

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра теорії та технології програмування

«ЗАТВЕРДЖУЮ»

Заступник декана  
з навчальної роботи

\_\_\_\_\_ Кашпур О.Ф.

«\_\_\_» \_\_\_\_\_ 2017 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
ВАЛІДАЦІЯ ТА ВЕРИФІКАЦІЯ  
ПРОГРАМНИХ СИСТЕМ**

для студентів

галузь знань **12 «Інформаційні технології»**  
(шифр і назва)  
спеціальність **122 «Комп'ютерні науки»**  
(шифр і назва спеціальності)  
освітній рівень **магістр**  
(молодший бакалавр, бакалавр, магістр)  
освітня програма **«Інформатика»**  
(назва освітньої програми)  
вид дисципліни за вибором ВНЗ

Форма навчання	<b>денна</b>
Навчальний рік	<b>2017/2018</b>
Семестр	<b>2</b>
Кількість кредитів ECTS	<b>3</b>
Мова викладання, навчання та оцінювання	<b>українська</b>
Форма заключного контролю	<b>екзамен</b>

Викладачі: **д.ф.-м.н., проф. Дорошенко А.Ю.** (лекції),  
**к.п.н., асистент Русіна Н.Г.** (лабораторні заняття)

Пролонговано: на 20\_\_/20\_\_ н.р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_» 20\_\_ р.  
(підпис, ПІБ, дата)

на 20\_\_/20\_\_ н.р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_» 20\_\_ р.  
(підпис, ПІБ, дата)

**КИЇВ – 2017**

Розробник: Дорошенко Анатолій Юхимович, д.ф.-м.н., професор кафедри «Теорії та технології програмування»

ЗАТВЕРДЖЕНО

В.О. Зав. кафедри «Теорії та технології програмування»

\_\_\_\_\_ (Панченко Т.В.)  
(підпис) (прізвище та ініціали)

Протокол № \_\_\_\_ від « \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

---

Протокол від « \_\_\_\_ » \_\_\_\_\_ 20\_\_ року № \_\_\_\_

Голова науково-методичної комісії \_\_\_\_\_ (Хусаїнов Д.Я.)  
(підпис) (прізвище та ініціали)

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ року

## ВСТУП

**1. Мета дисципліни** – засвоєння техніки перевірки правильності програм шляхом побудови моделей програм з подальшим доведенням властивості побудованих моделей, що характеризують її правильність, а також техніки тестування програмних систем в основних його видах.

**2. Попередні вимоги до опанування або вибору навчальної дисципліни (за наявності):**

1. *Знати:* базові поняття математичної логіки і теорії алгоритмів і математичного аналізу, формальних моделей алгоритмів та складності алгоритмів.

2. *Вміти:* будувати логічні специфікації програм з урахуванням встановлених вимог, виконувати верифікаційні процедури з перевірки правильності та валідності програмного забезпечення, а також проводити основні види тестування програм.

3. *Володіти засобами:* розробки тестів у середовищі Visual Studio.

**3. Анотація навчальної дисципліни:**

Нормативна навчальна дисципліна "Верифікація і валідація програмних систем" є складовою циклу професійної підготовки фахівців освітньо-кваліфікаційного рівня "магістр" галузі знань 12 „Інформаційні технології” зі спеціальності 122 „Комп’ютерні науки”, освітньо-професійної програми – „Інформатика”. Цей курс потрібен для глибшого вивчення нормативних дисциплін магістерської програми з системного програмування та інформаційних технологій, а також низки спецкурсів відповідного напрямку.

Дана дисципліна є обов’язковою навчальною дисципліною за *програмою “Інформатика”*.

Викладається у 2 семестрі 1 курсу магістерської програми в обсязі – 120 год.

*(4 кредитів ECTS)* зокрема: лекції – 26 год., лабораторні – 12 год., консультації – 2 год., самостійна робота – 80 год. У курсі передбачено 2 змістових модулів та 2 модульні контрольні роботи. Завершується дисципліна – **екзаменом у 2 семестрі**.

В результаті вивчення навчальної дисципліни студент повинен:

**знати** – місце і значення верифікації і валідації програмних систем в життєвому циклі програмних систем; основні етапи розробки програмних систем, методи інженерії вимог до програмного забезпечення; класифікацію відмов і помилок у програмних системах, основні поняття та методи тестування програмних систем, методи доведення правильності програм, логічні засоби специфікації програм та методи перевірки на моделях, процеси, артефакти та стандарти верифікації програмного забезпечення.

