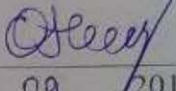


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ АВТОМОБІЛЬНО-ДОРОЖНИЙ
УНІВЕРСИТЕТ

Кафедра комп'ютерних технологій і мехатроніки

«ЗАТВЕРДЖУЮ»

Гарант освітньо-професійної програми
«Програмне забезпечення систем»
першого (бакалаврського) рівня вищої
освіти, завідувач кафедри КТМ, д.т.н.,

професор  Ніконов О.Я.
« 03 » 09 / 2019р.

СИЛАБУС
АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ /
ALGORITHMIZATION AND PROGRAMMING
SYLLABUS

освітній ступінь	бакалавр / bachelor
галузь знань	12 Інформаційні технології / Information Technology
спеціальність	121 Інженерія програмного забезпечення / Software Engineering
освітня програма	Програмне забезпечення систем / Systems Software

Харків 2019

Автор: Тімонін Володимир Олексійович, кандидат технічних наук,
старший науковий співробітник, доцент кафедри комп'ютерних технологій і
мехатроніки

Силабус розглянуто та затверджено на засіданні кафедри комп'ютерних
технологій і мехатроніки, протокол № 18 від «27» червня 2019 р.

СИЛАБУС

АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ /

ALGORITHMIZATION AND PROGRAMMING

SYLLABUS

освітній ступінь	бакалавр / bachelor
галузь знань	12 Інформаційні технології / Information Technology
спеціальність	121 Інженерія програмного забезпечення / Software Engineering
освітня програма	Програмне забезпечення систем / Systems Software

ВСТУП

Складовою частиною навчальних дисциплін спеціальності 121 «Інженерія програмного забезпечення» є вивчення алгоритмічної мови високого рівня C#, що дозволяє студентам оволодіти принципами структурного, процедурного і об'єктно-орієнтованого програмування, способами представлення і запису даних в алгоритмічних мовах, методами і способами алгоритмізації та програмування професійних задач на мовах високого рівня.

Набування студентами практичних навичок у розв'язанні задач на мові C# забезпечить їм уміння у вирішенні професійних задач, пов'язаних з програмуванням і моделюванням систем автоматизованого управління технічними та виробничими процесами з використанням сучасних технологій розробки програмного забезпечення.

Програма вивчення навчальної дисципліни «**Алгоритмізація та програмування**» складена відповідно до освітньо-кваліфікаційної характеристики та навчального плану підготовки **бакалавра спеціальності 121 «Інженерія програмного забезпечення»**.

Викладання дисципліни розраховано на виконання студентами практичних робіт та самостійних завдань у програмному середовищі MS Visual Studio.

1. МЕТА, ПРЕДМЕТ ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

1.1. **Метою** вивчення навчальної дисципліни є викладення основних понять алгоритмізації і техніки застосування у програмуванні базових алгоритмічних структур (організація програм) і базових структур даних (організація даних).

1.2. **Предметом** вивчення навчальної дисципліни є теорія і практика застосування у програмуванні базових алгоритмічних структур і базових структур даних на основі сучасних технологій розробки програмного забезпечення.

1.3. **Основними завданнями** вивчення навчальної дисципліни є формування у студентів необхідних знань у галузі сучасних технологій програмування; ознайомлення з технічними, алгоритмічними, програмними і технологічними рішеннями у області програмування; вироблення практичних навичок аналітичного та експериментального дослідження основних методів і засобів, що використовуються в області програмування при рішенні обчислювальних задач.

1.4. **По завершенні вивчення дисципліни студенти повинні:**

знати:

етапи підготовки і рішення задач на ПК; типові конструкції алгоритмів; принципи структурного та процедурного програмування; особливості використання операторів управління програмою; особливості обробки базових і складних типів даних; правила використання функцій; принципи побудови вводу-виводу даних; типові алгоритми рішення обчислювальних задач.

вміти:

розробляти алгоритми рішення задач; розробляти програми на алгоритмічній мові високого рівня для обробки різних типів даних; використовувати функції при розробці складних програм; виконувати налагодження програм на ПК за допомогою вбудованих інструментальних засобів; використовувати алгоритмічні мови високого рівня для рішення типових професійних задач; проектувати компоненти програмного забезпечення; проектувати людино-машинний інтерфейс інформаційних систем.

