

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ  
 Декан факультету ІТ \_\_\_\_\_  
 Савенко О.С.  
 вересня 2022 р.



**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

**CASE-оцінювання критичних програмних систем: якість, надійність, безпека**  
 Назва

**Статус дисципліни:** вибіркова дисципліна  
**Факультет** – Інформаційних технологій  
**Кафедра** – Комп'ютерної інженерії та інформаційних систем

Форма навчання	Курс	Семестр	Загальне навантаження		Кількість годин						Форма семестрового контролю			
			Кредити ЄКТС	Години	Аудиторні заняття						Курсовий проект	Курсова робота	Залік	Іспит
					Разом	Лекції	Лабораторні роботи	Практичні заняття	Індивідуальна робота студента	Самостійна робота, в т.ч. ІРС				
Д		2	8	240	90	36	36	18		150			+	
<b>Разом ДФН</b>			<b>8</b>	<b>240</b>	<b>90</b>	<b>36</b>	<b>36</b>	<b>18</b>		<b>150</b>			<b>1</b>	

Програма складена \_\_\_\_\_  
 Підпис \_\_\_\_\_ Говорущенко Т.О.  
 Ініціали, прізвище викладача(ів)

Схвалена на засіданні кафедри комп'ютерної інженерії та інформаційних систем

Протокол № 1 від 12 серпня 2022 р.

Зав. кафедри комп'ютерної інженерії та інформаційних систем \_\_\_\_\_  
 Підпис \_\_\_\_\_ Говорущенко Т.О.  
 Ініціали, прізвище

Робоча програма розглянута та схвалена Вченою радою факультету інформаційних технологій

Голова Вченої ради \_\_\_\_\_  
 Підпис \_\_\_\_\_ Савенко О.С.  
 Ініціали, прізвище

## ВСТУП

**Мета викладання дисципліни.** Дисципліна "CASE-оцінювання критичних програмних систем: якість, надійність, безпека" є однією з вибірових дисциплін.

Метою дисципліни «CASE-оцінювання критичних програмних систем: якість, надійність, безпека» є: 1) ознайомити студентів з парадигмами оцінювання критичних програмних систем; 2) надати глибокі та міцні знання з оцінювання критичних програмних систем; 3) ознайомити студентів з теоретичною базою, що використовується при вирішенні задач оцінювання критичних програмних систем; 4) виробити у студентів вміння використовувати набуті знання при оцінюванні критичних програмних систем; 5) підготувати студентів до оцінювання критичних програмних систем у складних і непередбачуваних умовах, що потребує застосування нових підходів та породження нових ідей (креативності), самостійного пошуку помилок, оцінювання своєї поведінки та результатів мислення і постійного самовдосконалення; 6) підготувати студентів до ініціювання та автономного провадження дослідницької та інноваційної діяльності в галузі оцінювання критичних програмних систем.

**Предмет дисципліни.** Оцінювання критичних програмних систем.

**Завдання дисципліни.** Надати студентам знання і практичні навички з оцінювання критичних програмних систем; підготувати студентів до ініціювання та автономного провадження дослідницької та інноваційної діяльності в галузі оцінювання критичних програмних систем.

Після вивчення дисципліни "CASE-оцінювання критичних програмних систем: якість, надійність, безпека" студент має досягти таких результатів навчання (сукупність знань, умінь, навичок, компетентностей):

### **знати:**

- об'єкт, предмет, задачі, проблематику дисципліни та її основні розділи;
- базові поняття й визначення, використовувані у галузі оцінювання критичних програмних систем;
- основи оцінювання критичних програмних систем;
- інновації у галузі оцінювання критичних програмних систем;
- методологію системного аналізу та етапи застосування системного підходу при дослідженні критичних програмних систем;
- концептуальні засади інженерної інноватики в галузі оцінювання критичних програмних систем;
- методи пошуку та прийняття оптимальних рішень в галузі оцінювання критичних програмних систем;
- сучасні (в т.ч. й інтелектуальні) методи математичного та імітаційного моделювання і прогнозування; методи критичного аналізу найбільш передові моделі, методи та алгоритми розв'язання задач оцінювання критичних програмних систем;

