

**МЕЛІТОПОЛЬСЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ БОГДАНА ХМЕЛЬНИЦЬКОГО**

ФАКУЛЬТЕТ ІНФОРМАТИКИ, МАТЕМАТИКИ ТА ЕКОНОМІКИ

Кафедра інформатики і кібернетики

Назва курсу <i>Нормативний/вибірковий</i>	ОПП-03 Стандартизація та сертифікація програмного забезпечення. <i>Нормативний</i>
Ступінь освіти Освітня програма	Магістр Професійна освіта. Цифрові технології
Рік викладання/ Семестр/ Курс (рік навчання)	<i>2019/ I,II семестр /1 курс (1,9 років)</i>
Викладач	Осадчий В.В.
Профайл викладача	http://inf.mdpu.org.ua/2017/10/25/osadchij-vjacheslav-volodimirovich-2/
Контактний тел.	+38 (096) 27-98-686
E-mail:	osadchy@mdpu.org.ua
Сторінка курсу в ЦОДТ МДПУ ім. Б.Хмельницького	http://www.dfn.mdpu.org.ua/enrol/index.php?id=407 http://www.dfn.mdpu.org.ua/enrol/index.php?id=417
Консультації	Очні консультації: згідно графіку роботи кафедри інформатики і кібернетики. Онлайн-консультації: через систему ЦОДТ МДПУ ім. Б. Хмельницького.

1. АНОТАЦІЯ

Вимоги міжнародних, регіональних та державних стандартів України у процесі впровадження і налаштування програмного забезпечення та інформаційних ресурсів, при розробці й виготовленні програмного забезпечення та експлуатаційної документації на створювані програмні продукти є важливими на сучасному етапі розвитку цифрових технологій. Актуальними також є питання ліцензування програмних продуктів та їх використання у закладах освіти та ІТ-підприємствах.

Курс «Стандартизація та сертифікація програмного забезпечення» є логічним продовженням циклу комп'ютерних дисциплін і разом з цим виступає інструментом надання майбутньому спеціалісту чіткого розуміння про моделі і методи та програмні засоби для забезпечення якості в життєвому циклі засобів інформаційних технологій. Програма дисципліни передбачає формування у здобувачів вищої освіти комплексу знань, необхідних для розуміння проблем, які виникають під час побудови та при використанні сучасних програмних систем, формування теоретичних знань та

практичних навичок щодо відповідності програмного забезпечення з урахуванням законодавчої бази, що діє в Україні, і вимог національних і міжнародних стандартів.

2. МЕТА ТА ЗАВДАННЯ КУРСУ

Мета навчальної дисципліни – формування у студентів глибоких систематизованих знань у галузі загальних науково-технічних і практичних основ, засадних методологічних та організаційних положень стандартизації та сертифікації програмного забезпечення в Україні і в світі для досягнення при практичній діяльності високих кінцевих результатів; ознайомлення із сучасними програмними засобами верифікації та оптимізації програмного забезпечення на заключних етапах його розробки.

Завданнями курсу є розвинування у студентів теоретичних і практичних навичок роботи з нормативними документами стосовно сертифікації, стандартизації та правового захисту програмних засобів; формування знань необхідних для застосування сучасних методів верифікації та оптимізації програмних засобів та необхідних для цього інструментальних програмних засобів.

3. ПЕРЕЛІК КОМПЕТЕНТНОСТЕЙ, ЯКІ НАБУВАЮТЬСЯ ПІД ЧАС ОПАНУВАННЯ ОСВІТНІМ КОМПОНЕНТОМ

Загальні компетентності (ЗК)

ЗК 7. Здатність вчитися і оволодівати сучасними знаннями.

ЗК 8. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

Спеціальні (фахові, предметні компетентності) (ФК)

ФК 10. Здатність розробляти програмне забезпечення із застосуванням різних парадигм програмування, методів та алгоритмів обчислень, структур даних, дотримуючись усіх етапів життєвого циклу проекту.

ФК 12. Здатність розв'язувати широке коло проблем і задач сертифікації програмного забезпечення.

4. РЕЗУЛЬТАТИ НАВЧАННЯ

ПРН 13. Кваліфіковано проводити сертифікацію проектних інформаційних засобів.

ПРН 15. Володіти знаннями про основні види інформаційних систем та інструментальні засоби їх розробки.

5. ОБСЯГ КУРСУ

Вид заняття	Загальна кількість	Лекції	Практичні /лабораторні заняття	Самостійна робота
Кількість годин перший семестр	135 годин	16 годин	28 годин	91 годин
Кількість годин другий семестр	135 годин	18 години	42 годин	75 години

6. ПОЛІТИКА КУРСУ

Політика академічної поведінки та етики:

- не пропускати та не запізнюватися на заняття за розкладом;
- вчасно виконувати завдання практичних занять та питань самостійної роботи;
- на заняттях дотримуватись правил роботи у групі, шанобливо ставитись до поглядів один одного;
- вчасно та самостійно виконувати контрольні завдання, презентації та творчі проекти;
- дотримуватись здобувачами вищої освіти політики доброчесності під час виконання самостійної або індивідуальної роботи;
- активно брати участь у побудові і реалізації власного індивідуального освітнього маршруту.

Політика щодо дедлайнів та перескладання: роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку. Перескладання модулів відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

Політика щодо академічної доброчесності: усі письмові роботи перевіряються на наявність плагіату і допускаються до захисту із коректними текстовими запозиченнями не більше 20%. Списування під час контрольних робіт заборонені (в т.ч. із використанням мобільних девайсів). Політика щодо відвідування: Відвідування занять є обов'язковим. За об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання може відбуватись в онлайн формі за погодженням із керівником курсу.

Поточний контроль: усне опитування в ході лекцій та практичних занять, перевірка завдань (у тому числі самостійної роботи), оцінювання правильності вирішення тестових та практичних завдань.