1.5. Міждисциплінарні зв'язки:

Вивченню дисципліни:

передують: “Основи інформатики” середньою школи, “Вища математика”, “Дискретна математика”, “Теорія алгоритмів”;

потребують її вивчення: “Об'єктно-орієнтоване програмування”, “Організація баз даних і знань”, “Веб-технології та веб-дизайн”, “Крос-платформне програмування”, “Технологія створення програмних продуктів”, “Технологія розподілених систем та паралельних обчислень”, “Методи та системи штучного інтелекту”, “Проектно-технологічна практика”, “Дипломне проектування”.

2. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Характеристика навчальної дисципліни
Кількість кредитів - 8 Кількість годин - 240	<u>нормативна</u> (нормативна, за вибором ВНЗ, за вибором студента)
Семестр викладання дисципліни	2
Вид контролю:	<u>екзамен</u> (залік, екзамен)
Розподіл часу:	
- лекції (годин)	32
- практичні, семінарські (годин)	-
- лабораторні роботи (годин)	48
- самостійна робота студентів (годин)	130
- курсовий проект (годин)	-
- курсова робота (годин)	-
розрахунково-графічна робота (контрольна робота)	-
підготовка та складання екзамену	30

3. РОЗПОДІЛ ДИСЦИПЛІНИ У ГОДИНАХ ЗА ФОРМАМИ ОРГАНІЗАЦІЇ ОСВІТНЬОГО ПРОЦЕСУ ТА ВИДАМИ НАВЧАЛЬНИХ ЗАНЯТЬ

Результати навчання	Навчальна діяльність	Робочий час студента (год.)	Оцінювання (бал)
1	2	3	4
Тема 1. Архітектура комп'ютерів, принципи фон Неймана.			
Знати архітектуру комп'ютерів, принципи фон Неймана; технічні характеристики складових частин ПК.	Лекція №1. Архітектура комп'ютерів, принципи фон Неймана. <i>Основні питання:</i> 1. Архітектура комп'ютера. 2. Принципи фон Неймана. 3. Етапи розвитку ЕОМ. 4. Функції складових частин ПК. <i>Список рекомендованих джерел:</i> Основний – 5, 6	2	
	Самостійна робота студента (СРС ¹). Технічні характеристики ПК. <i>Основні питання:</i> 1. Архітектура комп'ютера. 2. Принципи фон Неймана. 3. Етапи розвитку ЕОМ. 4. Технічні характеристики складових частин ПК.	6	
Тема 2. Поняття алгоритму та типові алгоритмічні структури програмування.			
Знати етапи підготовки і рішення задач на ПК; типові конструкції алгоритмів; вимоги ДСТУ щодо оформлення технічної документації на програмне забезпечення; етапи	Лекція №2. Алгоритмізація обчислювальних процесів. <i>Основні питання:</i> 1. Етапи розробки обчислювальних задач 2. Способи подання алгоритмів 3. Основні елементи графічного способу 4. Етапи налагодження програм <i>Список рекомендованих джерел:</i> Основний – 6	2	
	Практична робота №1 (ПР1). Дослідження можливостей	4	0...100

¹ Завданнями для самостійної роботи є вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем, а також в практичному відпрацюванні незавершених завдань в практичних роботах.

