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

ФОРМАЛЬНІ МЕТОДИ РОЗРОБКИ 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	обов'язкова

Форма навчання	**денна**
Навчальний рік **2017/2018**	
Семестр	**2**
Кількість кредитів ECTS	**4**
Мова викладання, навчання та оцінювання **українська**	
Форма заключного контролю	**екзамен**

Викладач: д**.ф.-м.н., проф. Нікітченко М.С.** (лекції)

Пролонговано: на 20__/20__ н.р. _____(_____) «__»___ 20__р.
(підпис, ПІБ, дата)

на 20__/20__ н.р. _____(_____) «__»___ 20__р.
(підпис, ПІБ, дата)

КИЇВ – 2017

Розробник: Нікітченко Микола Степанович, 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 основних концепцій, принципів та понять сучасних методів розробки програмних систем та їх застосування для адекватного моделювання мов специфікацій і програмування та використання побудованих моделей для створення сучасних програмних та інформаційних систем високої якості.

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

   1. *Знати:* теорію програмування, формальні моделі програм, математичну логіку, методи проектування, розробки та тестування програм;
   2. *Вміти:* розробляти специфікації програм з урахуванням встановлених вимог;.
   3. *Володіти елементарними навичками:* програмування в сучасних мовах, перевірки виконуваності формул.

3. **Анотація навчальної дисципліни**:

Навчальна дисципліна «Формальні методи розробки програмних систем» є складовою освітньо-професійної програми підготовки фахівців за освітньо-кваліфікаційним рівнем «магістр» галузі 12 „Інформаційні технології” зі *спеціальності 122* „Комп’ютерні науки”, *освітньо-професійної програми* – „Інформатика”.

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

Викладається в 2 семестрі 1 курсу магістратури в обсязі 120 годин.

(4 кредити ECTS)) зокрема: *лекції – 38 год., консультації – 2 год., самостійна робота – 80 год.* У курсі передбачено **2** *змістових модулі* та 2 *модульні контрольні роботи*. Завершується дисципліна – **екзаменом в 2 семестрі.**

В результаті вивчення навчальної дисципліни студент повинен:

знати: основні поняття програмування, методи формалізації мов програмування та мов специфікацій, методи моделювання предметних областей; логічні числення.

вміти: формалізувати мові специфікацій та програм, моделювати предметні області за допомогою відповідних мов, застосувати програмні засоби аналізу специфікацій.

Дисципліна «Формальні методи розробки програмних систем» є базовою для засвоєння інших курсів та спецкурсів *спеціальності 122* „Комп’ютерні науки”.

4. **Завдання (навчальні цілі):**

набуття знань, умінь та навичок, формування компетентностей на рівні новітніх досягнень у програмуванні, відповідно до кваліфікації магістра з комп'ютерних наук:

- знання та розуміння предметної області та методів їх формалізації;
- застосовувати логічні числення у практичних ситуаціях;
- здатність вчитися й оволодівати сучасними знаннями;
- здатність генерувати нові ідеї (креативність);
- здатність розробляти й управляти проектами;
- здатність оцінювати та забезпечувати якість виконуваних робіт.

Контроль знань і розподіл балів, які отримують студенти.

Контроль здійснюється за модульно-рейтинговою системою.

У змістовий модуль 1 (ЗМ1) входять теми 1 - 9, у змістовий модуль 2 (ЗМ2) – теми 10 - 19. Обов'язковим для іспиту в 2 семестрі є отримання студентом протягом 2 семестру не менше 32 балів.

Оцінювання за формами контролю:

	ЗМ1		***ЗМ2***	
	Min. – 16 балів	*Max. – 28 бали*	*Min. – **16** балів*	*Max. – **32** балів*
Модульна контрольна робота	16	20	16	20
Активна робота	0	8	0	12
„3" – мінімальна/максимальна оцінку, яку може отримати студент. [1] – мінімальна/максимальна залікова кількість робіт чи завдань.				

Для студентів, які набрали сумарно меншу кількість балів ніж *критично-розрахунковий мінімум – 32 балів* для допуску до іспиту обов'язково повинні перескласти контрольні роботи та здати заплановані лабораторні роботи. Студент має право на одне перескладання контрольної роботи. 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.