### **уміти:**

- виявляти проблему в галузі оцінювання критичних програмних систем, відстежувати найновіші досягнення та знаходити інформацію із сфери наукових інтересів, виявляти в них дискусійні питання та пропонувати шляхи їх розв'язання;
- здійснювати методологічний пошук для визначення сенсу, основ, ідей в галузі оцінювання критичних програмних систем;
- критично сприймати й аналізувати наукові погляди та ідеї у галузі оцінювання критичних програмних систем; концептуально та інноваційно мислити, виявляти невідповідності, суперечності в існуючих науково-теоретичних та науково-методичних підходах, критично оцінювати можливості їх вирішення за допомогою інноваційного науково-методичного інструментарію;
- досліджувати та оцінювати критичні програмні системи;
- виконувати математичне та імітаційне моделювання і прогнозування; збирати, систематизувати, аналізувати, оцінювати, верифікувати, атестувати та критично аналізувати наявну інформацію предметної галузі;
- захищати, пояснювати та аргументувати розробку, отримані результати;
- використовувати фахову літературу досліджуваної предметної галузі; підвищувати професійний рівень шляхом опрацювання сучасних англомовних джерел предметної галузі;

### **бути здатним:**

- розв'язувати найважливіші задачі щодо оцінювання критичних програмних систем, включаючи власні дослідження, які дають можливість створити нове цілісне знання та/або професійну практику;

- критично аналізувати найбільш передові моделі, методи і алгоритми розв'язування задач оцінювання критичних програмних систем, збирати, систематизувати, аналізувати, оцінювати, верифікувати та атестувати передову інформацію;

- діяти у складних і непередбачуваних умовах, що потребує застосування нових підходів та породження нових ідей (креативності), до саморозвитку та самовдосконалення протягом життя;

- провадити дослідницьку та/або інноваційну діяльність, здійснювати критичний аналіз, оцінювати та синтезувати нові складні ідеї в галузі оцінювання критичних програмних систем;

- ініціювати, автономне проведення і адаптацію процесу дослідження з науковою цілісністю; проведення автономного оригінального дослідження, що розширює межі знань за допомогою розробок істотного обсягу, які заслуговують на національні або міжнародні рецензовані публікації;

- спілкуватись в діалоговому режимі з широкою науковою спільнотою та громадськістю в галузі оцінювання критичних програмних систем, до взаємодії та роботи в команді: лідирувати та ефективно організовувати роботу і соціальну комунікацію;

- ініціювати інноваційні комплексні проекти (організовувати процеси проектування, розроблення та обслуговування програмного забезпечення) та повністю автономно їх реалізовувати; відповідати за результати прийняття стратегічних рішень;

- керувати командою при проектуванні, розробленні та реалізації програмних систем, відповідати за результати функціонування таких систем, виявляти та усувати ризики при реалізації проектів, відповідати за стратегічний розвиток команди та за навчання інших.

## CASE-ОЦІНЮВАННЯ КРИТИЧНИХ ПРОГРАМНИХ СИСТЕМ: ЯКІСТЬ, НАДІЙНІСТЬ, БЕЗПЕКА

Тип дисципліни	Вибіркова
Мова викладання	Українська
Семестр	Парний
Кількість встановлених кредитів ЄКТС	8,0
Форми навчання, для яких викладається дисципліна	Денна

### Результати навчання

Студент, який успішно завершив вивчення дисципліни, повинен: вміло *застосовувати* теоретичні та інженерні методи і технології аналізу та оцінювання програмних систем, що базуються на CASE-підході, який припускає використання набору комп'ютеризованих засобів для забезпечення потрібної якості, надійності та безпеки програмних систем при розробленні та застосуванні.

**Зміст навчальної дисципліни.** Оцінювання якості програмних систем: метрико-енергетичні методи та технології; інтелектуальні методи та технології. Оцінювання надійності програмних систем: теорія динаміки та її застосування, ймовірнісні моделі та методи врахування вторинних дефектів. Оцінювання готовності програмних систем: багатофрагментний марківський аналіз, оперативна коригуюча верифікація. Оцінювання функційної безпеки програмних систем: формальні методи та model-checking технології, safety-case методи. Оцінювання функційної безпеки людинно-машинних інтерфейсів: safety-case методи.

**Запланована навчальна діяльність:** лекції - 36 год., практичні заняття – 18 год., лабораторні заняття – 36 год., самостійна робота – 150 год.; разом – 240 год.