<p>налагодження програм; засоби розробки програмного забезпечення.</p> <p>Вміти розробляти блок-схеми різних видів алгоритмів; використовувати засоби розробки для створення програмного забезпечення.</p>	<p>інтегрованого середовища розробки Visual C # для створення консольних додатків. <i>Мета роботи - дослідити можливості інтегрованого середовища розробки Visual Studio 2017 і отримати практичні навички зі створення консольних додатків ОС Windows.</i></p> <p>План заняття:</p> <ol style="list-style-type: none"> 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань. 		
	<p>Лекція №3. Види алгоритмів. <i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Лінійні алгоритми. 2. Алгоритми, що розгалужуються. 3. Циклічні алгоритми. 4. Комбіновані алгоритми. <p><i>Список рекомендованих джерел:</i> Основний – 1, 6</p>	2	
	<p>СРС. Алгоритмізація обчислювальних процесів. <i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Етапи розробки обчислювальних задач 2. Способи подання алгоритмів 3. Основні елементи графічного способу. 4. Етапи налагодження програм. 5. Лінійні алгоритми. 6. Алгоритми, що розгалужуються. 7. Циклічні алгоритми. 8. Комбіновані алгоритми. 	8	
<p>Тема 3. Елементи алгоритмічних мов: концепція типів даних, імена, значення, покажчики, змінні, константи, операції, вирази.</p>			
<p>Знати типи даних мови С#; особливості обробки базових і</p>	<p>Лекція №4. Основні поняття мови С #. <i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Основні поняття мови С #. 2. Типи даних. 	2	

<p><i>складних типів даних; принципи побудови вводу-виводу даних; оператори мови С#; можливості інтегрованого середовища розробки Visual С# для створення консольних додатків.</i></p> <p>Вміти <i>налаштувати інтегроване середовище розробки; розробляти програми з введення даних і виведення результатів; обробляти помилки; тестувати програми.</i></p>	<p>3. Змінні. 4. Перетворення і приведення типів.</p> <p><i>Список рекомендованих джерел:</i> Основний – 1, 2, 3, 6, 8, 9 Додатковий – 12 ÷ 17 Інтернет-ресурси – 18, 19</p>		
	<p>СРС. Інтегроване середовище розробки Visual Studio 2017.</p> <p><i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Структура головного вікна ІСР Visual Studio 2017. 2. Можливості редактора коду Visual С #. 3. Методика створення консольного застосування Visual С #. 4. Створення поліпшеного консольного додатку Visual С #. 	2	
	<p>ПР2. Дослідження можливостей інтегрованого середовища розробки Visual С# для створення консольних додатків. Введення-виведення даних в консольному додатку.</p> <p><i>Мета роботи - дослідити можливості інтегрованого середовища розробки Visual Studio 2017 і отримати практичні навички зі створення консольних додатків ОС Windows з введенням і виведенням даних.</i></p> <p>План заняття:</p> <ol style="list-style-type: none"> 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань. 	4	0...100
	<p>Лекція №5. Оператори мови С #.</p> <p><i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Оператор присвоювання. 2. Арифметичні оператори. 3. Логічні оператори. 	2	

	<p>4. Пріоритет операторів. <i>Список рекомендованих джерел:</i> Основний – 1, 2, 3, 6, 8, 9 Додатковий – 12 ÷ 17 Інтернет-ресурси – 18, 19</p>		
	<p>СРС. Характеристики вбудованих типів даних. <i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Цілочисельні типи. 2. Символьний тип. 3. Речові типи. 4. Логічний тип. 	4	
Тема 4 Структурне програмування: послідовність, розгалуження та цикли.			
<p><i>Знати принципи структурного та процедурного програмування; особливості використання операторів управління програмою; можливості інтегрованого середовища розробки Visual C# для створення консольних додатків різних видів алгоритмів.</i></p>	<p>Лекція №6. Лінійний алгоритм і його реалізація. <i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Структура програми. 2. Методика розробки програм. 3. Лінійний алгоритм. <p><i>Список рекомендованих джерел:</i> Основний – 1, 2, 3, 6, 8, 9 Додатковий – 12 ÷ 17</p>	2	
	<p>СРС. Методика розробки лінійного алгоритму. <i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Структура програми. 2. Методика розробки програм. 3. Лінійний алгоритм. 4. Оператор присвоювання. 5. Арифметичні оператори. 6. Перетворення і приведення типів. 	4	
<p><i>Вміти розробляти алгоритми рішення задач; розробляти програми на алгоритмічній мові високого рівня для різних видів алгоритмів; раціонально використовувати керуючі</i></p>	<p>ПР3. Дослідження можливостей інтегрованого середовища розробки Visual C# для створення додатків лінійної структури з використанням арифметичних операцій. <i>Мета роботи - дослідити можливості інтегрованого середовища розробки Visual Studio 2017 і отримати практичні навички зі створення додатків лінійної структури.</i></p> <p style="text-align: center;">План заняття:</p> <ol style="list-style-type: none"> 1. Актуалізація теоретичного матеріалу. 	4	0...100