**вміти** – будувати логічні специфікації програм з урахуванням встановлених вимог; виконувати верифікаційні процедури з перевірки правильності та валідності програмного забезпечення, а також проводити основні види тестування програм.

Дисципліна "Верифікація і валідація програмних систем" є необхідною для засвоєння всіх інших курсів та спецкурсів магістерської програми.

#### 4. Завдання (навчальні цілі):

набуття знань, умінь та навичок, формування компетентностей на рівні новітніх досягнень у інформатиці, відповідно до кваліфікації магістр з інформаційних технологій, відбувається через здатності:

- визначати необхідний метод верифікації і валідації програмних систем в життєвому циклі програмної системи;
- застосовувати методи інженерії вимог до програмного забезпечення;
- правильно класифікувати відмови і помилки у програмних системах;
- застосовувати методи доведення правильності програм;
- застосовувати логічні засоби специфікації програм та методи перевірки на моделях;
- правильно планувати та використовувати методи тестування програмних систем;
- використовувати процеси, артефакти та стандарти верифікації програмного забезпечення.

#### 5. Результати навчання за дисципліною:

Результат навчання (РН) (1. знати; 2. вміти; 3. комунікація*; 4. автономність та відповідальність*)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
РН 1.1	Знати необхідні методи верифікації і валідації програмних систем в життєвому циклі програмної системи	Лекція	Тест, 60% правильних відповідей	16%
РН 2.1	Вміти застосовувати методи інженерії вимог до програмного забезпечення	Лекція		
РН 2.2	Вміти правильно класифікувати відмови і помилки у програмних системах	Лекція	Тест, 60% правильних відповідей	22%
РН 2.3	Вміти застосовувати методи доведення правильності програм	Лекція		
РН 1.2	Знати логічні засоби специфікації програм та методи перевірки на моделях	Лекція		
РН 2.4	Вміти планувати та використовувати методи тестування програмних систем	Лабораторна робота, самостійна робота	Захист лабораторних робіт	44%
РН 1.3	Знати процеси, артефакти та стандарти верифікації програмного забезпечення	Лекція	Тест, 60% правильних відповідей	18%

\* заповнюється за необхідністю, наприклад для практик, лабораторних курсів тощо.

## 6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	РН 1.1	РН 2.1	РН 2.2	РН 2.3	РН 1.2	РН 2.4	РН 1.3
<b>Програмні результати навчання</b> <i>(з опису освітньої програми)</i>							
ПР9. Використовувати інструментальні засоби розробки алгоритмів верифікації, проектувати концептуальні, логічні та фізичні моделі програм, розробляти та оптимізувати їх реалізацію.		+				+	
ПР10. Створювати розподілені реалізації систем верифікації, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування.	+			+	+		
ПР11. Володіти навичками використання методології управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти готувати проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, креативний бриф, угоду, договір, контракт та ін.).		+	+	+	+		+

## 7. Схема формування оцінки.

### 7.1 Форми оцінювання студентів:

#### - семестрове оцінювання:

1. Контрольна робота 1 (тест): РН 1.1., РН 1.2 — 6 балів/5 балів.
2. Контрольна робота 2 (тест): РН1.3, - 6 балів/5 балів.
3. Лабораторна робота 1 (проект): РН 1.1 – 8 балів/5.8 балів.
4. Лабораторна робота 2 (проект): РН 1.2 – 8 балів/5.8 балів.
5. Лабораторна робота 3 (проект): РН 2.1 – 8 балів/5.8 балів.
6. Лабораторна робота 4 (проект): РН 2.2 – 8 балів/5.8 балів.
7. Лабораторна робота 5 (проект): РН 2.3 – 8 балів/5.8 балів.
8. Лабораторна робота 6 (проект): РН 2.4 – 8 балів/5.8 балів.