7. СТРУКТУРА КУРСУ

Кількість годин	Тема	Форма діяльності (заняття, кількість годин)	Література	Завдання	Вага оцінки	Термін виконання
Семестр 1.						
Блок 1. Сертифікація програмного забезпечення						
14	Тема 1. Систематизація процесів життєвого циклу.	Лекція (4 год.), лабораторне заняття (2 год.), самостійна робота (8 год.)	Основна: 3, 6, 7 Додаткова: 9, 23, 26, 50	опитування; виконання завдань лабораторної роботи	5	впродовж першого навчального семестру (перший періодичний контроль)
12	Тема 2. Основні моделі життєвого циклу.	Лабораторне заняття (4 год.), самостійна робота (8 год.)	Основна: 2, 3, 6 Додаткова: 9, 24, 28	опитування; виконання завдань лабораторної роботи	5	впродовж першого навчального семестру (перший періодичний контроль)

12	Тема 3. Моделі життєвого циклу в деяких реальних методологіях програмування.	Лекція (2 год.), лабораторне заняття (2 год.), самостійна робота (8 год.)	Основна: 3, 6, 7 Додаткова: 3, 9, 25, 39	опитування; виконання завдань лабораторної роботи	5	впродовж першого навчального семестру (перший періодичний контроль)
12	Тема 4. Забезпечення якості програмних засобів.	Лекція (2 год.), лабораторне заняття (2 год.), самостійна робота (8 год.)	Основна: 2, 7 Додаткова: 8, 15-18, 40-42	опитування; виконання завдань лабораторної роботи	5	впродовж першого навчального семестру (перший періодичний контроль)
10	Тема 5. Стандартизація програмних засобів.	Лабораторне заняття (2 год.), самостійна робота (8 год.)	Основна: 11, 15 Додаткова: 7, 10, 49	опитування; виконання завдань лабораторної роботи	5	впродовж першого навчального семестру (перший періодичний контроль)
14	Тема 6. Стандартизація в галузі забезпечення якості програмних засобів.	Лекція (2 год.), лабораторне заняття (4 год.), самостійна робота (8 год.)	Основна: 9, 10, 14 Додаткова: 6	опитування; виконання завдань лабораторної роботи	5	впродовж першого навчального семестру (перший періодичний контроль)
Блок 2. Сертифікація та ліцензування програмного забезпечення. Охорона інтелектуальної власності в Україні						
11	Тема 7. Документування програм.	Лабораторне заняття (4 год.), самостійна робота (7 год.)	Основна: 5 Додаткова: 11	опитування; виконання завдань лабораторної роботи	5	впродовж першого навчального семестру (другий періодичний контроль)
10	Тема 8. Сертифікація програмних засобів.	Лекція (2 год.), лабораторне заняття (2 год.), самостійна робота (6 год.)	Основна: 5, 10, 13 Додаткова: 9, 12, 21	опитування; виконання завдань лабораторної роботи	5	впродовж першого навчального семестру (другий періодичний контроль)
8	Тема 9. Ліцензування	Лекція (2 год.),	Основна: 5, 10	опитування; виконання	5	впродовж першого

	програмного продукту.	самостійна робота (6 год.)	Додаткова: 9, 21	завдань самостійної роботи		навчального семестру (другий періодичний контроль)
8	Тема 10. Об'єкти та суб'єкти права інтелектуальної власності	Лабораторне заняття (2 год.), самостійна робота (6 год.)	Основна: 5 Додаткова: 13, 38	опитування; виконання завдань лабораторної роботи	5	впродовж першого навчального семестру (другий періодичний контроль)
8	Тема 11. Охорона патентного права.	Лекція (2 год.), самостійна робота (6 год.)	Основна: 5 Додаткова: 19	опитування; виконання завдань самостійної роботи	5	впродовж першого навчального семестру (другий періодичний контроль)
8	Тема 12. Авторське право на програмний продукт.	Лабораторне заняття (2 год.), самостійна робота (6 год.)	Основна: 5 Додаткова: 31, 34	опитування; виконання завдань лабораторної роботи	5	впродовж першого навчального семестру (другий періодичний контроль)
8	Тема 13. Охорона комерційної таємниці та нових технологічних об'єктів.	Лабораторне заняття (2 год.), самостійна робота (6 год.)	Основна: 5 Додаткова: 14, 37	опитування; виконання завдань лабораторної роботи	5	впродовж першого навчального семестру (другий періодичний контроль)
Семестр 2.						
Блок 1. Тестування та верифікація програмного забезпечення						
10	Тема 1. Основи тестування програмного забезпечення	Лекція (2 год.), самостійна робота (8 год.)	Основна: 1, 4 Додаткова: 5, 43	виконання завдань самостійної роботи	5	впродовж другого навчального семестру (перший періодичний контроль)
4	Тема 2. Огляд загальних підходів до тестування	Лабораторне заняття (4 год.)	Основна: 4 Додатк	опитування; виконання завдань	5	впродовж другого навчального

			ова: 2, 43, 44	лабораторної роботи		семестру (перший періодичний контроль)
10	Тема 3. Види тестування.	Лекція (2 год.), самостійна робота (8 год.)	Основна: 8 Додаткова: 2, 4, 45	виконання завдань самостійної роботи	5	впродовж другого навчального семестру (перший періодичний контроль)
2	Тема 4. Техніки, що базуються на специфікації	Лекція (2 год.)	Основна: 2, 8 Додаткова: 1, 8, 46			впродовж другого навчального семестру (перший періодичний контроль)
6	Тема 5. Класи еквівалентності. Ручне тестування в MVSTE.	Лабораторне заняття (6 год.)	Основна: 2, 4 Додаткова: 9, 27	опитування; виконання завдань лабораторної роботи	5	впродовж другого навчального семестру (перший періодичний контроль)
1	Тема 6. Процес тестування, документування та аналіз результатів.	Лекція (1 год.)	Основна: 2, 8 Додаткова: 5, 22, 43-45			впродовж другого навчального семестру (перший періодичний контроль)
9	Тема 7. Поняття і структура UML.	Лекція (1 год.), самостійна робота (9 год.)	Основна: 4 Додаткова: 2, 10	виконання завдань самостійної роботи	5	впродовж другого навчального семестру (перший періодичний контроль)
8	Тема 8. Робота в Umbrello UML Modeller. Користувачський інтерфейс	Лабораторне заняття (8 год.)	Основна: 4, 8 Додаткова: 5	опитування; виконання завдань лабораторної роботи	5	впродовж другого навчального семестру (перший періодичний контроль)