<p><i>оператори при написання програм; виконувати налагодження програм на ПК за допомогою вбудованих інструментальних засобів.</i></p>	<p>2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків).</p> <p>3. Презентація і захист виконаної роботи.</p> <p>4. Тестування знань.</p>		
	<p>СРС. Функціонування логічних операцій.</p> <p><i>Основні питання:</i></p> <p>1. Оператори відносини.</p> <p>2. Логічні оператори.</p> <p>3. Порозрядні логічні оператори.</p>	8	
	<p>Лекція №7. Алгоритми, що розгалужуються, і їх реалізація.</p> <p><i>Основні питання:</i></p> <p>1. Базові конструкції структурного програмування.</p> <p>2. Оператор розгалуження if ... else.</p> <p>3. Оператор розгалуження switch.</p> <p>4. Тернарний оператор.</p> <p><i>Список рекомендованих джерел:</i></p> <p>Основний – 1, 2, 3, 6, 8, 9</p> <p>Додатковий – 12 ÷ 17</p> <p>Інтернет-ресурси – 18, 19</p>	2	
	<p>СРС. Методика розробки алгоритму, що розгалужуються.</p> <p><i>Основні питання:</i></p> <p>1. Базові конструкції структурного програмування.</p> <p>2. Оператор розгалуження if ... else.</p> <p>3. Оператор розгалуження switch.</p> <p>4. Тернарний оператор.</p>	6	
	<p>ПР4. Дослідження можливостей інтегрованого середовища розробки Visual C# для створення додатків з структур, що розгалужується.</p> <p><i>Мета роботи - дослідити можливості інтегрованого середовища розробки Visual Studio 2017 і отримати практичні навички зі створення додатків з структур, що розгалужується.</i></p> <p>План заняття:</p> <p>1. Актуалізація теоретичного</p>	4	0...100

	<p>матеріалу.</p> <ol style="list-style-type: none"> 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань. 		
	<p>Лекція №8. Циклічні алгоритми і їх реалізація.</p> <p><i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Оператор while. 2. Оператор do ... while. 3. Оператор for. 4. Оператор foreach. <p><i>Список рекомендованих джерел:</i></p> <p>Основний – 1, 2, 3, 6, 8, 9 Додатковий – 12 ÷ 17 Інтернет-ресурси – 18, 19</p>	2	
	<p>СРС. Види циклічних операторів і особливості їх функціонування.</p>	8	
	<p>ПР5. Дослідження можливостей інтегрованого середовища розробки Visual C# для створення додатків циклічної структури (оператори while, do ... while).</p> <p><i>Мета роботи - дослідити можливості інтегрованого середовища розробки Visual Studio 2017 і отримати практичні навички зі створення додатків циклічної структури з використанням операторів while і do ... while.</i></p> <p style="text-align: center;">План заняття:</p> <ol style="list-style-type: none"> 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань. 	4	0...100
	<p>СРС. Методика розробки циклічного</p>	4	

	<p>алгоритму</p> <p><i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Оператор while. 2. Оператор do ... while. 3. Оператор for. 4. Оператор foreach. 		
	<p>ПР6. Дослідження можливостей інтегрованого середовища розробки Visual C# для створення додатків циклічної структури (оператор for).</p> <p><i>Мета роботи - дослідити можливості інтегрованого середовища розробки Visual Studio 2017 і отримати практичні навички зі створення додатків циклічної структури з використанням операторів for.</i></p> <p>План заняття:</p> <ol style="list-style-type: none"> 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань. 	4	0...100
	<p>СРС. Види безумовних операторів і особливості їх функціонування.</p> <p><i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Оператор break. 2. Оператор continue. 3. Оператор return. 4. Оператор goto. 	8	
Тема 5. Процедурно-орієнтоване програмування. Рекурсія.			
<p>Знати правила використання функцій; можливості інтегрованого середовища розробки Visual C# для створення додатків, що використовують функції.</p>	<p>Лекція №9. Функції.</p> <p><i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Призначення функції. 2. Опис і виклик функції. 3. Передача параметрів. 4. Перевантаження функцій. <p><i>Список рекомендованих джерел:</i></p> <p>Основний – 2, 3, 6, 8, 9</p> <p>Додатковий – 12 ÷ 17</p> <p>Інтернет-ресурси – 18, 19</p>	2	
	<p>СРС. Особливості використання</p>	8	