#### - підсумкове оцінювання (у формі екзамену):

- максимальна кількість балів які можуть бути отримані студентом: 40 балів;
- результати навчання які будуть оцінюватись: РН1.1, РН1.2, РН1.3, РН2.1, РН2.2, РН2.3, РН2.4;
- форма проведення: письмова
- види завдань:

#### Структура екзаменаційної роботи та критерії оцінювання:

1. Теоретичне запитання з етапів життєвого циклу (РН1.1).
2. Теоретичне запитання з логічних засобів специфікації програм (РН1.2).
3. Теоретичне запитання з методів перевірки на моделях (РН1.2).
4. Теоретичне запитання процеси, артефакти та стандарти верифікації програмного забезпечення (РН1.3).
5. Письмове завдання з методів інженерії вимог (РН2.1).
6. Письмове завдання з класифікації відмов і помилок (РН2.2).
7. Письмове завдання з методів доведення правильності програм (РН2.3).
8. Письмове завдання з методів планування та проведення тестування програмних систем (РН2.4).

#### Критерії оцінювання екзаменаційної роботи

Завдання	Види завдань	Максимальний відсоток за завдання (бал)	Всього відсотків (балів)
Завдання 1, 2, 3, 4,	Письмове запитання з курсу	по 8,5% (6 балів)	36% (16 балів)
Завдання 5, 6,7,8	Письмове завдання	По 16% (6 балів)	64% (24 балів)
Всього			100% (40 балів)

## Запитання для підготовки до екзамену

1. Програмний проект та його тріада
2. Проблема розробка програмних проектів.
3. Основні категорії розробки ПЗ.
4. Методології розробки програмного забезпечення MSF
5. Методології розробки програмного забезпечення RUP
6. Методології розробки програмного забезпечення XP
7. Поняття життєвого циклу ПЗ. Послідовні та ітераційні методології розробки
8. Методи збирання та аналізу вимог до ПЗ
9. Розробка вимог до ПЗ
10. Специфікація вимог до ПЗ
11. Сценарний підхід до розробки вимог
12. Завдання верифікації та валідації ПЗ
13. Класифікація відмов та помилок у ПЗ
14. Причини появи помилок на етапах ЖЦ
15. Основні поняття та методи тестування програм
16. Процеси тестування програмних систем
17. Засоби автоматизації процесу тестування в середовищі MS Visual Studio
18. Засоби формальної специфікації програм
19. Моделювання станів програми за допомогою моделей Кріпке
20. Методи створення моделі першого порядку з тексту програми
21. Методи доведення правильності програм
22. Логічні засоби специфікацій програм
23. Специфікація поведінки програм засобами темпоральної логіки
24. Перевірка формул на моделі
25. Метод двійкових діаграм рішень для подання булевих функцій
26. Подання моделей Кріпке засобами двійкових діаграм
27. Символьна верифікація моделей
28. Процеси верифікації ПЗ
29. Верифікація різних артефактів життєвого циклу ПЗ
30. Інспекційні методи верифікації ПЗ
31. Методи статичного аналізу ПЗ
32. Динамічні методи верифікації ПЗ
33. Формальні методи верифікації
34. Синтетичні методи верифікації ПЗ

**Для отримання загальної позитивної оцінки з дисципліни оцінка за екзамен не може бути меншою 24 балів.**

**Студент не допускається до екзамену, якщо під час семестру набрав менше ніж 20 балів. Студент допускається до екзамену за умови виконання 70% передбачених планом лабораторних робіт.**

### 7.2 Організація оцінювання:

#### Терміни проведення форм оцінювання:

1. Виконання лабораторних робіт 1-4: до 7 тижня семестру.
2. Виконання лабораторних робіт 5-10 (тест): до 15 тижня семестру.
3. Контрольна робота 1: до 6 тижня семестру.
4. Контрольна робота 2: до 15 тижня семестру.

Студент має право на одне перескладання контрольної роботи з можливістю отримання не більше 80% попередньо визначених балів за роботу. Термін перескладання визначається викладачем.

У випадку відсутності студента з поважних причин відпрацювання та перездачі МКР здійснюються у відповідності до „Положення про порядок оцінювання знань студентів при кредитно-модульній системі організації навчального процесу” від 1 жовтня 2010 року.

У разі неякісного виконання лабораторної роботи, викладач має право не зарахувати лабораторну роботу, або знизити за неї бали.

Студент має право здавати лабораторні роботи після закінчення визначеного для них терміну, але з втратою 10% балів за кожен тиждень, який пройшов з моменту закінчення терміну її здачі.

