

**Одеська національна академія зв'язку ім. О. С. Попова»
Кафедра комп'ютерних наук**

Назва курсу	Моделювання та верифікація програмного забезпечення
Викладачі	д.т.н., доц. Шмельова Тетяна Рудольфівна
Контактний телефон	705 02 45
E-mail:	t.shmeleva@onat.edu.ua
Консультації	ауд. 485, Пн. 14.00-15.00

1. Анотація до курсу

Навчальна дисципліна «Моделювання та верифікація програмного забезпечення» призначена для підготовки студентів другого ступеня вищої освіти за спеціальністю «121. *Інженерія програмного забезпечення*».

У цьому курсі подано баланс теорії та практики моделювання й верифікації програмного забезпечення, зокрема паралельного та розподіленого. Викладені методи являють собою основу для розробки надійного програмного забезпечення. Більшість відомостей про надійність сучасних програм ґрунтується на методах тестування, які гарантують певну вірогідність виконання програмою заданої функції. Формальне доведення коректності програм є наступним кроком підвищення надійності програмного забезпечення для спеціальних застосувань у системах реального часу, а також життєво критичних сферах.

2. Мета та цілі курсу

Метою викладання навчальної дисципліни «Моделювання та верифікація програмного забезпечення» є формування знань основних термінів і понять математичного апарату моделювання та верифікація програмного забезпечення, вивчення теоретичних та практичних основ моделювання програмного забезпечення, принципів та основних методів верифікації програмного забезпечення, набуття практичних навичок доведення коректності прикладних алгоритмів, набуття умінь, необхідних для подальшої наукової та професійної діяльності.

Цілі курсу:

- оволодіння основними поняттями формальних засобів доведення коректності програм, зокрема відповідності програм їхнім специфікаціям, поданих у вигляді додаткових анотацій до тексту програми;
- набуття фахових навичок формалізації комп'ютерних програм у вигляді сіток Петрі та алгебри процесів, дослідження їхніх властивостей доведення коректності програм;
- набуття практичних навичок побудови основних моделей послідовних, паралельних та розподілених програм;
- ознайомлення з сучасними програмними засобами верифікації послідовних, паралельних та розподілених програм з використання методів теорії сітей Петрі та алгебри процесів ;
- набуття практичних навичок моделювання та верифікації програм сітками Петрі та алгеброю процесів.

3. Формат курсу

Навчальний курс «Моделювання та верифікація програмного забезпечення» реалізується як цикл лекційних і лабораторних занять, які знайомлять студентів з теоретичними основами та засобами моделювання, верифікації розподілених програм, їхніми можливими практичними реалізаціями та застосуванням у процесі вирішення реальних завдань. У межах курсу студенти повинні здобути навички та компетенції для вирішення прикладних завдань моделювання та верифікації програмного забезпечення.

Під час навчання студенти магістратури відвідують лекційні та семінарські заняття, виконують лабораторні роботи з використанням комп'ютерного обладнання і програмного забезпечення. Студенти мають змогу отримувати індивідуальні консультації. Навчальний курс містить самостійну роботу студентів, що може бути забезпечена наявною обчислювальною технікою в комп'ютерних аудиторіях навчального закладу та використанням електронної бази даних кафедри.

4. Результати навчання

Вивчення навчальної дисципліни «Моделювання та верифікація програмного забезпечення» передбачає формування та розвиток у студентів компетентностей:

загальних:

- здатність до абстрактного мислення, аналізу та синтезу;
- здатність спілкуватися іноземною мовою як усно, так і письмово;
- здатність проведення теоретичних та прикладних досліджень на відповідному рівні;
- здатність мотивувати людей та рухатися до спільної мети, працювати в команді співробітників;
- здатність спілкуватися з представниками інших професійних груп різного рівня (з експертами з інших галузей знань/видів економічної діяльності);
- здатність удосконалювати свої навички на основі аналізу попереднього досвіду.
- здатність генерувати нові ідеї (креативність);
- здатність управляти своїм часом та розуміти важливість дедлайнів.