Вміти розробляти програми, що використовують функції; використовувати рекурсивну функцію в розроблених додатках; використовувати функції при розробці складних програм.	функцій в програмуванні. <i>Основні питання:</i> 1. Призначення функції. 2. Опис і виклик функції. 3. Передача параметрів. 4. Перевантаження функцій.		
	ПР7. Дослідження можливостей інтегрованого середовища розробки Visual C# для створення додатків, що використовують функції. <i>Мета роботи - дослідити можливості інтегрованого середовища розробки Visual Studio 2017 і отримати практичні навички зі створення додатків, що використовують функції.</i> План заняття: 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.	4	0...100
	СРС. Розробка програми з рекурсією. <i>Основні питання:</i> 1. Призначення рекурсії. 2. Розробка додатка з рекурсією.	8	
Темаб. Методологія розробки програм: низхідне та висхідне проектування, модульне програмування.			
Знати принципи низхідного, висхідного та модульного програмування; концепції модульного програмування; різновиди модулів; характеристики модулів.	Лекція №10. Методологія розробки програм: низхідне та висхідне проектування, модульне програмування. <i>Основні питання:</i> 1. Цілі структурного програмування. 2. Програмування з використанням покрокової деталізації. 3. Низхідне та висхідне програмування. 4. Модульне програмування. <i>Список рекомендованих джерел:</i> Основний – 6	2	

	<p>СРС. Характеристика технологій програмування.</p> <p><i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Концепції модульного програмування. 2. Визначення модуля і його приклади. 3. Різновиди модулів. 4. Характеристики модуля. 	6	
Тема 7. Організація даних (масиви, рядки, структури) та алгоритми їх оброблення.			
<p>Знати особливості роботи з масивами, рядками, покажчиками; можливості інтегрованого середовища розробки Visual C# для створення додатків по обробці масивів даних, рядків.</p> <p>Вміти розробляти програми по обробці масивів даних, рядків; створювати додатки, що використовують покажчики; виконувати налагодження програм на ПК за допомогою вбудованих інструментальних засобів.</p>	<p>Лекція №11. Масиви.</p> <p><i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Одномірні масиви. 2. Ініціалізація масивів. 3. Багатовимірні масиви. 4. “Рвані” масиви. <p><i>Список рекомендованих джерел:</i> Основний – 2, 3, 6, 8, 9 Додатковий – 12 ÷ 17 Інтернет-ресурси – 18, 19</p>	2	
	<p>СРС. Розробка алгоритму обробки одновимірних масивів.</p> <p><i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Оголошення одновимірних масивів. 2. Оператори, що використовуються для обробки масивів. 	6	
	<p>ПР8. Дослідження можливостей інтегрованого середовища розробки Visual C# для створення додатків по обробці одновимірних масивів даних.</p> <p><i>Мета роботи - дослідити можливості інтегрованого середовища розробки Visual Studio 2017 і отримати практичні навички зі створення додатків, що використовують функції.</i></p> <p style="text-align: center;">План заняття:</p> <ol style="list-style-type: none"> 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 	4	0...100