### 7.3 Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74

<b>Незадовільно / Fail</b>	0-59
<b>Зараховано / Passed</b>	60-100
<b>Не зараховано / Fail</b>	0-59



## 8. Структура навчальної дисципліни. Тематичний план лекцій і лабораторних занять

№ п/п	Назва теми	Кількість годин		
		Лекції	Лабораторні	Самостійна робота
<b>Змістовий модуль 1. Життєвий цикл розробки програмного забезпечення (ПЗ)</b>				
1	<p><b>Тема 1. Проблема розробки програмного програмного забезпечення.</b> – 2 год.</p> <p>Програмний проект та його тріада. Проблема розробка програмних проектів. Основні категорії розробки ПЗ. [1,6]. <b>Завдання для самостійної роботи</b> (2 год.)</p> <p>Основні питання управління програмними проектами. [1,4]</p>	2		6
2	<p><b>Тема 2. Життєвий цикл ПЗ та методології розробки його життєвого циклу.</b> – 2 год.</p> <p>Поняття життєвого циклу ПЗ. Послідовні та ітераційні методології розробки ПЗ [1,4,6].</p> <p><b>Лабораторна робота 1. (2 год.)</b> <b>Завдання для самостійної роботи</b> (2 год.)</p> <p>Вивчення особливостей та порівняльна характеристика різних методологій розробки програм (MSF,RUP,XP) [1, 4, 6].</p>	2	2	4
3	<p><b>Тема 3. Інженерія вимог до ПЗ– 2 год.</b></p> <p>Структура специфікації вимог згідно міжнародних стандартів щодо ПЗ [1,6].</p> <p><b>Завдання для самостійної роботи</b> (2 год.)</p> <p>Вивчення різних підходів до моделювання вимог (моделі станів, зміни станів та поведінки) [1,4].</p>	2		6
	<i>Модульна контрольна робота 1</i>			
Модульний контроль за підсумками лабораторної роботи 1				
Всього по модулю 1		6	2	16
<b>Змістовий модуль 2. Методи перевірки правильності та тестування програм</b>				
4	<p>Тема 4. Поняття верифікації і валідації програмного забезпечення – 2 год.</p> <p>Місце верифікації серед процесів розробки програмного забезпечення. Завдання верифікації та валідації ПЗ [1,3,6]. <b>Завдання для самостійної роботи</b> (2 год.)</p> <p>Вивчення проблеми забезпечення якості ПЗ та основних положень міжнародних стандартів щодо якості програмного забезпечення. [1,2,6].</p>	2		6
5	<p>Тема 5. Класифікація відмов та помилок у ПЗ. – 2 год.</p> <p>Класифікація помилок ПЗ: помилки, дефекти та відмови. [1]. <b>Завдання для самостійної роботи</b> (2 год.)</p> <p>Категорії дефектів розробки програм. Причини появи помилок на етапах ЖЦ [1,3]</p>	2		6
6	<p>Тема 6. Основні поняття та методи тестування програм – 2 год. [3]</p> <p><b>Лабораторна робота 2. (2 год.)</b> <b>Завдання для самостійної роботи</b> (2 год.)</p> <p>Вивчення статичних та динамічних методів тестування програм [3].</p>	2	2	7
7	<p><b>Тема 7. Формальне моделювання програмних систем - (2 год.)</b></p> <p>Засоби специфікації програм. Формальне моделювання</p>	2		6