2	Тема 9. Критерії вибору тестів.	Лекція (2 год.)	Основна: 4, 14 Додаткова: 22			впродовж другого навчального семестру (перший періодичний контроль)
10	Тема 10. Проведення формальних інспекцій.	Самостійна робота (10 год.)	Основна: 8 Додаткова: 16, 18	виконання завдань самостійної роботи	5	впродовж другого навчального семестру (перший періодичний контроль)
8	Тема 11. Робота в Umbrello UML Modeller. Імпорт і генерація кода	Лабораторне заняття (8 год.)	Основна: 4 Додаткова: 20	опитування; виконання завдань лабораторної роботи	5	впродовж другого навчального семестру (перший періодичний контроль)
12	Тема 12. Формальні інспекції програмного коду.	Лекція (2 год.), самостійна робота (10 год.)	Основна: 8 Додаткова: 33, 45	виконання завдань самостійної роботи	5	впродовж другого навчального семестру (перший періодичний контроль)
Блок 2. Верифікація програмного забезпечення.						
10	Тема 13. Модульне тестування.	Самостійна робота (10 год.)	Основна: 4, 8 Додаткова: 44, 48	виконання завдань самостійної роботи	5	впродовж другого навчального семестру (другий періодичний контроль)
8	Тема 14. Робота в Umbrello UML Modeller. Додаткові можливості	Лабораторне заняття (8 год.)	Основна: 4, 15 Додаткова: 30, 32	опитування; виконання завдань лабораторної роботи	5	впродовж другого навчального семестру (другий періодичний контроль)
10	Тема 15. Інтеграційне та системне	Лекція (2 год.), самостійна	Основна: 8, 14 Додатк	виконання завдань самостійної	5	впродовж другого навчального

	тестування.	робота (10 год.)	ова: 35, 36	роботи		семестру (другий періодичний контроль)
10	Тема 16. Тестування для користувача інтерфейсу.	Лекція (2 год.), самостійна робота (10 год.)	Основна: 8, 14 Додаткова: 14, 47	виконання завдань самостійної роботи	5	впродовж другого навчального семестру (другий періодичний контроль)
8	Тема 17. Побудова TTCN тестів на основі UML-діаграм	Лабораторне заняття (8 год.)	Основна: 3, 12 Додаткова: 24	опитування; виконання завдань лабораторної роботи	5	впродовж другого навчального семестру (другий періодичний контроль)
2	Тема 18. Підтримка процесу тестування при промисловій розробці програмного забезпечення.	Лекція (2 год.)	Основна: 1, 12 Додаткова: 29, 34, 35			впродовж другого навчального семестру (другий періодичний контроль)

7.2. СТРУКТУРА КУРСУ (ЛЕКЦІЙНИЙ БЛОК)

Тема лекції	Зміст лекції
Семестр 1.	
<i>Систематизація процесів життєвого циклу.</i>	1. Життєвий цикл програмного забезпечення та його стандартизація 2. Систематизація процесів життєвого циклу програмного засобу 3. Основні процеси життєвого циклу програмного засобу
<i>Моделі життєвого циклу в деяких реальних методологіях програмування.</i>	1. Моделі життєвого циклу 2. Водоспадна (каскадна) модель (англ. waterfall model) 3. Ітеративна й інкрементна модель – еволюційний підхід 4. Спиральна модель
<i>Забезпечення якості програмних засобів.</i>	1. Способи забезпечення якості програмних продуктів 2. Забезпечення надійності на етапі кодування і компіляції програмного забезпечення. 3. Тестування програмного забезпечення
<i>Стандартизація в галузі забезпечення якості програмних засобів.</i>	1. Стандарти в галузі програмного забезпечення 2. Сертифікація. Загальні положення 3. комплексів програм 4. Особливості забезпечення надійності функціонування імпортованих програмних засобів
<i>Сертифікація</i>	1. Сертифікація. Загальні положення

<i>програмних засобів.</i>	<ol style="list-style-type: none"> Сертифікація комплексів програм Сертифікація програмного продукту Порядок сертифікації програмних засобів та інформаційних технологій
<i>Ліцензування програмного продукту.</i>	<ol style="list-style-type: none"> Комп'ютерна програма як об'єкт авторського права Правова охорона програмного забезпечення в Україні Види ліцензій на програмні продукти Моделі відкритості програмного забезпечення
<i>Охорона патентного права.</i>	<ol style="list-style-type: none"> Загальна характеристика патентного права та його джерел. Патент та порядок його отримання.
Семестр 2.	
<i>Основи тестування програмного забезпечення</i>	<ol style="list-style-type: none"> Термінологія тестування: поняття дефекту, збою та відмови; типи дефектів, їх класифікація та статистика виникнення. Кількісні оцінки при тестуванні: метрики підрахунку дефектів; щільність дефектів; метрики покриття й глибини тестування.
<i>Види тестування</i>	<ol style="list-style-type: none"> Рівні тестування: об'єкти тестування; модульне тестування; інтеграційне тестування; системне тестування. Тестування методами «білого ящика», «сірого ящика» та «чорного ящика». Цілі тестування. Кваліфікаційні випробування. Інсталяційне, альфа-і бета-тестування. Функціональне, регресійне, навантажувальне, стрес-тестування. Тестування продуктивності. Сумісність ролей при виконанні проекту.
<i>Техніки тестування</i>	<ol style="list-style-type: none"> Техніки, що базуються на досвіді й інтуїції. Техніки, що базуються на специфікації. Техніки, орієнтовані на код. Тестування, орієнтоване на дефекти. Техніки, що базуються на аналізі подальшого використання. Техніки, що базуються на природі застосування
<i>Процес тестування, документування та аналіз результатів</i>	<ol style="list-style-type: none"> Управління та структура процесу тестування. Внутрішні та незалежні команди. Розробка тест-стратегії. Генерація сценаріїв тестування. Виконання тестів. Процес завершення тестування. Повторне використання й шаблони тестів. Документування тестів і робочого продукту.
<i>Поняття і структура UML</i>	<ol style="list-style-type: none"> Діаграми варіантів використання, діаграми взаємодії, їх елементи та зв'язки між елементами. Види діаграм взаємодії та область їх застосування. Типи повідомлень на діаграмах послідовності, комбіновані фрагменти взаємодії, оператори взаємодії. Діаграми класів, станів, діяльності, діаграми компонентів та діаграми розміщення, їх елементи та зв'язки, область застосування
<i>Критерії вибору тестів</i>	<ol style="list-style-type: none"> Структурні критерії Функціональні критерії Критерії стохастичного тестування Мутаційні критерії
<i>Формальні інспекції програмного коду</i>	<ol style="list-style-type: none"> Формальні інспекції програмного коду Особливості етапу перегляду коду, що інспектується Особливості етапу проведення зборів Особливості етапу завершення й повторної інспекції Особливості етапу перегляду документації