	<p>3. Презентація і захист виконаної роботи.</p> <p>4. Тестування знань.</p>		
	<p>СРС. Розробка алгоритму обробки багатовимірних масивів.</p> <p><i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Оголошення багатовимірних масивів. 2. Ініціалізація масивів. 3. Оператори, що використовуються для обробки масивів. 	6	
	<p>ПР9. Дослідження можливостей інтегрованого середовища розробки Visual C# для створення додатків по обробці багатовимірних масивів даних.</p> <p><i>Мета роботи - дослідити можливості інтегрованого середовища розробки Visual Studio 2017 і отримати практичні навички зі створення додатків, що обробляють багатовимірні масиви даних.</i></p> <p>План заняття:</p> <ol style="list-style-type: none"> 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань. 	4	0...100
	<p>Лекція №12. Рядки.</p> <p><i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Створення рядків. 2. Робота з рядками. 3. Масиви рядків. 4. Використання рядків в switch-операторах. <p><i>Список рекомендованих джерел:</i> Основний – 2, 3, 6, 8, 9 Додатковий – 12 ÷ 17 Інтернет-ресурси – 18, 19</p>	2	
	<p>СРС. Розробка алгоритму обробки рядків. Функції, що використовуються</p>	6	

	<p>при обробки рядків. <i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Створення рядків. 2. Функції, що використовуються при обробки рядків. 3. Масиви рядків. 4. Використання рядків в switch-операторах. 		
	<p>ПР10. Дослідження можливостей інтегрованого середовища розробки Visual C# для створення додатків по обробці рядків. <i>Мета роботи - дослідити можливості інтегрованого середовища розробки Visual Studio 2017 і отримати практичні навички зі створення додатків, що обробляють рядки.</i></p> <p style="text-align: center;"><i>План заняття:</i></p> <ol style="list-style-type: none"> 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань. 	4	0...100
	<p>СРС. Структури та перерахування. Розробка алгоритму обробки структур даних. <i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Оголошення структури. 2. Створення і використання змінних типу структур. 3. Масиви структур. 4. Оголошення перерахувань. 5. Ініціалізація перерахувань. 	6	
	<p>Лекція №13. Показчики. <i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Оголошення показчиків. 2. Оператори "*" і "&". 3. Арифметичні операції над показчиками. 4. Показчики та масиви. <p><i>Список рекомендованих джерел:</i></p>	2	

	<p>Основний – 6, 8, 9 Додатковий – 12 ÷ 17</p>		
	<p>СРС. Показчики та дії над ними. <i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Оголошення показчиків. 2. Оператори “*” і “&”. 3. Арифметичні операції над показчиками. 4. Показчики та масиви. 	6	
	<p>ПР11. Дослідження можливостей інтегрованого середовища розробки Visual C# для створення додатків, що використовують показчики. <i>Мета роботи - дослідити можливості інтегрованого середовища розробки Visual Studio 2010 і отримати практичні навички зі створення додатків, що використовують показчики.</i></p> <p>План заняття:</p> <ol style="list-style-type: none"> 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань. 	4	0...100
Тема 8. Файлові структури даних.			
<p>Знати механізм обробки виняткових ситуацій; особливості організації зберігання даних в файлах; принципи побудови файлового вводу-виводу даних; програмні засоби обробки файлових даних.</p>	<p>Лекція №14. Виняткові ситуації. <i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Поняття про виняткову ситуацію. 2. Використання try- і catch-блоків. 3. Генерування виключень. 4. Використання блоку finally. <p><i>Список рекомендованих джерел:</i> Основний – 6, 8, 9 Додатковий – 12 ÷ 17 Інтернет-ресурси – 18, 19</p>	2	
	<p>СРС. Обробка виняткових ситуацій. <i>Основні питання:</i></p> <ol style="list-style-type: none"> 1. Поняття про виняткову ситуацію. 2. Використання try- і catch-блоків. 3. Генерування виключень. 	6	

Вміти використовувати механізм обробки виняткових ситуацій; розробляти програми по обробці даних в текстових і бінарних файлах.	4. Використання блоку finally. Лекція №15. Файловий ввід і вивід. Текстові файли. <i>Основні питання:</i> 1. Відкриття та закриття файлів. 2. Запис байтів в файл. 3. Зчитування байтів з файлів. 4. Зчитування і запис символів з (в) файлів. <i>Список рекомендованих джерел:</i> Основний – 6, 8, 9 Додатковий – 12 ÷ 17 Інтернет-ресурси – 18, 19	2	
	СРС. Символьний і байтовий введення- виведення в С#.	6	
	ПР12. Дослідження можливостей інтегрованого середовища розробки Visual С# для створення додатків по обробці текстових файлів. <i>Мета роботи - дослідити можливості інтегрованого середовища розробки Visual Studio 2017 і отримати практичні навички зі створення додатків по обробці текстових файлів.</i> План заняття: 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.	4	0...100
	Лекція №16. Файловий ввід і вивід. Бінарні файли. <i>Основні питання:</i> 1. Відкриття та закриття файлів. 2. Запис двійкових даних в файл.	2	