**Методи навчання:** методи проблемного викладання, словесні, наочні, проблемно-пошукові (лекції); пояснювально-ілюстративні, проблемного викладання, практичні, проблемно-пошукові, дослідницькі, частково-пошукові (практичні та лабораторні заняття), проблемного викладання, практичні, дослідницькі, частково-пошукові (самостійна робота: індивідуальні завдання).

**Форми і методи оцінювання результатів навчання:** захист практичних робіт, захист лабораторних робіт, контрольна робота

**Вид семестрового контролю:** залік

### Навчальні ресурси:

1. Говорущенко Т. О. Аналіз, дослідження та оцінювання програмних систем: навчальний посібник. – Хмельницький: Хмельницький національний університет, 2019. – 358 с.
2. Міщенко В. О. CASE-оцінка критичних програмних систем: в 3 т. Т. 1. Якість: монографія / Міщенко В. О., Поморова О. В., Говорущенко Т. О. ; за ред. В. С. Харченка. Харків: Нац. аерокосмічний університет «ХАІ», 2012. 201 с.
3. Електронна бібліотека університету. Доступ до ресурсу: [http://lib.khnu.km.ua/asp/php\\_f/page\\_lib.php](http://lib.khnu.km.ua/asp/php_f/page_lib.php).
4. Модульне середовище для навчання MOODLE. Доступ до ресурсу: <https://msn.khnu.km.ua>.

**Викладач:** доктор технічних наук, професор Говорущенко Т.О.

## СТРУКТУРА ЗАЛІКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Назва теми	Кількість годин, відведених на:			
	Денна форма			
	Лекції	Практичні роботи	Лабораторні роботи	СРС
<i>Другий семестр</i>				
Тема 1. CASE-оцінювання критичних програмних систем: якість	12	6	12	50
Тема 2. CASE-оцінювання критичних програмних систем: надійність	12	6	12	50
Тема 3. CASE-оцінювання критичних програмних систем: безпека	12	6	12	50
<b>Разом за 2-й семестр:</b>	<b>36</b>	<b>18</b>	<b>36</b>	<b>150</b>

## ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
	<i>Другий семестр</i>	
1	<p><b>CASE-оцінювання якості критичних програмних систем: увага до ранніх етапів життєвого циклу</b>                      Аналіз впливу інформації специфікації вимог на якість програмного забезпечення. Емерджентні властивості як наслідки недостатності інформації у специфікації вимог до програмного забезпечення  <i>Літ.: [1-14]</i></p>	4
2	<p><b>CASE-оцінювання якості критичних програмних систем: інтелектуальні методи та технології</b>                      Дослідження стандартів та моделей предметної галузі якості програмного забезпечення. Програмна інженерія та штучний інтелект. Онтологічне моделювання як перспективний напрям для галузі інженерії програмного забезпечення  <i>Літ.: [1-14]</i></p>	4
3	<p><b>CASE-оцінювання якості критичних програмних систем: метрико-енергетичні методи та технології</b>                      Основи енергетичного підходу. Метричні оцінки. Наукові метрики продукції та процесу. Емпіричні закони Холстеда та термодинаміка. Енергія, робота й тепло у програмуванні.  <i>Літ.: [1-14]</i></p>	4
4	<p><b>CASE-оцінювання надійності критичних програмних систем: ґрунтовні поняття проблематики надійності</b>                      Ґрунтовні поняття проблематики надійності. Характеристики надійності ПЗ. Порівняльна характеристика надійності апаратних та програмних засобів.  <i>Літ.: [1-14]</i></p>	4
5	<p><b>CASE-оцінювання надійності критичних програмних систем: моделі надійності</b>                      Класифікація моделей надійності ПЗ. Моделі надійності ПЗ на основі неоднорідного пуассонового процесу. Моделі надійності ПЗ на основі компонентного підходу.  <i>Літ.: [1-14]</i></p>	4
6	<p><b>CASE-оцінювання надійності критичних програмних систем: політика оптимального введення ПЗ в експлуатацію</b>                      Моделі і методи визначення політики оптимального введення ПЗ в експлуатацію. Процеси оцінки надійності. Забезпечення надійності в процесах життєвого циклу. Сертифікація ПЗ.  <i>Літ.: [1-14]</i></p>	4
7	<p><b>CASE-оцінювання безпеки критичних програмних систем: проблеми безпеки програмного забезпечення</b>                      Огляд політики в галузі безпеки. Безпека інформаційних технологій. Загрози безпеці програмного забезпечення. Блокчейн-технології..  <i>Літ.: [1-14]</i></p>	4
8	<p><b>CASE-оцінювання безпеки критичних програмних систем: аналіз сучасного стану застосування компонентів безпеки комп'ютерних систем та мереж</b>                      Загальні особливості компонентів безпеки комп'ютерних систем та мереж. Базові підходи до побудови компонентів безпеки комп'ютерних систем та</p>	4