При простому розрахунку отримаємо:

	Змістовий модуль1	Змістовий модуль2	Іспит	Підсумкова оцінка
Мінімум	16	16	*28*	*60*
Максимум	**28**	**32**	**40**	**100**

При цьому, кількість балів:

- **1-34** відповідає оцінці «незадовільно» з обов'язковим повторним вивченням дисципліни;
- **35-59** відповідає оцінці «незадовільно» з можливістю повторного складання;
- **60-64** відповідає оцінці «задовільно» («достатньо»);
- **65-74** відповідає оцінці «задовільно»;
- **75 - 84** відповідає оцінці «добре»;
- **85 - 89** відповідає оцінці «добре» («дуже добре»);
- **90 - 100** відповідає оцінці «відмінно».

Шкала відповідності *(за умови іспиту)*

За 100 – бальною шкалою	За національною шкалою	
90 – 100		відмінно
85 – 89		добре
75 – 84		
65 – 74		задовільно
60 – 64		
35 – 59		не задовільно
1 – 34		

Шкала відповідності *(за умови заліку)*

За 100 – бальною шкалою	За національною шкалою
90 – 100	Зараховано
85 – 89	
75 – 84	
65 – 74	
60 – 64	
1 – 59	не зараховано

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Змістовий модуль 1 Формальні моделі програм та методи їх специфікації

Тема 1. Вступ до предмету. Основні методологічні принципи побудови формальних моделей програм. – 2 год.

Основні методологічні принципи побудови понять програмування. Принципи розвитку та гносеологічності. Принцип єдності теорії та практики. Принцип побудови сутнісної теорії розробки програмних систем. Пентада основних понять програмування та їх властивості. Головні аспекти програм. Синтаксис та семантика програм. [1,4,5]

Тема 2. Принципи формалізації програмних понять. Інтенсіонал та екстенсіонал понять множини, функції та програми. – 2 год.

Теоретико-функціональна та теоретико-номінатна платформи формалізації програмних понять. Інтенсіонал поняття, екстенсіонал поняття. Класи не детермінованих функцій. Формалізація композицій. Формалізація програм. Композиційні системи, дескриптивні системи, денотаційні системи. Рівні абстракції даних та функцій. [1,2,4,5]

Тема 3. Мови специфікації та програмування.– 2 год.

Особливості мов специфікацій у порівнянні з мовами програмування. Денотативні композиції в мовах специфікацій. Логічні композиції у мовах специфікацій. [1–5]

Тема 4. Формальні моделі обчислюваних функцій.– 2 год.

Традиційні формальні моделі обчислюваних функцій та їх обмеженість. Моделі абстрактної обчислюваності. Аксіоматичні визначення обчислюваних функцій. Номінативні дані та їх класифікація. Базові функції. Моделювання натуральних чисел номінативними даними. Композиції номінативних функцій. Дані скінченої структури. Функції натуралізації та денатуралізації. Схема натуральної обчислюваності. Повні класи обчислюваних функцій. [1,4,5]

Тема 5. Предметні області та методи їх опису. Методи розробки програм.– 2 год.

Предметні області, їх властивості, рівні розгляду. Огляд основних методів розробки програм – RAISE, B, Z, ASM, TLA. [1–5,7–13]

Тема 6. Структури даних та класи функцій у мовах специфікацій.– 2 год.

Структури даних у мовах специфікацій: множини, списки, кортежі, мапи, номінати. Властивості типів, базові функції та предикати. Класи функцій у мовах специфікацій. Однозначні та багатозначні функції. Подання та перетворення функцій. [1,3,7–13]

Тема 7. Класи композицій у мовах специфікацій.– 2 год.

Класи композицій у мовах специфікацій. Методи подання та перетворення композицій. [1–3]

Тема 8. Методи уточнення даних, функцій та композицій.– 2 год.

Уточнення даних на підставі схеми Хоара. Приклади. Доведення властивостей уточнених даних. Уточнення функцій для даних різного типа абстракції. Приклади. Доведення властивостей уточнених функцій. Методи уточнення композицій. [1,6,7,10]. Уточнення композицій для даних та функцій різного типа абстракції. Приклади. Доведення властивостей уточнених композицій. [1–5,7–13]

Тема 9. Приклади специфікацій програмних систем.– 2 год.

Приклади специфікації програмних систем: банкомати. [3,11]

Консультація (1 год).

Змістовий модуль 2 Методи розробки та верифікації програм

Тема 10. Верифікація в логіках Флойда-Хоара.– 2 год.