	програмних систем [2]. <b>Завдання для самостійної роботи (4 год.)</b> Методи створення моделі першого порядку з тексту програми			
8	<b>Тема 8. Методи доведення правильності програм - (2 год.)</b> Метод індуктивних тверджень Флойда-Наура та метод структурної індукції Хоара. [1] <b>Лабораторна робота 3. (2 год.)</b> <b>Завдання для самостійної роботи (4 год.)</b> Метод Дейкстри формального доведення правильності програм [1].	2	2	7
9	<b>Тема 9. Логічні засоби специфікацій програм - (2 год.)</b> Темпоральні логіки, логіка дерев обчислень CTL* [2]. <b>Лабораторна робота 4. (2 год.)</b> <b>Завдання для самостійної роботи (4 год.)</b> Повнота операторів логіка дерев обчислень CTL* [2]..	2	2	6
10	<b>Тема 10. Перевірка формул на моделі - (2 год.)</b> Верифікація моделей для логіки дерев обчислень CTL. [2]. Завдання для самостійної роботи (2 год.) Алгоритми верифікації моделей для логіка дерев обчислень CTL [2].	2		6
11	<b>Тема 11. Двійкові діаграми рішень - (2 год.)</b> Ефективне подання булевих функцій та метод двійкових діаграм рішень [2]. <b>Лабораторна робота 5. (2 год.)</b> <b>Завдання для самостійної роботи (2 год.)</b> Подання моделей Кріпке засобами двійкових діаграм [2].	2	2	7
12	<b>Тема 12. Процеси верифікації ПЗ - (2 год.)</b> Процеси, артефакти та стандарти верифікації ПЗ [3]. <b>Лабораторна робота 6. (2 год.)</b> <b>Завдання для самостійної роботи (2 год.)</b> Завдання верифікації в ЖЦ ПЗ. Верифікація і валідація ПЗ [3].	2	2	7
13	<b>Тема 13. Інспекційні методи верифікації ПЗ - (2 год.)</b> Класифікація методів верифікації. Експертизи ПЗ. [1,3] <b>Завдання для самостійної роботи (2 год.)</b> Практика застосування експертиз ПЗ [3].	2		8
	<i>Модульна контрольна робота 2</i>			
	Модульний контроль за підсумками лабораторних робіт 2 - 6			
	Всього по модулю 2	16	10	64
	Консультація		2	
	Екзамен			
	<b>ВСЬОГО</b>	<b>26</b>	<b>12</b>	<b>80</b>

**Загальний обсяг 120 год.**, в тому числі:

Лекцій – 26 год.

Лабораторні заняття - 12 год.

Консультації – 2 год.

Самостійна робота - 80 год.

## 9. Перелік лабораторних робіт

1. Модульні тести (unit tests)
2. Веб-тести (web tests)
3. Тести навантаження (load tests)
4. Впорядковані тести (ordered tests)

5. Тести, залежні від даних (data-driven tests)
6. Розробка на основі тестів (test-driven development)
7. Тестування на основі перевірки станів (state verification) та поведінки (behavior verification)
8. Тестування з використанням підміни об'єктів (mock objects)
9. Mock object frameworks (на прикладі Rhino Mocks)

## 10. Рекомендовані джерела:

### *Основна*

1. Л.П. Бабенко, К.М. Лавріщева. Основи програмної інженерії. К.: "Знання", 2001.- 269 с.
2. Э. М. Кларк, О. Грамберг, Д. Пелед, Верификация моделей программ: Model Checking. Пер. с англ. под ред. Р. Смелянского.-- Издательство Московского центра непрерывного математического образования, Москва 2002, 416 с.
3. С.В. Сеницын, Н.Ю. Налютин, Верификация программного обеспечения, Интернет-Университет Информационных Технологий, [www.INTUIT.ru](http://www.INTUIT.ru)

### *Додаткова:*

4. М. Кантор, Управление программными проектами. - М.: ИД «Вильямс», 2002.- 176 с.
5. И. Соммервилл, Инженерия программного обеспечения. М.: ИД «Вильямс», 2002.- 624 с.
6. С.А. Орлов, Технологии разработки программного обеспечения.- СПб: «Питер», 2002.- 464.
7. Microsoft Visual Studio 2005-2008, [www.microsoft.com](http://www.microsoft.com) .

## 11. Завдання для самостійної роботи з елементами дистанційного навчання

з дисципліни «Валідація та верифікація програмних систем»  
на період з 24 січня до 28 лютого 2018 р.

для студентів

I курсу

рівня магістр

освітньої програми «Інформатика»

викладач-лектор: д.ф.-м.н., проф. Дорошенко А.Ю. (e-mail: [a.doroshenko@knu.ua](mailto:a.doroshenko@knu.ua))

викладач, що проводить лабораторні заняття:

к.п.н., асист. Русіна Н.Г. (e-mail: [rusina@knu.ua](mailto:rusina@knu.ua)).