	6. Особливості етапу завершення.
<i>Інтеграційне та системне тестування</i>	1. Завдання та цілі інтеграційного тестування. 2. Організація інтеграційного тестування. 3. Завдання та цілі системного тестування. 4. Види системного тестування.
<i>Тестування для користувача інтерфейсу</i>	1. Задачі і цілі тестування користувальницького інтерфейсу. 2. Функціональне тестування користувачьких інтерфейсів. 3. Тестування зручності використання користувачьких інтерфейсів.
<i>Підтримка процесу тестування при промисловій розробці</i>	1. Управління якістю. 2. Аудит процесів розробки та верифікації. 3. Конфігураційне управління. 4. Завдання процесу конфігураційного управління

7.3. СТРУКТУРА КУРСУ (ЛАБОРАТОРНІ ЗАНЯТТЯ)

Тема	Зміст лабораторних занять
Семестр 1	
Систематизація процесів життєвого циклу. Система управління проектами MS PROJECT	1. Хто основний користувач діаграми Ганта? 2. Що міститься у таблиці уявлення Ганта, а що подається на графічній частині? 3. Об'ясніть структуру таблиці завдань.
Основні моделі життєвого циклу. Моделювання проекту на стадії його створення	1. Як поєднується інформація про робочий час для проекту і для конкретного ресурсу ? 2. Як встановлюється зв'язок між таблицею робіт і таблицею ресурсів?
Основні моделі життєвого циклу. Регулювання процесу виконання проекту у відповідності зі зміною умов його реалізації	1. Для яких цілей використовується узгоджений план? Як його додати до моделі проекту? 2. Якими способами забезпечується узгодження використання ресурсів на різних роботах ?
Моделі життєвого циклу в деяких реальних методологіях програмування. Створення та планування реалізації мультипроекту засобами пакета MS PROJECT	1. Як упорядкувати ресурси в таблиці ресурсів у порядку зменшення вартості їх використання? 2. Як розташувати роботи на графіку Ганта в порядку скорочення їх тривалості? 3. Як відобразити текстові дані на графіку Ганта?
Забезпечення якості програмних засобів. Аналіз плану проектних робіт та вартість проекту	1. Як зменшити чи збільшити витрати на проект? 2. Викладить сутність метода PERT. 3. Викладить сутність метода критичного шляху.
Стандартизація програмних засобів. Моніторинг проекту	1. Дайте визначення плановим показникам, фактичним показникам, показникам поточного розкладу. 2. Назвіть склад котролюємих параметрів при контролі за виконанням проекту.
Стандартизація в галузі	1. Основні функції та призначення систем CRM.

<p>забезпечення якості програмних засобів. Інформаційна система CRM</p>	<ol style="list-style-type: none"> 2. Загальні правила роботи з системи Monitor CRM. Додавання, видалення і зміна (редагування) елементів. 3. Групи і підгрупи елементів.
<p>Документування програм. Розробка технічного завдання.</p>	<ol style="list-style-type: none"> 1. Що таке технічне завдання на розробку програмного продукту? 2. Для чого потрібне технічне завдання? 3. На якій стадії проекту розробляється технічне завдання?
<p>Сертифікація програмних засобів. Сертифікаційні іспити</p>	<ol style="list-style-type: none"> 1. Що представляє собою сертифікаційний іспит? 2. Де програміст може отримати сертифікат? 3. Чи може програміст отримати сертифікат заочно?
<p>Об'єкти та суб'єкти права інтелектуальної власності Авторське право і суміжні права.</p>	<ol style="list-style-type: none"> 1. Що слід зробити, якщо узгодження використання ресурсів призводить до неприйнятно великих затримок ? 2. Як обчислюється тривалість роботи у разі зміни обсягу виділеного на її виконання ресурсу при включеному перемикачі Фіксований обсяг робіт?
<p>Авторське право на програмний продукт. Підготовка документів на отримання авторського права.</p>	<ol style="list-style-type: none"> 1. У яких випадках не слід включати перемикач Фіксований обсяг робіт при введенні в модель проекту нової роботи ? 2. Чому може виникати неузгодженість між даними стовпців Початок, Закінчення і Тривалість ? Як уникнути цієї ситуації?
<p>Охорона комерційної таємниці та нових технологічних об'єктів. Право промислової власності.</p>	<ol style="list-style-type: none"> 1. Об'єкти та суб'єкти авторського права. 2. Закон України «Про захист від недобросовісної конкуренції».
<p>Семестр 2.</p>	
<p>Огляд загальних підходів до тестування.</p>	<p>На основі опису системи "Калькулятор" наданої в теоретичній частині розробити функційні вимоги до програми, а саме:</p> <ol style="list-style-type: none"> 1. Загальні вимоги до програми. 2. Вимоги до арифметичних операцій. 3. Вимоги до вхідного виразу.
<p>Класи еквівалентності. Ручне тестування в MVSTE.</p>	<p>Скласти тест-вимоги і провести ручне тестування таких методів:</p> <ol style="list-style-type: none"> 1. Знаходження залишку 2. Унарний плюс 3. Унарний мінус 4. Віднімання
<p>Робота в Umbrello UML Modeller. Користувацький інтерфейс</p>	<ol style="list-style-type: none"> 1. Користувацький інтерфейс. 2. Ознайомитися з інтерфейсом програми для створення UML -діаграм.