	мереж. Загальні особливості автентифікації користувачів в Інтернеті. Принципи використання біометричних даних в компонентах безпеки комп'ютерних систем та мереж. <i>Літ.: [1-14]</i>	
9	<b>CASE-оцінювання безпеки критичних програмних систем: аналіз сучасних джерел походження загроз інформаційній безпеці</b> Аналіз джерел загроз інформаційній безпеці України. Підходи до формування переліку загроз інформаційній безпеці. Підхід до формування переліку уразливостей інформаційній безпеці. <i>Літ.: [1-14]</i>	4
	Разом за другий семестр:	36 год.

### Зміст практичних занять

№ з/п	Тема практичного заняття	Кількість годин
<i>Другий семестр</i>		
1	CASE-ОЦІНЮВАННЯ ЯКОСТІ КРИТИЧНИХ ПРОГРАМНИХ СИСТЕМ <i>Лім.: [1-14]</i>	6
2	CASE-ОЦІНЮВАННЯ НАДІЙНОСТІ КРИТИЧНИХ ПРОГРАМНИХ СИСТЕМ <i>Лім.: [1-14]</i>	6
3	CASE-ОЦІНЮВАННЯ БЕЗПЕКИ КРИТИЧНИХ ПРОГРАМНИХ СИСТЕМ <i>Лім.: [1-14]</i>	6
Разом за другий семестр		18 год.

### Зміст лабораторних занять

№ з/п	Тема лабораторного заняття	Кількість годин
<i>Другий семестр</i>		
1	ОЦІНЮВАННЯ ЯКОСТІ КРИТИЧНИХ ПРОГРАМНИХ СИСТЕМ <i>Лім.: [1-14]</i>	12
2	ОЦІНЮВАННЯ НАДІЙНОСТІ КРИТИЧНИХ ПРОГРАМНИХ СИСТЕМ <i>Лім.: [1-14]</i>	12
3	ОЦІНЮВАННЯ БЕЗПЕКИ КРИТИЧНИХ ПРОГРАМНИХ СИСТЕМ <i>Лім.: [1-14]</i>	12
Разом за другий семестр		36 год.



### Зміст самостійної (індивідуальної) роботи

Самостійна робота студентів денної форми навчання полягає у систематичному опрацюванні програмного матеріалу, підготовці до виконання і захисту практичних та лабораторних робіт, тестування з теоретичного матеріалу, тощо.

Номер тижня	Вид самостійної роботи	К-ть годин
<i>Другий семестр</i>		
1-2	Опрацювання лекційного матеріалу. Підготовка до практичної роботи №1. Підготовка до лабораторної роботи №1	16
3-4	Опрацювання лекційного матеріалу. Підготовка до практичної роботи №1. Підготовка до лабораторної роботи №1	16
5-6	Опрацювання лекційного матеріалу. Підготовка до практичної роботи №1. Підготовка до лабораторної роботи №1	16
7-8	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №1. Підготовка до захисту практичної роботи №1. Підготовка до практичної роботи №2. Підготовка до лабораторної роботи №2.	17
9-10	Опрацювання лекційного матеріалу. Підготовка до практичної роботи №2. Підготовка до лабораторної роботи №2	17
11-12	Опрацювання лекційного матеріалу. Підготовка до практичної роботи №2. Підготовка до лабораторної роботи №2	17
13-14	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №2. Підготовка до захисту практичної роботи №2. Підготовка до практичної роботи №3. Підготовка до лабораторної роботи №3	17
15-16	Опрацювання лекційного матеріалу. Підготовка до практичної роботи №3. Підготовка до лабораторної роботи №3	17
17-18	Опрацювання лекційного матеріалу. Підготовка до практичної роботи №3. Підготовка до лабораторної роботи №3. Підготовка до захисту лабораторної роботи №3. Підготовка до захисту практичної роботи №3. Підготовка до контрольної роботи	17
Разом за другий семестр:		150