Стани транзиційних систем. Логіки Флойда-Хоара. Аксіоматика. [6]

Тема 11. Приклади застосування. Коректність та повнота логіки.– 2 год.

Приклади специфікації програмних систем: телекомунікації. Коректність числення. Відносна повнота числення. [6]

Тема 12. Технологічні та інструментальні засоби специфікації та розробки програм за допомогою логік Флойда-Хоара.– 2 год.

Основні технологічні та інструментальні засоби специфікації та розробки програм за допомогою логік Флойда-Хоара. Система VRS. [6,10]

Тема 13. Верифікація систем в TLA. Загальні положення.– 1 год.

Основні засади методу TLA. Досяжність і безпека. Структури даних. Визначення транзиційного переходу. [13]

Тема 14. Верифікація систем в TLA. Приклади. – 2 год.

Принципи верифікації в TLA. Приклади верифікації. [13]

Тема 15. Перевірка моделей за допомогою SPIN.– 2 год.

Принципи перевірки моделей, заданих в TLA, за допомогою SPIN. [13]

Тема 16. Верифікація систем в Z.– 2 год.

Принципи перевірки моделей, заданих в Z. Приклади. Властивості Z-методу. [6,9,10]

Тема 17. Верифікація систем в B.– 2 год.

Принципи перевірки моделей, заданих в B. Приклади. Властивості B-методу.[11]

Тема 18. Верифікація систем в RAISE.– 2 год.

Принципи перевірки моделей, заданих в RAISE. Приклади. Властивості RAISE-методу.[3]

Тема 19. Програмні композиційно-номінативні логіки та їх застосування.– 2 год.

Визначення програмних композиційно-номінативних логік, їх класифікація, властивості та їх застосування. [2]

Консультація (1 год).

ТЕМАТИЧНИЙ ПЛАН ЛЕКЦІЙ ТА ЛАБОРАТОРНИХ ЗАНЯТЬ

№ лекції	Назва лекції	Кількість годин		
		Лекції	Практ занять	Сам. р-та
	Змістовий модуль 1. Формальні моделі програм та методи їх специфікації			
1	Тема 1. Вступ до предмету. Основні методологічні принципи побудови формальних моделей програм.	2		4
2	Тема 2**.** Принципи формалізації програмних понять. Інтенсіонал та екстенсіонал понять множини, функції та програми.	2		4
3	Тема 3. Мови специфікації та програмування	2		4
4	Тема 4. Формальні моделі обчислюваних функцій	2		4
5	Тема 5. Предметні області та методи їх опису. Методи розробки програм.	2		4
6	Тема 6. Структури даних та класи функцій у мовах специфікацій	2		4
7	Тема 7. Класи композицій у мовах специфікацій	2		4
8	Тема 8. Методи уточнення даних, функцій та композицій.	2		4
9	Тема 9. Приклади специфікацій програмних систем	2		6
	Консультація	1		
	Змістовий модуль 2. Методи верифікації програм			
10.	Тема 10. Верифікація в логіка Флойда-Хоара.	2		4
11.	Тема 11. Приклади застосування. Коректність та повнота логіки.	2		4
12.	Тема 12. Технологічні та інструментальні засоби специфікації та розробки програм за допомогою логік Флойда-Хоара.	2		4
13.	Тема 13. Верифікація систем в TLA. Загальні положення.	2		4
14.	Тема 14. Верифікація систем в TLA. Приклади.	2		4
15.	Тема 15. Перевірка моделей за допомогою SPIN.	2		4
16.	Тема 16. Верифікація систем в Z.	2		4
17.	Тема 17. Верифікація систем в B.	2		4
18.	Тема 18. Верифікація систем в RAISE.	2		4
19.	Тема 19. Програмні композиційно-номінативні логіки та їх застосування	2		6
	Консультація	1		
	ВСЬОГО	40		80

Загальний обсяг годин – 120, у тому числі

Лекцій – 38 год.

Консультацій – 2 год.

Самостійна робота – 80 год.

ТЕМАТИЧНО-ЗМІСТОВНА ЧАСТИНА КУРСУ

Змістовий модуль 1. Формальні моделі програм та методи їх специфікації

.

Лекція 1 (2 год). Тема 1. Вступ до предмету. Основні методологічні принципи побудови формальних моделей програм. – 2 год.