	<p>3. Намалювати у програмі діаграму варіантів використання, діаграму класів, діаграму послідовностей, діаграму взаємодії. Робота в Umbrello UML Modeller.</p> <p>4. На основі розробленої на попередньому занятті діаграми класів згенеруйте код програми. Зробіть скріншот або збережіть код у текстовому редакторі. Намалюйте діаграму стану, діаграму активності, діаграму клонпонентів</p>
Робота в Umbrello UML Modeller. Імпорт і генерація кода	<p>1. Ознайомитися з інтерфейсом програми для створення UML -діаграм.</p> <p>2. Намалювати у програмі діаграму варіантів використання, діаграму класів, діаграму послідовностей, діаграму взаємодії.</p> <p>3. Робота в Umbrello UML Modeller. Імпорт і генерація кода</p> <p>4. На основі розробленої на попередньому занятті діаграми класів згенеруйте код програми. Зробіть скріншот або збережіть код у текстовому редакторі.</p> <p>5. Намалюйте діаграму стану, діаграму активності, діаграму клонпонентів</p>
Робота в Umbrello UML Modeller. Додаткові можливості	<p>1. Ознайомитися з додатковими можливостями Umbrello UML Modeller.</p> <p>2. Навчитися копіювати об'єкти у вигляді зображень та працювати з логічними теками.</p>
Побудова TTCN тестів на основі – діаграм	<p>1. Спроектуйте тест для вашої програми.</p> <p>2. Розробіть сценарій тестування для вашої програми.</p>

7.4 СТРУКТУРА КУРСУ (ТЕМИ ДЛЯ САМОСТІЙНОГО ОПРАЦЮВАННЯ)

№ з/п	Теми для самостійного опрацювання
Семестр 1.	
1	Систематизація процесів життєвого циклу: Розглянути варіанти розвитку проекту розробки програмного забезпечення.
2	Основні моделі життєвого циклу: Зробити порівняльний аналіз моделей життєвого циклу.
3	Моделі життєвого циклу в деяких реальних методологіях програмування: Розробити алгоритм здійснення екстремального програмування.
4	Забезпечення якості програмних засобів: Провести тестування та налагодити програмні компоненти у реальному часі
5	Стандартизація програмних засобів: Провести дослідження процесів життєвого циклу програмних засобів відповідно до

	стандарту ISO 12207
6	Стандартизація в галузі забезпечення якості програмних засобів: Провести тестування та випробування надійності комплексу програм при впливі оператора-користувача
7	Документування програм: Провести оцінювання програмного продукту відповідно до стандарту ДСТУ ISO/IEC 14598.
8	Сертифікація програмних засобів: Здати сертифікаційний іспит у сертифікаційному центрі за вибором
9	Ліцензування програмного продукту: Зробити огляд академічних програм ліцензування. Розробити план заходів щодо перевірки ліцензійності ПЗ, яке використовується на підприємстві
10	Об'єкти та суб'єкти права інтелектуальної власності: Зробити аналіз нормативних документів України щодо правової охорони ПЗ. Оформити позовну заяву на факт порушення авторських прав.
11	Охорона патентного права: Розробити порядок оформлення патенту на корисну модель.
12	Авторське право на програмний продукт: Оформити заявку на отримання авторського права на тестовий програмний продукт.
13	Охорона комерційної таємниці та нових технологічних об'єктів: Дати характеристику проблемі збереження права інтелектуальної власності в мережі Інтернет. Охарактеризувати правовий захист цифрових систем поширення.
Семестр 2.	
1	Основи тестування програмного забезпечення: Ключові питання: ефективність тестування; проблема оракула; теоретичні і практичні обмеження тестування; придатність до тестування. Зв'язок тестування з іншими видами діяльності. Розробка управляючого тестування.
2	Види тестування: Навантажувальне та стрес-тестування. Порівняльне тестування. Тестування спроможності системи на відновлення. Конфігураційне тестування. Тестування зручності і простоти використання.
3	Поняття і структура UML: Механізми розширення UML (стереотипи, метасвойствам, обмеження, коментарі). Діаграми профілю, їх елементи та зв'язку. Область їх застосування. Робота в Umbrello UML Modeller. Імпорт і генерація кода.

	Додаткові можливості.
4	Проведення формальних інспекцій: Завдання та цілі проведення формальних інспекцій. Етапи формальної інспекції та ролі її учасників. Документування процесу формальної інспекції. Оцінка відтестованості проекту: метрики і методика інтегральної оцінки
5	Формальні інспекції програмного коду: Формальні інспекції проектної документації.
6	Модульне тестування: Завдання та цілі модульного тестування. Поняття модуля та його кордонів. Тестування класів. Підходи до проектування тестового оточення. Організація модульного тестування.
7	Інтеграційне та системне тестування: Системне тестування Приймально-здавальні та сертифікаційні випробування при розробці програмного забезпечення, що сертифікується.
8	Тестування для користувача інтерфейсу: Методи розробки стійкого коду. Класифікація проблем, що виникають при роботі програмних систем.