фахових:

- здатність аналізувати предметні галузі, формувати, аналізувати та моделювати вимоги до програмного забезпечення;
- здатність ідентифікувати, класифікувати та описувати проєктні завдання, знаходити раціональні методи й підходи до їх розв'язання;
- здатність проєктувати програмне забезпечення, включаючи проведення моделювання його архітектури, поведінки та процесів функціонування окремих підсистем і модулів;
- здатність розвивати й реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення;
- здатність оцінювати ступінь обґрунтованості застосування специфікацій, стандартів, правил і рекомендацій в професійній галузі та дотримуватися їх при реалізації процесів життєвого циклу програмного забезпечення;
- здатність ефективно керувати фінансовими, людськими, технічними та іншими проєктними ресурсами.
- здатність систематизувати професійні знання щодо створення й супроводження програмного забезпечення.
- здатність розробляти й координувати процеси, фази та ітерації життєвого циклу програмних систем на основі застосування відповідних моделей, методів та технологій розробки програмного забезпечення;

- здатність застосовувати й розвивати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення з використанням та ризику знань теоретичних і практичних основ методології системного аналізу, методів формалізації системних завдань, що мають суперечливі цілі, невизначеності;
- здатність використовувати знання сучасних комп'ютерних інформаційних технологій та інструментів інженерних і наукових досліджень, розрахунків, обробки та аналізу даних, моделювання та оптимізації;
- здатність демонструвати та застосовувати на практиці знання методів моделювання програмних систем, оцінювання їхньої ефективності та якості.

Результати навчання цієї дисципліни деталізують такі **програмні результати навчання**:

- знати й системно застосовувати методи аналізу та моделювання прикладної галузі, виявлення інформаційних потреб і збору вихідних даних для проєктування програмного забезпечення;
- оцінювати й вибирати методи та моделі розробки, впровадження, експлуатації програмних засобів та управління ними на всіх етапах життєвого циклу;
- розробляти й оцінювати стратегії проєктування програмних засобів, в тому числі з урахуванням впливу факторів різновекторного спрямування; обґрунтовувати, аналізувати й оцінювати прийняті проєктні рішення з точки зору якості кінцевого програмного продукту;
- аналізувати, оцінювати й вибирати методи, сучасні програмно-апаратні інструментальні та обчислювальні засоби, технології, алгоритмічні та програмні рішення для ефективного виконання конкретних виробничих завдань з програмної інженерії;
- обґрунтовано вибирати парадигми й мови програмування для вирішення прикладних завдань; застосовувати на практиці системні та спеціалізовані засоби, компонентні технології (платформи) та інтегровані середовища розробки програмного забезпечення;
- проводити аналітичне дослідження параметрів функціонування програмних систем для їхньої валідації та верифікації, а також проводити аналіз обраних методів, засобів автоматизованого проєктування та реалізації програмного забезпечення;
- знати й застосовувати сучасні професійні стандарти, а також інші нормативно-правові документи з інженерії програмного забезпечення;
- вміти приймати організаційно-управлінські рішення в умовах невизначеності;
- набувати нові наукові й професійні знання, вдосконалювати навички, прогнозувати розвиток програмних систем та інформаційних технологій;
- формулювати, експериментально підтверджувати, обґрунтовувати й застосовувати на практиці в процесі розробки програмного забезпечення конкурентоспроможні ідеї, методи, технології вирішення професійних, науково-технічних завдань в умовах невизначеності;
- здобувати необхідну інформацію з іншомовної літератури, аналізувати та вибирати необхідні для вирішення фахових наукових і прикладних завдань інформаційно-довідкові та науково-технічні ресурси й джерела знань з урахуванням сучасних досягнень науки й техніки, здійснювати різні види комунікації;
- організовувати командну роботу, управляти проєктами, підбирати команду проєкту, ефективно працювати в групі, визначати та розподіляти завдання з метою вирішення різноманітних дослідницьких та практичних завдань;
- проєктувати програмні засоби із використанням оптимальних сполучень багатоконпонентних стратегій.