Основні методологічні принципи побудови понять програмування. Принципи розвитку та гносеологічності. Принцип єдності теорії та практики. Принцип побудови сутнісної теорії розробки програмних систем.[1,4,5]

Завдання для самостійної роботи (4 год.)

Основні поняття програмування. Формальні моделі програм. Формалізація основних понять. Методи формалізації мов програмування та специфікацій. Приклади специфікацій простих систем. [1,4,5]

Лекція 2 (2 год). Тема 2. Принципи формалізації програмних понять. Інтенсіонал та екстенсіонал понять множини, функції та програми.– 2 год.

Теоретико-функціональна та теоретико-номінатна платформи формалізації програмних понять. Інтенсіонал поняття, екстенсіонал поняття. Класи не детермінованих функцій. Формалізація композицій. Формалізація програм. Композиційні системи, дескриптивні системи, денотаційні системи. Рівні абстракції даних та функцій. [1,2,4,5]

Завдання для самостійної роботи (4 год.)

Рівні абстракції в розгляді основних понять програмування. Властивості основних понять програмування. Головні аспекти програм. Формалізми подання синтаксис та семантика програм. [1,2,4,5]. Теоретико-функціональна та теоретико-номінатна платформи формалізації програмних понять. Інтенсіонал поняття, екстенсіонал поняття. Класи не детермінованих функцій. Формалізація композицій. [1,2,4,5]. Сформулювати відмінності між теоретико-функціональною та теоретико-номінатною платформою формалізації програмних понять. Навести приклади. Надати визначення інтенсіоналу поняття та екстенсіоналу. Навести приклади. Визначити класи не детермінованих функцій. Розглянути методи формалізація композицій. [1,2, 4,5]

Лекція 3 (2 год). Тема 3. Мови специфікації та програмування.– 2 год.

Особливості мов специфікацій у порівнянні з мовами програмування. Денотативні композиції в мовах специфікацій. Логічні композиції у мовах специфікацій. [1–5]

Завдання для самостійної роботи (4 год.)

Сформулювати особливості мов специфікацій у порівнянні з мовами програмування. Визначити денотативні композиції в мовах специфікацій. Які логічні композиції використовують у мовах специфікацій? [1–5]

Лекція 4 (2 год). Тема 4. Формальні моделі обчислюваних функцій.– 2 год.

Традиційні формальні моделі обчислюваних функцій та їх обмеженість. Моделі абстрактної обчислюваності. Аксіоматичні визначення обчислюваних функцій. Номінативні дані та їх класифікація. Базові функції. Моделювання натуральних чисел номінативними даними. Композиції номінативних функцій. Дані скінченої структури. Функції натуралізації та денатуралізації. Схема натуральної обчислюваності. Повні класи обчислюваних функцій. [1,4,5]

Завдання для самостійної роботи (4 год.)

Традиційні формальні моделі обчислюваних функцій та їх обмеженість. Моделі абстрактної обчислюваності. Аксіоматичні визначення обчислюваних функцій. [1,4,5]. Розглянути традиційні формальні моделі обчислюваних функцій та їх сформулювати

випадки їх обмеженості. Перерахувати основні моделі абстрактної обчислюваності. Розглянути аксіоматичні визначення обчислюваних функцій. [1,4,5]. Номінативні дані та їх класифікація. Базові функції. Моделювання натуральних чисел номінативними даними. Композиції номінативних функцій. [1,4,5]. Дати визначення даним скінченої структури. Сформулювати визначення функцій натуралізації та денатуралізації. Розглянути схему натуральної обчислюваності. Довести теореми про повні класи обчислюваних функцій. [1,4,5]

Лекція 5 (2 год). Тема 5. Предметні області та методи їх опису. Методи розробки програм.– 2 год.

Предметні області, їх властивості, рівні розгляду. Огляд основних методів розробки програм – RAISE, B, Z, ASM, TLA. [1–5,7–13]

Завдання для самостійної роботи (4 год.) Навести приклади предметних областей, їх властивостей, рівнів розгляду. Зробити огляд основних методів розробки програм. [1–5,7–13]

Лекція 6 (2 год). Тема 6. Предметні області та методи їх опису. Методи розробки програм.– 2 год.

Предметні області, їх властивості, рівні розгляду. Огляд основних методів розробки програм – RAISE, B, Z, ASM, TLA. [1–5,7–13]

Завдання для самостійної роботи (4 год.)

Лекція 7 (2 год). Тема 7. Класи композицій у мовах специфікацій.– 2 год.