8. СИСТЕМА ОЦІНЮВАННЯ ТА ВИМОГИ

Загальна система оцінювання курсу	<p>За семестр з курсу дисципліни проводяться два періодичні контролі (ПКР), результати яких є складником результатів контрольних точок першої (КТ1) і другої (КТ2). Результати контрольної точки (КТ) є сумою поточного (ПК) і періодичного контролю (ПКР): $КТ = ПК + ПКР$. Максимальна кількість балів за контрольну точку (КТ) складає 50 балів. Максимальна кількість балів за періодичний контроль (ПКР) становить 60 % від максимальної кількості балів за контрольну точку (КТ), тобто 30 балів. А 40 % балів, тобто решта балів контрольної точки, є бали за поточний контроль, а саме 20 балів. Результати поточного контролю обчислюються як середньозважена оцінок ($X_{ср}$) за діяльність студента на практичних (семінарських) заняттях, що входять в число певної контрольної точки. Для трансферу середньозваженої оцінки ($X_{ср}$) в бали, що входять до 40 % балів контрольної точки (КТ), треба скористатися формулою: $ПК = (X_{ср}) * 20 / 5$. Таким чином, якщо за поточний контроль (ПК) видів діяльності студента на всіх заняттях $X_{ср} = 4.1$ бали, які були до періодичного контролю (ПКР), то їх перерахування на 20 балів здійснюється так: $ПК = 4.1 * 20 / 5 = 4.1 * 4 = 16.4 // 16$ (балів). За періодичний контроль (ПКР) студентом отримано 30 балів. Тоді за контрольну точку (КТ) буде отримано $КТ = ПК + ПКР = 16 + 30 = 46$ (балів).</p> <p>Студент має право на підвищення результату тільки одного періодичного контролю (ПКР) протягом двох тижнів після його складання у випадку отримання незадовільної оцінки. Якщо підсумковим контролем вивчення дисципліни є диференційований або недиференційований залік, то набраних таким чином 60 і більше балів достатньо для його зарахування.</p>
--	---

	<p>Якщо підсумковим контролем є екзамен, на його складання надається 100 балів за виконання тестів (або задач чи завдань іншого виду). Загальний рейтинг з дисципліни (ЗР) складається з суми балів (Е), отриманих на екзамені, і підсумкової оцінки (ПО) та ділиться навпіл. $ZP = (ПО + E) / 2$</p>
Практичні заняття	<p>«5» – студент в повному обсязі володіє навчальним матеріалом, вільно самостійно та аргументовано його викладає під час усних виступів та письмових відповідей, глибоко та всебічно розкриває зміст теоретичних питань та практичних завдань, використовуючи при цьому нормативну, обов'язкову та додаткову літературу. Правильно вирішив усі розрахункові / тестові завдання. Здатен виділяти суттєві ознаки вивченого за допомогою операцій синтезу, аналізу, виявляти причинно-наслідкові зв'язки, формувати висновки і узагальнення, вільно оперувати фактами та відомостями.</p> <p>«4» – студент достатньо повно володіє навчальним матеріалом, обґрунтовано його викладає під час усних виступів та письмових відповідей, в основному розкриває зміст теоретичних питань та практичних завдань, використовуючи при цьому нормативну та обов'язкову літературу. Але при викладанні деяких питань не вистачає достатньої глибини та аргументації, допускаються при цьому окремі несуттєві неточності та незначні помилки. Правильно вирішив більшість розрахункових / тестових завдань. Студент здатен виділяти суттєві ознаки вивченого за допомогою операцій синтезу, аналізу, виявляти причинно-наслідкові зв'язки, у яких можуть бути окремі несуттєві помилки, формувати висновки і узагальнення, вільно оперувати фактами та відомостями.</p> <p>«3» – студент в цілому володіє навчальним матеріалом, викладає його основний зміст під час усних виступів та письмових розрахунків, але без глибокого всебічного аналізу, обґрунтування та аргументації, допускаючи при цьому окремі суттєві неточності та помилки. Правильно вирішив половину розрахункових / тестових завдань. Має ускладнення під час виділення суттєвих ознак вивченого; під час виявлення причинно-наслідкових зв'язків і формулювання висновків.</p> <p>«2» – студент не в повному обсязі володіє навчальним матеріалом. Фрагментарно, поверхово (без аргументації та обґрунтування) викладає його під час усних виступів та письмових розрахунків, недостатньо розкриває зміст теоретичних питань та практичних завдань, допускаючи при цьому суттєві неточності. Правильно вирішив окремі розрахункові / тестові завдання. Безсистемно відділяє випадкові ознаки вивченого; не вміє зробити найпростіші операції аналізу і синтезу; робити узагальнення, висновки.</p> <p>«1» – студент виконав менше половини завдання лабораторної роботи або не виконав зовсім; під час усних відповідей не розкриває зміст теоретичних питань та практичних завдань. Не відповідає на елементарні питання.</p>
Умови допуску до підсумкового контролю	<p>Студент, який навчається стабільно на «відмінні» оцінки і саме такі оцінки має за періодичні контролю, накопичує впродовж вивчення навчального курсу 90 і більше балів, має право не складати екзамен з даної дисципліни.</p> <p>Студент зобов'язаний відпрацювати всі пропущені семінарські заняття протягом двох тижнів. Невідпрацьовані заняття (невиконання навчального плану) є підставою для недопущення студента до підсумкового контролю.</p>

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового	для заліку

діяльності		проекту (роботи), практики	
90 – 100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