### *Види та форми контрольних заходів з перевірки самостійної роботи студентів, критерії оцінювання*

Контроль за виконанням самостійної роботи студентами здійснюється у двох формах:

- у січні-лютому за допомогою електронних засобів (електронною поштою та навчальним середовищем moodle <https://dl.knu.ua/course/view.php?id=8320>);
- у березні – шляхом проведення письмової контрольної роботи та здачі лабораторних робіт.

Додаткові матеріали для самостійної роботи студентів розміщені за посиланнями:  
[https://drive.google.com/drive/u/2/folders/165\\_hIdbZxVOp3VFhbtQZMqYyOvn3K8Rj](https://drive.google.com/drive/u/2/folders/165_hIdbZxVOp3VFhbtQZMqYyOvn3K8Rj)  
<https://dl.knu.ua/course/view.php?id=8320>.

Контроль у січні-лютому 2018 р. відбувається у два етапи.

#### **Перший етап**

24 січня – 1 лютого 2018 р. - студенти мають ознайомитись з лекцією (ім'я файлу 1-вступ.pdf).

2 лютого – 6 лютого 2018 р. - студенти мають ознайомитись з лекцією (ім'я файлу 5-Основні поняття та методи тестування програмних систем.pdf) та вибрати тему лабораторної роботи 1 для виконання.

7 лютого – 14 лютого 2018 р. студенти мають ознайомитись з лекцією (ім'я файлу 6-Тести програм і систем.pdf) та вибрати тему лабораторної роботи 1 для виконання.

Під час **першого етапу** для підтвердження ознайомлення з навчальним матеріалом студенти повинні надіслати відповіді на тестові завдання (**додаток 1**), а також розгорнуту відповідь на одне з теоретичних питань викладачу, що проводить лекційні заняття **Дорошенку А.Ю.** на електронну пошту (зазначено вище) не пізніше **14 лютого 2018 р.** Також на першому етапі студенти мають обрати собі 2 теми лабораторних робіт для виконання завдання другого етапу самостійної роботи.

На **другому етапі** самостійної роботи (15 лютого – 28 лютого 2018 р.) кожен студент має опанувати питання винесені на самостійну роботу на поглибленому рівні. Підтвердженням його роботи в межах цього етапу є виконання звіту до лабораторних робіт виконаних протягом другого етапу. Загальну умову лабораторної роботи наведено у **додатку 2**. Уточнення умови лабораторної роботи **обов'язково** погоджується з викладачем, що веде лабораторні заняття не пізніше **10 лютого**.

Викладач оцінює виконані завдання в категоріях «зараховано» або «не зараховано». Щоб отримати оцінку «зараховано» потрібно правильно відповісти на 51% і більше тестових питань, надати розгорнуту відповідь на теоретичне питання, а також надіслати звіти (**додаток 3**) по лабораторним роботам:

1. На e-mail викладача, що веде лекційні заняття надіслати файл (у форматі rtf) з заповненою тестовою формою та розгорнуту відповідь на теоретичне питання.
2. На e-mail викладача, що веде лабораторні заняття надіслати файл-звіт1 та файл-звіт2 по лабораторним роботам.

Якщо студент отримає оцінку «не зараховано», у нього є час до **20 лютого (тест)** до **27 лютого (звіт)** переробити завдання та надіслати їх викладачу повторно.

**Виконання першого етапу самостійної роботи (тестові завдання та одне теоретичне питання) є допуском до другого етапу.**

**Виконання другого етапу самостійної роботи (виконання звіту до лабораторної роботи) є допуском до написання контрольної роботи у березні 2018 р.**

**Якщо відповіді на питання та звіт до лабораторної роботи здані невчасно без поважних причин, або не зараховані, студент втрачає можливість написання контрольної роботи та отримання відповідних модульних балів без можливості перескладання.**

На контрольну роботу за підсумками самостійної роботи виносяться теоретичні питання. Робота оцінюється максимум в **10 балів**. Вона включає в себе 5 тестових питань з проблематики, винесеної на самостійну роботу, та одне теоретичне питання. Правильна відповідь на кожне тестове завдання оцінюється в 1 бал. За розгорнуту відповідь на теоретичне питання студент може отримати від 1 до 5 балів.

Критерії оцінювання відповіді студента на теоретичне питання:

- повнота розкриття питання 1-2 бали;
- логіка викладення 1 бал;
- аналітичні міркування 1-2 бали.

