосунку з графічним інтерфейсом, з використанням класів і інтерфейсів, і захистити отримані результати розрахунку з оцінкою «відмінно».

### 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

1. Скоб, Ю.О. Основи програмування Windows мовою С# [Текст] : навч. посібник до лаб. робіт / Ю.О. Скоб, М.Л. Угрюмов, К.П. Коробчинський. – Харків : Нац. аерокосм. ун-т «Харк. авіац. ін-т», 2010. – 108 с.
2. Скоб, Ю.О. Основи інформаційних технологій та програмування [Текст] : навч. посібник до лаб. практикуму / Ю.О. Скоб, О.В. Патокіна, О.В. Халтурін. – Харків : Нац. аерокосм. ун-т «Харк. авіац. ін-т», 2005. – 77 с.
3. Скоб, Ю.О. Теорія програмування [Текст] : навч. посібник до лаб. практикуму / Ю.О. Скоб, О.В. Патокіна, В.О. Халтурін. – Харків : Нац. аерокосм. ун-т «Харк. авіац. ін-т», 2005. – 73 с.
4. Скоб, Ю.О. Сучасні технології програмування [Текст] : навч. посібник до лаб. практикуму / Ю.О. Скоб. – Харків : Нац. аерокосм. ун-т «Харк. авіац. ін-т», 2006. – 100 с.
5. Соколов, О.Ю. Інформатика для інженерів [Текст] / О.Ю. Соколов, І.Т. Зарецька, Г.М. Жолткевич, О.В. Ярова. – Харків : Факт, 2005. – 423с.
6. Скоб, Ю.О. Програмування елементів управління ActiveX засобами бібліотеки активних шаблонів [Текст] : навч. посібник до кур. проекту / Ю.О. Скоб, М.Л. Угрюмов, К.П. Коробчинський. – Харків : Нац. аерокосм. ун-т «Харк. авіац. ін-т», 2006. – 108 с.

### 14. Рекомендована література

#### Базова література

1. Соколов, О.Ю. Інформатика [Текст] : конспект лекцій / О.Ю. Соколов, М.Л. Угрюмов, В.О. Халтурін, Ю.К. Чернишев. – Харків : ХАІ ім. М. Є. Жуковського, 2003. – 191 с.
2. Турчак Л.І. Основи чисельних методів [Текст] : навч. посібник / Л.І. Турчак. – М. : Наука, Гол.ред.фіз.-мат.літ., 1987. – 320с.
3. Нікітченко М. С. Теоретичні основи програмування [Текст]: навчальний посібник / М.С. Нікітченко — Ніжин : Видавництво НДУ імені Миколи Гоголя, 2010. — 121с.
4. Яровая, О.В. Численні методи з використанням математичних пакетів MathCAD, МАВ [Текст] : навч. посібник до лаб. практикуму. Ч. 1 / О.В. Яровая, А.В. Патокіна. – Харків : ХАІ, 2008.- 87с.
5. Яровая, О.В. Численні методи з використанням математичних пакетів MathCAD, МАВ [Текст] : навч. посібник до лаб. практикуму. Ч. 2 / О.В. Яровая, А.В. Патокіна. – Харків : ХАІ, 2012. – 98с.
6. Михайленко, С.В. Прикладна математика [Текст] : лаб. практикум із чисельних методів / С.В. Михайленко. – Харків : ХАІ ім. М. Є. Жуковського, 1992. – 102с.
7. Баргеньев О.В. Сучасний Фортран [Текст]. – М.: Діалог-МІФІ, 2005. – 560 с.

8. Зубенко В. В. Програмування [Текст] : навчальний посібник (гриф МОН України) / В. В. Зубенко, Л. Л. Омельчук. — К. : ВПЦ «Київський університет», 2011. — 623 с.
9. Lawrence Norman. раq Visual Visual C#: A Guide to Creating Windows Applications [Текст]. — Boston: Digital Press, 2002. — 462 p.
10. Рижиков Ю.І. Сучасний Фортран [Текст]. — СПб.: КОРОНА принт, 2004. — 288 с.
11. Бартенєв О.В. Графіка OpenGL: програмування на Фортрані [Текст]. — М.: Діалог-МІФІ, 2000. — 368 с.
12. Бартенєв О.В. Фортран для професіоналів. Математична бібліотека IMSL [Текст] : Ч. 1-3. — М.: Діалог-МІФІ, 2005. — 448 с.
13. Бартенєв О.В. Фортран для студентів [Текст]. — М.: Діалог-МІФІ, 1999. — 400 с.
14. Бухтіяров А.М., Малікова Ю.П., Фролов Г.Д. Практикум з програмування на фортрані (ОС ЄС ЕОМ) [Текст] : Навч. посібник для вузів. — М.: Наука, Гл. ред. фіз.-мат. літ., 1988. — 228 с.
15. Фортран 77 ЄС ЕОМ [Текст] / З.С. Брич, О.Н. Гулецька, Д.В. Капілевич та інш. — М.: Фінанси та статистика, 1989. — 351 с.
16. Очков В.Ф. Mathcad 7 Pro для студентів та інженерів [Текст] . — М.: Комп'ютерпрес, 1998. — 384с.
17. Потьомкін В.Г. Система МАВ [Текст]: Довідковий посібник. — М.: Діалог-МІФІ, 1997. — 350с.
18. Демідович Б.П., Марон І.А. Основи обчислювальної математики [Текст]. М., 1963.

### Допоміжна література

1. Немнюгін С.А., Стесик О.Л. Сучасний Фортран. Самовчитель [Текст] . — СПб.: БХВ-Петербург, 2004. — 496 с.
2. Алгазін С.Д., Кондратьєв В.В. Програмування на Visual Visual C# [Текст] . - М.: Діалог-МІФІ, 2008.- 472 с.
3. Артёмов І.Л. Visual C# Основи програмування [Текст] .- М.: Діалог-МІФІ, 2007.- 304 с.
4. Звягін В.Ф., Яньшина Н.А., Голиничев В.Н. Практикум з сучасного Фортрану в курсі Інформатики [Текст]: Навчальний посібник. - СПб: ІТМО, 2010.- 134 с.
5. Мак-Кракен Д., Дорн У. Чисельні методи та програмування на ФОРТРАНі [Текст].-М.: Мир, 1977.- 584 с.
6. Фаронов В.В. VISUAL C# 5. Навчальний курс [Текст]:.-М.: Нолидж, 2000.-608 с.
7. Charman Stephen J. Visual C# 95/2003 for Scientists & Engineers [Text]: McGraw-Hill, 2007. — 682 p.

### 15. Інформаційні ресурси

1. <http://www.twirpx./files/informatics/languages/Visual C#/> - Книги мовою C#.
2. <http://progbook.ru/Visual C#/> - Книги на язьке C#.
3. [www.Visual C#-online.info/](http://www.Visual C#-online.info/) - Мова програмування Фортран (Visual C#).
4. <ftp://79.110.128.93/books/programming/Visual C#/Bartenev-Visual C#.pdf> - Бартенєв О.В. Сучасний C#.
5. <http://citforum.ru/> – Форум. Новини, статті, розсилки, форуми.
6. <http://programming.in.ua/>- Програмування українською.
7. <http://pers.narod.ru/study/methods/> - Навчання. Лекції з чисельних методів.