Класи композицій у мовах специфікацій. Методи подання та перетворення композицій. [1–3]

Завдання для самостійної роботи (2 год.) Класифікувати класи композицій у мовах специфікацій. Визначити коннотативні та денотативні композиції. Сформулювати їх властивості. Визначити основні методи подання та перетворення композицій. [1–3].

Лекція 8 (2 год). Тема 8. Методи уточнення даних, функцій та композицій.– 2 год.

Уточнення даних на підставі схеми Хоара. Приклади. Доведення властивостей уточнених даних. Уточнення функцій для даних різного типа абстракції. Приклади. Доведення властивостей уточнених функцій. Методи уточнення композицій. [1,6,7,10]. Уточнення композицій для даних та функцій різного типа абстракції. Приклади. Доведення властивостей уточнених композицій. [1–5,7–13]

Завдання для самостійної роботи (2 год.)

Розглянути уточнення даних на підставі схеми Хоара. Навести приклади. Довести властивості уточнених даних. [1,6] Класифікувати класи функцій у мовах специфікацій. Навести приклади використання однозначних та багатозначних функцій. Розглянути способи подання та перетворення функцій. [1–3] Класифікувати класи композицій у мовах специфікацій. Визначити коннотативні та денотативні композиції. Сформулювати їх властивості. Визначити основні методи подання та перетворення композицій. [1–3]

Лекція 9 (2 год). Тема 9. Приклади специфікацій програмних систем.– 2 год.

Приклади специфікації програмних систем: банкомати. [3,11]

Завдання для самостійної роботи (6 год.) Побудувати моделі та специфікації обраних програмних систем.

Консультація (1 год).

Типове завдання модульної роботи 1

Теоретичні питання:

1. Властивості основної пентади програмування.

2. Властивості програмної пентади.
3. Що таке часткова коректність програм? Яким чином доводиться часткова коректність програм?
4. Що таке повна коректність програм? Яким чином доводиться повна коректність програм?
5. Розкрийте зміст формалізації поняття програми.
6. Визначте різні класи функцій.
7. Визначте програмні системи різного рівня абстракції.
8. Дайте визначення класу номінативних даних.
9. Повний клас обчислюваних функцій над номінативними даними.

Самостійна робота. Побудувати моделі та специфікації наступних програмних систем:

1. Банкомат
2. Кавовий автомат
3. Склад
4. Е-магазин
5. Мікрохвильова піч
6. Телефон
7. Електроплита

Контрольні запитання до змістового модуля 1.

Які принципи є основними методологічними принципами теорії програмування?

Як формулюються принципи розвитку та гносеологічності?

Як визначається пентода основних понять програмування?

Як визначаються поняття користувача, проблеми, програми, обчислюваності, програмування?

Які властивості основних понять програмування?

Як аспекти програм є головними?

На підставі яких принципів відбувається формалізація програмних понять?

Як визначаються інтенсіональні та екстенсіональні аспекти програмних понять?

Як визначаються системи різних рівнів абстракції?

Як визначаються мови специфікацій та програмування?

Які є формальні моделі обчислюваних функцій?

Як визначається обчислюваність над складними структурами даних?

Як визначається обчислюваність над номінативними даними?

Що таке натуральна обчислюваність?

Як визначається обчислюваність композицій програм?

Повні класи обчислюваних функцій.

Якими методами описують предметні області?

Як методи використовують для специфікації вимог до програмних систем?

Які засади RAISE-методу розробки програм?

Які логічні формалізми використовують для специфікацій програм?

Як використовується класична та некласична логіка для специфікацій програм?

Як визначаються темпоральні та модальні логіки?

Як визначаються аксіоматичні методи специфікації програм?

Як визначається логіка Флойда-Хоара та які властивості вона має?

Які особливості має семантико-синтаксична технологія розробки програм?

Які особливості має метод послідовних уточнень?

Які особливості у розробку програм вносять об’єктно-орієнтовані методи?

Яка мета стандартів програмування?

Як використовують технологічні та інструментальні засоби специфікації та розробки програм?

Рекомендована література: [1–5]

Змістовий модуль 2. Методи розробки та верифікації програм

Лекція 10 (2 год). Тема 10. Верифікація в логіках Флойда-Хоара.– 2 год.

Стани транзиційних систем. Логіки Флойда-Хоара. Аксіоматика. [6]

Завдання для самостійної роботи (4 год.)