	3. Зчитування двійкових даних з файлу. 4. Довільний доступ до даних у файлі. <i>Список рекомендованих джерел:</i> Основний – 6, 8, 9 Додатковий – 12 ÷ 17 Інтернет-ресурси – 18, 19		
	СРС. Розробка додатків по обробці бінарних файлів. <i>Основні питання:</i> 1. Відкриття та закриття файлів. 2. Запис двійкових даних в файл. 3. Зчитування двійкових даних з файлу. 4. Довільний доступ до даних у файлі.	8	
Разом		210	0... 100

5. КОНТРОЛЬ ТА ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Під час вивчення дисципліни «Алгоритмізація та програмування» викладачем здійснюється поточний та підсумковий контроль з використанням тестових завдань. Поточний контроль та оцінювання передбачає:

- перевірку рівня засвоєння теоретичного матеріалу (тестування за матеріалами лекції);
- захист практичних робіт (перевірка розроблених проектів та тестування за матеріалами практичних робіт);
- перевірка засвоєння матеріалу, що винесений на самостійне опрацювання під час фронтального опитування на лекції.

Тестування знань студентів здійснюється під час проведення практичних занять за допомогою автоматизованої системи знань. Підсумкова оцінка (ПО) знань з навчальної дисципліни визначається як середньозважена результатів засвоєння окремих блоків в інтервалі (1... 100) балів:

$$ПО = \frac{\sum_{i=1}^I K_i}{N},$$

де K – кількість балів за кожен блок,

I – кількість зданих блоків,

N – загальна кількість блоків за дисципліну.

У разі незгоди студента з підсумковою оцінкою, підсумковий контроль знань та компетентностей студента з навчальної дисципліни здійснюється на підставі екзамену.

Відповідність підсумкових семестрових рейтингових оцінок у балах оцінкам за національною шкалою та шкалою ECTS

Оцінка в балах	Оцінка за національною шкалою	Оцінка за шкалою ECTS	
		Оцінка	Критерії
90-100	Відмінно	A	“Відмінно” - теоретичний зміст курсу освоєний цілком, без прогалин, необхідні практичні навички роботи з освоєним матеріалом сформовані, усі передбачені програмою навчання навчальні завдання виконані, якість їхнього виконання оцінено числом балів, близьким до максимального.
82 – 89	Добре	B	“Дуже добре” - теоретичний зміст курсу освоєний цілком, без прогалин, необхідні практичні навички роботи з освоєним матеріалом в основному сформовані, усі передбачені програмою навчання навчальні завдання виконані, якість виконання більшості з них оцінено числом балів, близьким до максимального.
75 – 81		C	“Добре” - теоретичний зміст курсу освоєний цілком, без прогалин, деякі практичні навички роботи з освоєним матеріалом сформовані недостатньо, усі передбачені програмою навчання навчальні завдання виконані, якість виконання жодного з них не оцінено мінімальним числом балів, деякі види завдань виконані з помилкам
67 – 74	Задовільно	D	“Задовільно” - теоретичний зміст курсу освоєний частково, але прогалини не носять істотного характеру, необхідні практичні навички роботи з освоєним матеріалом в основному сформовані, більшість передбачених програмою навчання навчальних завдань виконано, деякі з виконаних завдань, можливо, містять помилки.
60 – 66		E	“Достатньо” - теоретичний зміст курсу освоєний частково, деякі практичні навички роботи не сформовані, багато передбачені програмою навчання навчальні завдання не виконані, або якість виконання деяких з них оцінено числом балів, близьким до мінімального.
35 – 59	Незадовільно	FX	“Незадовільно” - теоретичний зміст курсу освоєний частково, необхідні практичні навички роботи не сформовані, більшість передбачених програм навчання навчальних завдань не виконано, або якість їхнього виконання оцінено числом балів, близьким до мінімального; при додатковій самостійній роботі над матеріалом курсу можливе підвищення якості виконання навчальних завдань(з можливістю повторного складання)
1 – 34		F	“Непринятно” - теоретичний зміст курсу не освоєно, необхідні практичні навички роботи не сформовані, усі виконані навчальні завдання містять грубі помилки, додаткова самостійна робота над матеріалом курсу не приведе до якого-небудь значимого підвищення якості виконання навчальних завдань.(з обов'язковим повторним курсом)