Контрольна робота проводиться на першому лекційному занятті з курсу у березні 2018 р. Її тривалість – 1 академічна година.

### **Список основної рекомендованої літератури для виконання самостійної роботи**

#### ***Основна***

1. Л.П. Бабенко, К.М. Лавріщева. Основи програмної інженерії. К.: "Знання", 2001.- 269 с.
2. Э. М. Кларк, О. Грамберг, Д. Пелед, Верификация моделей программ: Model Checking. Пер. с англ. под ред. Р. Смелянского.-- Издательство Московского центра непрерывного математического образования, Москва 2002, 416 с.
3. С.В. Сеницын, Н.Ю. Налютин, Верификация программного обеспечения, Интернет-Университет Информационных Технологий, [www.INTUIT.ru](http://www.INTUIT.ru)

#### ***Додаткова:***

4. М. Кантор, Управление программными проектами. - М.: ИД «Вильямс», 2002.- 176 с.
5. И. Соммервилл, Инженерия программного обеспечения. М.: ИД «Вильямс», 2002.- 624 с.
6. С.А. Орлов, Технологии разработки программного обеспечения.- СПб: «Питер», 2002.- 464.
7. Microsoft Visual Studio 2005-2008, [www.microsoft.com](http://www.microsoft.com) .



*II. Дайте розгорнуту відповідь на одне з питань*

- 1. Що включає в себе метод аналізу інтерфейсів**
- 2. Засоби документування результатів тестування**

*\*Виконане завдання першого етапу необхідно надіслати на електронну пошту [a.doroshenko@knu.ua](mailto:a.doroshenko@knu.ua) не пізніше 14 лютого 2018 р.*

**Перелік тем лабораторних робіт  
з дисципліни «Верифікація і валідація програмних систем»**

1. Модульні тести (unit tests)
2. Веб-тести (web tests)
3. Тести навантаження (load tests)
4. Впорядковані тести (ordered tests)
5. Тести, залежні від даних (data-driven tests)
6. Розробка на основі тестів (test-driven development)
7. Тестування на основі перевірки станів (state verification) та поведінки (behavior verification)
8. Тестування з використанням підміни об'єктів (mock objects)
9. Mock object frameworks (на прикладі Rhino Mocks)

**Звіт до лабораторної роботи повинен включати наступні елементи:**

- *Титульний аркуш.*
- *Зміст.*
- *Вступ: актуальність розробки проекту за обраним варіантом лабораторної роботи), методи дослідження, цілі та завдання.*
- *Розділ 1. Огляд використаних технологій.*
- *Розділ 2. Призначення та цілі створення тестового інструментарію.*
- *Розділ 3. Реалізація проекту (опис загальної структури, можливі принципові фрагменти коду, рисунки).*
- *Список використаних джерел (в тексті звіту обов'язкова наявність посилань на усі джерела; джерела мають бути оформлені за правилами, які розміщені за посиланням <http://vak.in.ua/do.php>).*
- *Додатки (наприклад, діаграми, фрагменти коду, рисунки, принтскрін; в тексті обов'язкова наявність посилань на усі додатки).*

Оформлення звіту наступне: *інтервал* між рядками 1,5; *шрифт* Times New Roman; *кегель* 14; *вирівнювання тексту* – по ширині. Рекомендований обсяг звіту: 5-8 сторінок.

Звіт до першої лабораторної роботи має бути надісланий викладачу, що проводить лабораторні заняття не пізніше **21 лютого 2018 р.**

Звіт до другої лабораторної роботи має бути надісланий викладачу, що проводить лабораторні заняття не пізніше **27 лютого 2018 р.**

Викладач оцінює звіт в категоріях «зараховано» або «не зараховано». Викладач повідомляє студенту електронною поштою, чи зарахований його звіт. Якщо звіт не зарахований, викладач вказує недоліки. **Не допускається виконання одного звіту більш одним студентом академічної групи.** Ідентичні звіти до лабораторних робіт отримують оцінку «не зараховано», студенти мають повторно виконати звіти до лабораторних робіт. У разі не зарахування звіту лабораторної роботи у студента є час до **27 лютого** для його доопрацювання з урахуванням висловлених викладачем зауважень.

**За погодженням з викладачем лабораторних занять завдання може бути змінено.**