5. Обсяг курсу

Вид заняття	лекції	практичні заняття	лабораторні заняття	самостійна робота
К-сть годин	20	-	20	80

6. Ознаки курсу

Рік викладання	Семестр	Спеціальність	Курс, (рік навчання)	Нормативний\вибірковий
2020-2021	1.2	Інженерія програмного забезпечення	1	н

7. Технічне й програмне забезпечення /обладнання

Студенти здобувають практичні навички програмної реалізації керованої моделлю парадигми, основних принципів та алгоритмів аналізу коректності паралельних та розподілених обчислень на основі методів теорії сітей Петрі та алгебри процесів. При виконанні лабораторних робіт використовуються моделюючі системи Tina, mCRL2 ([http://projects.laas/fr/tina](http://projects.laas.fr/tina), <https://www.mcrl2.org>) в середовищі операційної системи Linux, сучасне відкрите програмне забезпечення та моделі, розташовані в середовищі системи GitHub.

Методичне забезпечення курсу: презентації та конспект лекцій, комплекс навчально-методичного забезпечення дисципліни, нормативні документи.

8. Політики курсу

З метою успішного проходження курсу та складання контрольних заходів студенти зобов'язані:

- не запізнюватися на заняття;
- не пропускати аудиторні заняття (у разі хвороби потрібно надати викладачу довідку або її ксерокопію);
- опрацьовувати весь лекційний матеріал та інформаційні ресурси;
- конструктивно підтримувати зворотній зв'язок з викладачем на всіх етапах проходження курсу;
- своєчасно й самостійно виконувати всі передбачені програмою лабораторні завдання;
- не користуватися мобільним телефоном під час аудиторних занять;
- брати очну участь у контрольних заходах;
- будь-яке відтворення результатів чужої праці, в тому числі використання завантажених з Інтернету матеріалів, як власних результатів, кваліфікується, як порушення норм і правил академічної доброчесності, та передбачає притягнення до відповідальності у порядку, визначеному чинним законодавством.

У разі відсутності заборгованостей та складанні модульних контрольних робіт студент допускається до іспиту.

9. Схема курсу

Тиж. / год.	Тема, план, короткі тези	Форма діяльності (заняття) / Формат	Матеріали	Завдання (в тому числі тематика лабораторної роботи), год	Вага оцінки	Термін виконання
Тиж. 1-6 12 акад. год. 1	Змістовий модуль 1. Інструментальні засоби верифікації послідовних та паралельних програм алгоритмічною мовою. Тема 1. Вступ до предмету. Огляд інструментальних засобів верифікації за допомогою формального доведення теорем стосовно властивостей програм. Статична та динамічна верифікація	Лекція1 <i>F2F</i>	Презентація	Передивитись презентацію, 2 год	1	1 тиж.
2		Лабораторна1	Презентація	Виконати завдання лабораторної роботи: «Статична верифікація програми без додаткових анотацій в системі Splint». 2 год	5	2
3	Тема 2. Система верифікації програм Splint. Системи VCC та Helix QAC для верифікації програм мовою C/C++ анотованих додатковими специфікаціями.	Лекція2 <i>F2F</i>	Презентація	Передивитись презентацію, 2 год	1	1 тиж.
4		Лабораторна2	Презентація	Виконати завдання лабораторної роботи: «Розробка анотацій та верифікація програм в системі VCC». 2 год	5	
5	Тема 3. Засоби анотації функцій, інваріантів даних та циклів системи VCC. Формальне	Лекція3 <i>F2F</i>	Презентація	Передивитись презентацію, 2 год	1	1 тиж.