6. ЧАС ТА МІСЦЕ ПРОВЕДЕННЯ:

- аудиторні заняття – відповідно до розкладу ХНАДУ, ауд. 214, 216;
- позааудиторна робота – самостійна робота студента, результат виконання якої висвітлено засобами Visual Studio.

7. СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

Основний

1. Государственный стандарт «Схемы алгоритмов, программ, данных и систем». ГОСТ 19.701-90. – М.: Госстандарт, 1991 – 37 с.
2. Ватсон Б. С# 4.0 на примерах. - СПб.: БХВ-Петербург, 2011. – 608с.: ил.
3. Петцольд Ч. Программирование для Microsoft Windows на С#. В 2-х томах. Том 1. Пер. с англ. - М.: «Русская Редакция», 2002.- 576 с.: ил.
4. Петцольд Ч. Программирование для Microsoft Windows на С#. В 2-х томах. Том 2. Пер. с англ. - М.: «Русская Редакция», 2002.- 624 с.: ил.
5. Таненбаум Э. Архитектура компьютера. 5-е изд. - СПб.: БХВ-Петербург, 2010. – 844с.: ил.
6. Тімонін В.О. Конспект лекцій з дисципліни «Алгоритмізація та програмування» для студентів за спеціальністю 121 «Інженерія програмного забезпечення». – Харків: ХНАДУ, 2018. – 128 с. (в електронній формі).
7. Тімонін В.О. Методичні вказівки до проведення практичних робіт з дисципліни «Алгоритмізація та програмування» для студентів за спеціальністю 121 «Інженерія програмного забезпечення». – Харків: ХНАДУ, 2018. – 84 с.
8. Троелсен Э. Язык программирования С# 2010 и платформа .NET 4.0. Пер. с англ. - М.: Издательский дом "Вильямс", 2011. — 1392 с.: ил.
9. Шилдт Г. С# 4.0: полное руководство. – М.: ООО «Вильямс», 2012. – 1056 с.: ил.
10. Фленов М.Е. Библия С#. - СПб.: БХВ-Петербург, 2011. – 560с.: ил.

Додатковий

11. Берковський В.В., Левтеров А.І., Костикова М.В., Онуфрей Ю.Є., Подоляка О.О., Попеленко А.А. Програмування в середовищі С(С++). Збірник задач. – Харків: ХНАДУ, 2006. – 224 с.
12. Голощапов А.Л. Microsoft Visual Studio 2010. – СПб.:БХВ-Петербург, 2011. – 544 с.: ил.
13. Дейтел Х., Дейтел П. Как программировать в С++. 5-е изд. – СПб.: Бином-Пресс, 2008. – 1113 с.: ил.
14. Вступ до програмування мовою С++. Організація обчислень : навч. посіб. / Ю. А. Белов, Т. О. Карнаух, Ю. В. Коваль, А. Б. Ставровський. – К. : Видавничо-поліграфічний центр "Київський університет", 2012. – 175 с. с.: іл. ISBN (укр.)
15. Лафоре Р. Объектно-ориентированное программирование в С++. Классика Computer Science. 4-е изд. – СПб.: Питер, 2008. – 928с. : ил.
16. Прата С. Язык программирования С++. Лекции и упражнения. 5-е

изд.: Пер. с англ. – М.: ООО «Вильямс», 2007. – 1184с.: ил.

17. Страуструп Б. Язык программирования С++. Специальное издание. – СПб.: Бином-Пресс, 2008. – 1104 с.: ил.

Интернет-ресурсы

18. <https://docs.microsoft.com>

19. <https://visualstudio.microsoft.com>