## МЕТОДИ НАВЧАННЯ

Лекції проводяться, в основному, з використанням методів проблемного викладання, словесних, наочних, проблемно-пошукових методів; практичні та лабораторні заняття проводяться пояснювально-ілюстративними методами, методами проблемного викладання, практичними, проблемно-пошуковими, дослідницькими та частково-пошуковими методами; самостійна робота передбачає виконання індивідуальних завдань із залученням методів проблемного викладання, практичних, дослідницьких, частково-пошукових методів.

## ФОРМИ І МЕТОДИ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Поточний контроль здійснюється під час лекційних та практичних занять, а також у дні проведення контрольних заходів, встановлених робочим планом дисципліни. Семестровий контроль проводиться у формі заліку. При цьому при виведенні остаточної оцінки враховуються результати поточного контролю.

Перед вивченням дисципліни, як правило, проводиться вхідний контроль знань з дисциплін, що їй передують і забезпечують. При цьому необхідно встановити рівні та критерії сформованості знань щодо змісту навчальних елементів. Такими рівнями є:

Ознайомчо-орієнтовний (ОО) – особа має орієнтовне уявлення щодо понять, які вивчаються, здатна: оцінювати якість, надійність, безпеку критичних програмних систем; взаємодіяти та працювати в команді при оцінюванні програмних систем: володіти методами і засобами підтримки командної роботи, планувати та ефективно організовувати роботу, соціальну комунікацію та безперервний контроль якості результатів роботи; діяти у складних і непередбачуваних умовах, що потребує застосування нових підходів, креативності, самостійного пошуку помилок, оцінювання своєї поведінки та постійного самовдосконалення; проводити дослідницьку та/або інноваційну діяльність в галузі оцінювання критичних програмних систем.

Понятійно-аналітичний (ПА) – особа має чітке уявлення щодо навчального об'єкту, здатна перенести раніше засвоєні знання на типові ситуації.

Продуктивно-синтетичний (ПС) – особа має глибоке розуміння щодо навчального об'єкту, здатна здійснювати синтез, генерувати нові ідеї та уявлення, переносити раніше засвоєні знання на нетипові, нестандартні ситуації.

Кожний вид роботи з дисципліни оцінюється за *чотирибальною* шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих *позитивно* з врахуванням коефіцієнта вагомості. Вагові коефіцієнти змінюються залежно від структури дисципліни і важливості окремих її видів робіт. Студент, який набрав позитивний середньозважений бал за поточну роботу і не здав залік, вважається невстигаючим.

При оцінюванні знань студентів використовуються різні засоби контролю, зокрема: усне опитування перед допуском до виконання практичної та лабораторної роботи – здійснюється на її початку; якість виконання, набуття теоретичних знань і практичних навичок перевіряється шляхом захисту кожної практичної та лабораторної роботи згідно з робочою програмою дисципліни і робочим навчальним планом.

Оцінка, яка виставляється за *лабораторне та практичне заняття*, складається з таких елементів: усне опитування студентів перед допуском до виконання роботи; знання теоретичного матеріалу з теми; якість оформлення звіту; вміння студента обґрунтувати прийняті конструктивні рішення; своєчасний захист роботи.

При *оцінюванні знань* студентів викладач керується такими критеріями.

Оцінку „зараховано”, за шкалою ECTS – А, отримує студент за глибоке і повне опанування змісту навчального матеріалу, в якому він легко орієнтується, понятійного апарату, за уміння зв'язувати теорію з практикою, вирішувати практичні завдання, висловлювати і обґрунтовувати свої судження. Така оцінка передбачає грамотний, логічний виклад відповіді (як в усній, так і в письмовій формі), якісне зовнішнє оформлення. Студент повинен набути практичних навичок із оцінювання критичних програмних систем. Оцінка „зараховано”, за шкалою ECTS – А, виставляється студенту, який глибоко засвоїв основні принципи оцінювання критичних програмних систем та вміє раціонально застосовувати їх при проектуванні програмних систем. Студент не повинен вагатися при видозміні запитання, повинен робити детальні та узагальнюючі висновки.