Тиж. / год.	Тема, план, короткі тези	Форма діяльності (заняття) / Формат	Матеріали	Завдання (в тому числі тематика лабораторної роботи), год	Вага оцінки	Термін виконання
	доведення відповідності програми її специфікаціям					
6		Лабораторна 3	Презентація	Виконати завдання лабораторної роботи: «Верифікація заданої програми з розробкою анотацій, функцій, інваріантів даних та циклів». 2 год	5	
Тиж. 7-15 16 акад. год. 7	Змістовий модуль 2. Моделювання та верифікація паралельного програмного забезпечення сітьми Петрі (СП). Тема 4. Подання основних типів процесів та ресурсів (змінних програми) СП.	Лекція4 <i>F2F</i>	Презентація	Передивитись презентацію, 2 год	1	1 тиж.
8		Лабораторна 4	Презентація	Виконати завдання лабораторної роботи: «Розробка моделі паралельної програми в формі СП». 2 год	5	
9	Тема 5. Основні властивості моделі СП ідеальної програми. Поведінкові та структурні методи дослідження властивостей СП.	Лекція 5 <i>F2F</i>	Презентація	Передивитись презентацію, 2 год	1	1 тиж.

Тиж. / год.	Тема, план, короткі тези	Форма діяльності (заняття) / Формат	Матеріали	Завдання (в тому числі тематика лабораторної роботи), год	Вага оцінки	Термін виконання
10		Лабораторна 5	Презентація	Виконати завдання лабораторної роботи: «Налагодження моделі паралельної програми в моделюючій системи Tіпа». 2 год	5	
11	Тема 6. Нескінченні СП як засіб моделювання програмного забезпечення комп'ютерних мереж, ґридів та обчислювальних хмар. Засоби скінченної специфікації та основні методи властивостей у параметричній формі.	Лекція 6 <i>F2F</i>	Презентація	Передивитись презентацію, 2 год	1	1 тиж.
12		Лабораторна 6	Презентація	Виконати завдання лабораторної роботи: «Верифікація програм шляхом дослідження поведінкових та структурних властивостей». 2 год	5	
13	Тема 7. Модальні та темпоральні логіки як засіб специфікації властивостей програм. Мова LTL, μ-числення.	Лекція7 <i>F2F</i>	Презентація	Передивитись презентацію, 2 год	1	1 тиж.

Тиж. / год.	Тема, план, короткі тези	Форма діяльності (заняття) / Формат	Матеріали	Завдання (в тому числі тематика лабораторної роботи), год	Вага оцінки	Термін виконання
14		Лабораторна 7	Презентація	Виконати завдання лабораторної роботи: «Дослідження простору станів моделі з використанням запитів поданих мовою LTL». 2 год	5	
Тиж. 15-20 12 акад. год. 15	Змістовий модуль 3. Алгебра та числення процесів як засіб моделювання та верифікації розподілених програм Тема 8. Вступ до алгебри та числення процесів: дії, поведінка, еквівалентність та абстракції. Нотації Хоара та Беркстри. Основні типи даних.	Лекція 8 <i>F2F</i>	Презентація	Передивитись презентацію, 2 год	1	1 тиж.
16		Лабораторна 8	Презентація	Виконати завдання лабораторної роботи: «Розробка моделі розподіленої програми алгеброю процесів». 2 год	5	
17	Тема 9. Організація послідовних та паралельних процесів в алгебрі процесів. Лінеаризація послідовних та паралельних процесів, лінійне рівняння процесів та методи його розв'язання.	Лекція 9 <i>F2F</i>	Презентація	Передивитись презентацію, 2 год	1	1 тиж.

Тиж. / год.	Тема, план, короткі тези	Форма діяльності (заняття) / Формат	Матеріали	Завдання (в тому числі тематика лабораторної роботи), год	Вага оцінки	Термін виконання
18		Лабораторна 9	Презентація	Виконати завдання лабораторної роботи: «Перетворення моделей, побудова та графічне подання простору станів». 2 год	5	
19	Тема 3. Модальні μ-числення як засіб специфікації запитів стосовно властивостей моделі. Логіка Хенесі-Мілнера, регулярні формули та модальності. Модальні формули з даними	Лекція 10 <i>F2F</i>	Презентація	Передивитись презентацію, 2 год	1	1 тиж.
20		Лабораторна 10	Презентація	Виконати завдання лабораторної роботи: «Організація процесів пошуку властивостей в ширину та глибину». 2 год	5	