9. РЕКОМЕНДОВАНА ЛІТЕРАТУРА ТА ІНФОРМАЦІЙНІ РЕСУРСИ

ОСНОВНА ЛІТЕРАТУРА

1. Андон Ф.И., Коваль Г.И., Коротун Т.М., Лаврищева Е.М., Суслов В.Ю Основы инженерии качества программных систем. 2-е изд., перераб. и доп. К.: Академперіодика, 2007. 672 с.
2. Бичківський Р.В., Столярчук П.Г. Методологія, стандартизація, управління якістю і сертифікація: Підручник. Львів : Львівська політехніка, 2004. 560 с.
3. Богданов Д.В., Фильчаков В.В. Стандартизация жизненного цикла и качества программных средств. СПбГУАП. СПб, 2000. 210 с.
4. Буч Г., Рамбо Д., Якобсон А. Язык UML. Руководство пользователя. Москва : ДМК, 2006. 496 с.
5. Гапотій В.Д., Солоненко А.М. Захист права інтелектуальної власності в Україні: навч. посіб. Мелітополь: ТОВ "Видавничий будинок ММД", 2011. 360 с.
6. ДСТУ 3918-1999 Інформаційні технології. Процеси життєвого циклу програмного забезпечення. 57 с.
7. ДСТУ ISO/IEC TR15504 Інформаційні технології. Оцінювання процесів життєвого циклу програмних засобів. – 315 с.
8. Криспин Л., Грегори Дж. Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд. Москва : «Вильямс», 2010. – 464 с.
9. Молодцова О.П. Управління якістю програмної продукції: Навч. посібник. Київ: КНЕУ, 2001, 248 с.
10. Саранча Г.А. Методологія, стандартизація, відповідність, акредитація та управління якістю: Підручник. Київ : Центр навчальної літератури, 2006. 672 с.
11. Сидорчук О.В., Ковалишин О.В., Городецький І.М. Стандартизація та сертифікація техніки і обладнання. Львів : Львівський ДАУ, 2007. 189 с.
12. Тамре Л. Введение в тестирование программного обеспечения.: Пер. с англ. Москва : Издательский дом «Вильямс», 2003. 368 с.
13. Шаповал М.І. Основи стандартизації, управління якістю і сертифікації: Підручник 3-є вид., пероб. і доп. Київ : Видавництво Європейського університету, 2001. 174 с.
14. Якість програмного забезпечення та тестування: Підручник / укладачі: М.О. Сидоров, М.А. Безверха. Київ : НАУ, 2010. 282 с.
15. Янушкевич Д.А., Коваль О.А. Національна та міжнародна стандартизація. Харків : ХНАДУ, 2010. 237 с.

ДОДАТКОВА ЛІТЕРАТУРА

1. David Rubinstein, Standish Group Report: There's Less Development Chaos Today. 2007.
2. Kent Beck, and others, Manifesto for Agile Software Development 2001.
3. M. Pomeroy-Huff, J. Mullaney, R. Cannon, M. Sebern, The Personal Software Process (PSP) Body of Knowledge version 1.0, SPECIAL REPORT CMU/SEI, 2005.
4. MSF, Microsoft, Microsoft Solutions Framework, Отдел MSF, Microsoft, 2002.
5. PMBOK. Руководство к Своду знаний по управлению проектами, 3-е изд., PMI, 2004.
6. Watts S. Humphrey, The Team Software Process (TSP), Technical Report CMU/SEI, 2000.
7. Державна система сертифікації. К.: Держстандарт України, 1994.
8. Койфман Ю.І. та ін. Принципи, методи та досвід роботи у сфері забезпечення якості і сертифікації: система якості, правила сертифікації. Посібник. Львів, 1995. 348 с.
9. Руководство к своду знаний по программной инженерии. The Guide to the Software Engineering Body of Knowledge, SWEBOK, IEEE Computer Society Professional Practices Committee, 2004.
10. ДСТУ 1.1:2015 Національна стандартизація. Стандартизація та суміжні види діяльності. Словник термінів (ISO/IEC Guide 2:2004, MOD)
11. ДСТУ 1.2:2015 Національна стандартизація. Правила проведення робіт з національної стандартизації
12. ДСТУ 2296-93 Державна система сертифікації. Знак відповідності. Форма, розміри, технічні вимоги та правила застосування
13. ДСТУ 2296-93. Національний знак відповідності. Форма, розміри, технічні вимоги та правила застосування.
14. ДСТУ 2462-94. Сертифікація. Основні поняття. Терміни та визначення.
15. ДСТУ 2844-94. Програмні засоби ЕОМ. Забезпечення якості. Терміни та визначення.
16. ДСТУ 2850-94. Програмні засоби ЕОМ. Показники та методи оцінювання.
17. ДСТУ 2851-94. Програмні засоби ЕОМ. Документування результатів випробувань.
18. ДСТУ 2853-94. Програмні засоби ЕОМ. Підготовлення і проведення випробувань.
19. ДСТУ 2938-94. Системи оброблення інформації. Основні поняття. Терміни та визначення.
20. ДСТУ 3329-96. Інформаційні технології. Система стандартів з баз даних. Концепція та термінологія для концептуальної схеми й інформаційної бази.
21. ДСТУ EN ISO /IEC 17067:2014 Оцінка відповідності. Основні положення сертифікації продукції та керівні вказівки щодо схем сертифікації продукції.
22. ДСТУ ISO/IEC 12119-2003 Інформаційні технології. Пакети програм. Тестування і вимоги до якості.
23. ДСТУ ISO/IEC 12207:2014 Інженерія систем і програмного забезпечення. Процеси життєвого циклу програмного забезпечення
24. ДСТУ ISO/IEC 14764:2014 Інженерія програмного забезпечення. Процеси життєвого циклу програмного забезпечення. Технічне обслуговування
25. ДСТУ ISO/IEC 15288:2014 Інженерія систем і програмного забезпечення. Процеси життєвого циклу систем
26. ДСТУ ISO/IEC 15289:2014 Інженерія систем і програмного забезпечення. Контент життєвого циклу інформаційної продукції (документації)
27. ДСТУ ISO/IEC 15939:2008 Інженерія систем і програмних засобів. Процес вимірювання
28. ДСТУ ISO/IEC 16085:2016 Інженерія систем і програмних засобів. Процеси життєвого циклу. Керування ризиками
29. ДСТУ ISO/IEC 17000:2007 Оцінювання відповідності. Словник термінів і загальні принципи.
30. ДСТУ ISO/IEC 25040:2015 Інженерія програмних засобів і систем. Вимоги щодо якості та оцінювання систем і програмного продукту (SQuaRE). Процес оцінювання
31. ДСТУ ISO/IEC 25041:2015 Інженерія програмних засобів і систем. Вимоги щодо