Оцінку „зараховано”, за шкалою ECTS – B, отримує студент за повне засвоєння навчального матеріалу, володіння понятійним апаратом, орієнтування у вивченому матеріалі, свідоме використання знань для вирішення практичних завдань, грамотний виклад відповіді, але у змісті і формі відповіді мали місце окремі неточності (похибки), нечіткі формулювання закономірностей тощо. Відповідь студента повинна будуватись на основі самостійного мислення.

Оцінку „зараховано”, за шкалою ECTS – C, отримує студент за правильну відповідь з однією-двома суттєвими помилками.

Оцінки „зараховано”, за шкалою ECTS – D, заслуговує студент, який виявив знання основного навчально-програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, що справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь студента будується на рівні репродуктивного мислення, студент слабо знає структуру курсу, допускає помилки у відповіді, засвоїв і набув практичних навичок з оцінювання критичних програмних систем, але припустився неточностей. Вагається при відповіді на видозмінене запитання, разом з тим студент володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.

Оцінки „зараховано”, за шкалою ECTS – E, заслуговує студент за неповне опанування програмного матеріалу, але отримані знання і набуті практичні навички із оцінювання якості та експертизи ПЗ.

Оцінка „незараховано”, за шкалою ECTS – FX, виставляється, коли студент має розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань. Як правило, оцінка "незараховано", за шкалою ECTS – F, виставляється студенту, який не може продовжити навчання без додаткових знань з курсу.

На основі результатів поточного контролю виставляється підсумкова семестрова оцінка. На основі аналізу контролю знань викладач удосконалює курс лекцій, звертаючи особливу увагу на ті розділи, чи теми, з яких було найбільше неточних відповідей, що свідчить про методичні чи інші недоліки при висвітленні вказаних тем або розділів.

Аналогічно вносяться корективи в методичні посібники для практичних та лабораторних робіт, детальніше розглядаються принципові питання при виконанні практичних та лабораторних робіт та їх захисті.

### Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів денної форми навчання у семестрі за ваговими коефіцієнтами

Аудиторна робота						Самостійна, індивідуальна робота	Залік
<i>II семестр</i>							
Лабораторні роботи №:			Практичні роботи №			Контроль	
1	2	3	1	2	3	Контрольна робота	
ВК:						0,8	0,2

Умовні позначення: Т – тема дисципліни; ВК – ваговий коефіцієнт.

Підсумкова семестрова оцінка за національною шкалою і шкалою ЄКТС встановлюється в автоматизованому режимі після внесення усіх оцінок до електронного журналу. Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС наведені у наступній таблиці.

### Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	Інституційна шкала балів	Інституцій на оцінка	Критерії оцінювання
A	4,75–5,00	5	Зараховано
B	4,25–4,74	4	
			<b>Відмінно</b> – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навичок
			<b>Добре</b> – повне знання навчального матеріалу з кількома незначними помилками

C	3,75–4,24			<i>Добре</i> – в загальному правильна відповідь з двома-трьома суттєвими помилками
D	3,25–3,74	3		<i>Задовільно</i> – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією
E	3,00–3,24	3		<i>Задовільно</i> – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00–2,99	2	Незараховано	<i>Незадовільно</i> – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00–1,99	2		<i>Незадовільно</i> – необхідна серйозна подальша робота і повторне вивчення дисципліни

## ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ СТУДЕНТІВ

1. Аналіз впливу інформації специфікації вимог на якість програмного забезпечення.
2. Емерджентні властивості як наслідки недостатності інформації у специфікації вимог до програмного забезпечення.
3. Дослідження стандартів та моделей предметної галузі якості програмного забезпечення.
4. Програмна інженерія та штучний інтелект.
5. Онтологічне моделювання як перспективний напрям для галузі інженерії програмного забезпечення
6. Основи енергетичного підходу при оцінюванні якості програмних систем.
7. Енергетичні метрики оцінювання якості продукції та процесів.
8. Технології та приклади оцінювання якості програмних систем в розробленні.
9. Основи теорії динаміки програмних систем.
10. Аналіз стійкості програмних систем з врахуванням потоків дефектів.
11. Технології та приклади оцінювання надійності програмних систем.
12. Сценарії усунення та внесення дефектів у програмних системах.
13. Ймовірнісні моделі зростання надійності програмних систем.
14. Методи оцінювання надійності програмних засобів на основі SRGM+.
15. Інформаційна технологія оцінювання готовності програмних систем на основі багатодіагностичних марківських моделей.
16. Елементи методології оперативної коригуючої верифікації.
17. Моделі готовності програмних систем.
18. Технологія оцінювання готовності програмних систем з оперативною коригуючою верифікацією.
19. Методи оцінювання безпеки на основі FMECA, RBD, MM.
20. Використання формальних методів в контексті інваріантного підходу до оцінювання безпеки.
21. Метод формальної верифікації Model Checking.
22. Основи Safety-Case-методології.
23. Методи оцінювання безпеки на основі Safety-Case-ядер.
24. Оцінювання функціональної безпеки програмних систем на основі COTS компонентів.
25. Аналіз інструментальних засобів, використовуваних при розробленні обґрунтувань безпеки та гарантії.
26. Моделі якості та безпеки інтерфейсів користувача в контексті Safety-Case-методології.
27. Методи і засоби оцінювання якості та безпеки інтерфейсів користувача.
28. Методи оцінювання впливу людського фактору на безпеку інформаційно-керуючих систем в контексті Safety-Case-методології.

## **МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ**

Навчальний процес з дисципліни «CASE-оцінювання критичних програмних систем: якість, надійність, безпека» повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою. Зокрема, викладачами кафедри підготовлені:

1. Говорущенко Т. О. Методологія оцінювання достатності інформації для визначення якості програмного забезпечення: монографія / Т. О. Говорущенко. – Хмельницький: Хмельницький національний університет, 2017. – 310 с.

2. Міщенко В. О. CASE-оцінка критичних програмних систем: в 3 т. Т. 1. Якість: монографія / Міщенко В. О., Поморова О. В., Говорущенко Т. О. ; за ред. В. С. Харченка. Харків: Нац. аерокосмічний університет «ХАІ», 2012. 201 с.

3. Говорущенко Т. О. Аналіз, дослідження та оцінювання програмних систем: навчальний посібник. – Хмельницький: Хмельницький національний університет, 2019. – 358 с.

## **РЕКОМЕНДОВАНА ЛІТЕРАТУРА**

1. Говорущенко Т. О. Аналіз, дослідження та оцінювання програмних систем: навчальний посібник. – Хмельницький: Хмельницький національний університет, 2019. – 358 с.

2. Міщенко В. О. CASE-оцінка критичних програмних систем: в 3 т. Т. 1. Якість: монографія / Міщенко В. О., Поморова О. В., Говорущенко Т. О. ; за ред. В. С. Харченка. Харків: Нац. аерокосмічний університет «ХАІ», 2012. 201 с.

3. McConnell S. Software Estimation: Demystifying the Black Art (Developer Best Practices). – Microsoft Press, 2016. – 308 p.

4. Levenson N. G. Engineering a safer world: systems thinking applied to safety / N. G. Levenson. – MIT Press, 2012. – 560 pp.

5. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models: ISO/IEC 25010:2011. – [Introduced 01.03.2011]. – Geneva (Switzerland): ISO, 2011. – 34 p. – (International standard).

6. Software Engineering. Guide to the software engineering body of knowledge (SWEBOK): ISO/IEC TR 19759:2015. – [Introduced 01.10.2015]. – Geneva (Switzerland): ISO, 2015. – 336 p. – (International standard).

7. Systems and software engineering. Software life cycle processes: ISO/IEC/IEEE CD 12207:2016. – [Introduced 21.09.2016]. – Geneva (Switzerland): ISO, 2016. – 162 p. – (International standard).

8. Systems and software engineering. Life cycle processes. Requirements engineering: ISO/IEC/IEEE 29148:2011. – [Introduced 01.12.2011]. – Geneva (Switzerland): ISO, 2011. – 28 p. – (International standard).

## **ІНФОРМАЦІЙНІ РЕСУРСИ**

### **Електронний університет:**

1. Електронна бібліотека університету
2. Модульне середовище для навчання