**10. Система оцінювання та вимоги
Розподіл балів, які отримують студенти**

Поточне тестування та самостійна робота				Контрольні завдання та підсумковий екзамен	Сума
Змістовний модуль № 1				40	100
T1	T2		T3		
6	6		6		
Змістовний модуль № 2					
T4	T5	T6	T7		
6	6	6	6		
Змістовний модуль № 3					
T8	T9		T10		
6	6		6		

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
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

11. Рекомендована література

Базова

1. The VCC Manual Working draft, version 0.2, July 10, 2015, VCC User Manual (working draft, ver. 0.2)
2. A Practical Verification Methodology for Concurrent Programs, February 12, 2009, Technical Report MSR-TR-2009-15 Microsoft Research, Microsoft Corporation, One Microsoft Way Redmond, WA 98052 12p
3. 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.
4. Zaitsev D.A. Clans of Petri Nets: Verification of protocols and performance evaluation of networks, LAP LAMBERT Academic Publishing, 2013, 292 p.
5. D. A. Zaitsev, I. D. Zaitsev and T. R. Shmeleva. Infinite Petri Nets: Part 1, Modeling Square Grid Structures. Complex Systems, 26 (2), 2017, pp.157-195.
6. J.F. Groote and M.R. Mousavi. Modeling and Analysis of Communicating Systems. The MIT Press, 2014.
7. Петрик М.Р. Моделювання програмного забезпечення : науково-методичний посібник / М.Р. Петрик, О.Ю. Петрик – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2015. – 200 с.
8. Г.В.Табунщик. Проектування та моделювання програмного забезпечення сучасних інформаційних систем Навч. посібник / Г.В.Табунщик, Т.І.Каплієнко, О.А.Петрова. – Запоріжжя, 2016. – 259 с.
9. Marieke Huisman, Vladimir Klebanov, Rosemary Monahan. VerifyThis 2012 A Program Verification Competition Int J Softw Tools Technol Transfer (2015) 17:647–657, DOI 10.1007/s10009-015-0396-8.

Додаткова

1. Berthomieu, B., Ribet, O.-P., Vernadat, F.: “The tool TINA-construction of abstract state space for Petri nets and time Petri nets,” Int. J. Prod. Res. 42(14), 2004, 2741–2756.
2. T.A.N. Engels, J.F. Groote, M.J. van Weerdenburg and T.A.C. Willemse. “Search algorithms for automated validation.” Journal of Logic and Algebraic Programming 78(4):274-287, 2009.
3. Введение в формальные методы верификации программ: учебное пособие / А. С. Камкин. – Москва: МАКС Пресс, 2018. – 272 с.
4. Shmeleva T.R. Verification of network protocols using Finite Automata and Petri Nets. Methodical instructions for practical and laboratory works on course «Modeling and Optimization of Telecommunication Systems and Networks» / T.R.Shmeleva. – Odessa: A.S. Popov ONAT, 2015. – 32 p.
5. Д.А.Зайцев. Математичні моделі дискретних систем. Навчальний посібник з дисципліни .– Одеса, ОНАЗ, 2003, 35с.

12. Інформаційні ресурси

1. MicrosoftVCC <https://rise4fun.com/vcc>. [Електронний ресурс]: [Веб-сайт].
2. mCRL2 <https://www.mcr12.org/>. [Електронний ресурс]: [Веб-сайт].
3. Tina [http://projects.laas/fr/tina](http://projects.laas.fr/tina). [Електронний ресурс]: [Веб-сайт].
4. Національна бібліотека України імені В. І. Вернадського [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Київ: НБУВ, 2013-2015. – Режим доступу: www.nbuv.gov.ua – Назва з екрана.
5. Електронний каталог Національної парламентської бібліотеки України [Електронний ресурс]: [політемат. база даних містить відом. про вітчизн. та зарубіж. кн., брош., що надходять у фонд НПБ України]. – Електронні дані (803 438 записів). – Київ: Нац. парлам. б-ка України, 2002-2015. – Режим доступу: catalogue.nplu.org . – Назва з екрана.
6. Український інститут інтелектуальної власності [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Київ: УІПВ, 2017. – Режим доступу: <http://www.uipv.org> – Назва з екрана.