якості та оцінювання систем і програмного продукту (SQuaRE). Настанова щодо оцінювання для розробників, замовників та незалежних експертів

32. ДСТУ ISO/IEC 25045:2015 Інженерія програмних засобів і систем. Вимоги щодо якості та оцінювання систем і програмного продукту (SQuaRE). Модуль оцінювання відновлюваності

33. ДСТУ ISO/IEC 25051:2015 Програмна інженерія. Вимоги до якості та оцінювання систем і програмного продукту (SQuaRE). Вимоги до якості готового до застосування програмного продукту та інструкції щодо їхнього тестування

34. ДСТУ ISO/IEC 25060:2015 Інженерія програмних засобів і систем. Вимоги щодо якості та оцінювання систем і програмного продукту (SQuaRE). Загальний промисловий формат (CIF) для оцінювання зручності використання програмних продуктів. Загальні принципи для інформації щодо оцінювання зручності використання програмних продуктів

35. ДСТУ ISO/IEC 29155-2:2015 Розроблення систем і програмного забезпечення. Платформи для тестування проєктів з розроблення інформаційних систем. Частина 2. Вимоги щодо тестування продуктивності

36. ДСТУ ISO/IEC 33063:2015 Інформаційні технології. Оцінювання процесу. Модель оцінювання процесу для тестування програмного забезпечення

37. ДСТУ ISO/IEC 9126-1:2013 Програмна інженерія. Якість продукту'. Частина 1. Модель якості

38. ДСТУ ISO/IEC Guide 67:2008 Оцінювання відповідності. Засади сертифікації продукції.

39. ДСТУ ISO/IEC TR 24748-3:2016 Інженерія систем і програмного забезпечення. Керування життєвим циклом. Частина 3. Настанова щодо застосування ISO/IEC 12207 (Процеси життєвого циклу програмного забезпечення)

40. ДСТУ ISO/IEC TR 9126-2:2008 Програмна інженерія. Якість продукту. Частина 2. Зовнішні метрики

41. ДСТУ ISO/IEC TR 9126-3:2012 Програмна інженерія. Якість продукту. Частина 3. Внутрішні метрики

42. ДСТУ ISO/IEC TR 9126-4:2012 Програмна інженерія. Якість продукту. Частина 4. Метрики якості під час використання

43. ДСТУ ISO/IEC/IEEE 29119-1:2015 Розроблення систем і програмного забезпечення. Тестування програмного забезпечення. Частина 1. Поняття і визначення.

44. ДСТУ ISO/IEC/IEEE 29119-2:2015 Розробка систем і програмного забезпечення. Тестування програмного забезпечення. Частина 2. Тестові процеси

45. ДСТУ ISO/IEC/IEEE 29119-3:2015 Розроблення систем і програмного забезпечення. Тестування програмного забезпечення. Частина 3. Тестова документація ДСТУ ISO/IEC 29155-1:2015 Розроблення систем і програмного забезпечення. Платформи для тестування проєктів з розроблення інформаційних систем. Частина 1. Концепції та визначення

46. ДСТУ ISO/IEC 25000:2015 Інженерія програмних засобів і систем. Вимоги щодо якості та оцінювання систем і програмного продукту (SQuaRE).

47. ДСТУ ISO/IEC 25045:2016 Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (SQuaRh). Модуль оцінювання відповідності

48. ДСТУ ISO/IEC 25062:2015 Програмна інженерія. Вимоги щодо якості та оцінювання систем і програмного продукту (SQuaRE). Загальний промисловий формат для звітів з тестування зручності використання

49. ДСТУ ISO/IEC 90003:2006 Програмна інженерія. Настанови щодо застосування ISO 9001:2000 до програмного забезпечення

50. ДСТУ ISO/IEC TR 24774:2016 Інженерія систем і програмних засобів. Керування життєвим циклом. Настанови щодо опису процесу

ІНФОРМАЦІЙНІ РЕСУРСИ

1. Дистанційний курс «Тестування програмного забезпечення». URL: <http://do.mdpu.org.ua/course/view.php?id=145>.
2. Закон України від 02.10.1992 № 2657-XII «Про інформацію» URL: <http://zakon3.rada.gov.ua/laws/show/2657-12>.
3. Закон України від 05.06.2014 № 1315-VII «Про стандартизацію». URL: <http://zakon3.rada.gov.ua/lavvs/show/1315-18>.
4. Закон України від 05.07.1994 № 80/94-ВР «Про захист інформації в інформаційно-телекомунікаційних системах» URL: <http://zakon3.rada.gov.ua/laws/show/80/94-%D0%B2%D1%80>.
5. Закон України від 15.01.2015 № 124-УІІ «Про технічні регламенти та оцінку відповідності». URL: <http://zakon5.rada.gov.ua/laws/show/124-19>.
6. Професія тестировщик ПО <http://tester.com.ua>.
7. Тестирование программного обеспечения - основные понятия и определения. URL: <http://www.znannya.org/?view=software-testing-testing>
8. Тестирование Программного Обеспечения с Михаилом Портновым 1. URL: <http://www.youtube.com/watch?v=wYnayqdP9Ck>.
9. Тестирование Программного Обеспечения с Михаилом Портновым 2. URL: <http://www.youtube.com/watch?v=fobdjVEEIQw>.
10. Тестирование Программного Обеспечения с Михаилом Портновым 3. URL: <http://www.youtube.com/watch?v=o7skGBsaUm4>.
11. Тестирование Программного Обеспечения с Михаилом Портновым 4. URL: <http://www.youtube.com/watch?v=IlwFD7gISOw>.
12. Тестирование Программного Обеспечения с Михаилом Портновым 5. URL: <http://www.youtube.com/watch?v=q7hqFZHPwJI>.
13. Форум тестировщиков. URL: <http://software-testing.ru/forum>.