Навести приклади станів транзиційних систем. Визначити логіки Флойда-Хоара. Сформулювати їх аксіоматику. [6]

Лекція 11 (2 год). Тема 11. Приклади застосування. Коректність та повнота логіки.– 2 год.

Приклади специфікації програмних систем: телекомунікації. Коректність числення. Відносна повнота числення. [6]

Завдання для самостійної роботи (4 год.)

Розглянути приклади специфікації програмних систем з області телекомунікації. Довести коректність числення та його відносну повноту. [6]

Лекція 12 (2 год). Тема 12. Технологічні та інструментальні засоби специфікації та розробки програм за допомогою логік Флойда-Хоара.– 2 год.

Основні технологічні та інструментальні засоби специфікації та розробки програм за допомогою логік Флойда-Хоара. Система VRS. [6,10]

Завдання для самостійної роботи (4 год.)

Розглянути основні технологічні та інструментальні засоби специфікації та розробки програм за допомогою логік Флойда-Хоара. [6,10]

Лекція 13 (2 год). Тема 13. Верифікація систем в TLA. Загальні положення. (2 год)

Основні засади методу TLA. Досяжність і безпека. Структури даних. Визначення транзиційного переходу. [13]

Завдання для самостійної роботи (4 год.)

Розглянути основні засади методу TLA та побудувати приклади. Сформулювати властивості досяжності і безпеки. Надати визначення транзиційного переходу. [13]

Лекція 14 (2 год). Тема 14. Верифікація систем в TLA. Приклади. – 2 год.

Принципи верифікації в TLA. Приклади верифікації. [13]

Завдання для самостійної роботи (4 год.)

Розглянути принципи верифікації в TLA. Навести приклади верифікації. [13]

Лекція 15 (2 год). Тема 15. Перевірка моделей за допомогою SPIN.– 2 год.

Принципи перевірки моделей, заданих в TLA, за допомогою SPIN. [13]

Завдання для самостійної роботи (4 год.)

Розглянути принципи перевірки моделей, заданих в TLA, за допомогою SPIN. Навести приклади. [13]

Лекція 16 (2 год). Тема 16. Верифікація систем в Z.– 2 год.

Принципи перевірки моделей, заданих в Z. Приклади. Властивості Z-методу. [6,9,10]

Завдання для самостійної роботи (4 год.)

Розглянути принципи перевірки моделей, заданих в Z. Навести приклади. [6,9,10]

Лекція 17 (2 год). Тема 17. Верифікація систем в B.– 2 год.

Принципи перевірки моделей, заданих в B. Приклади. Властивості B-методу.[11]

Завдання для самостійної роботи (4 год.)

Розглянути принципи перевірки моделей, заданих в B. Навести приклади. [11]

Лекція 18 (2 год). Тема 18. Верифікація систем в RAISE.– 2 год.

Принципи перевірки моделей, заданих в RAISE. Приклади. Властивості RAISE-методу.[3]

Завдання для самостійної роботи (4 год.)

Розглянути принципи перевірки моделей, заданих в RAISE. Навести приклади. [3]

Лекція 18 (2 год). Тема 19. Програмні композиційно-номінативні логіки та їх застосування.– 2 год.

Визначення програмних композиційно-номінативних логік, їх класифікація, властивості та їх застосування.

Завдання для самостійної роботи (6 год.)

Розглянути принципи побудови композиційно-номінативних логік, провести моделювання традиційних логік, сформулювати властивості програм та їх коректності в композиційно-номінативних логіках. [2]

Консультація (1 год).

Типове завдання модульної роботи 2

Верифікувати з використання програмних засобів моделі наступних програмних систем:

1. Банкомат
2. Кавовий автомат
3. Склад
4. Е-магазин
5. Мікрохвильова піч
6. Телефон
7. Електроплита

Контрольні запитання до змістового модуля 2.

1. Навести приклади станів транзиційних систем.
2. Визначити логіки Флойда-Хоара. Сформулювати їх аксіоматику.
3. Довести коректність числення та відносну повноту логіки Флойда-Хоара.
4. Засоби специфікації та розробки програм за допомогою логік Флойда-Хоара.
5. Основні засади методу TLA та побудувати приклади.
6. Сформулювати властивості досяжності і безпеки.
7. Надати визначення транзиційного переходу.

8. Принципи верифікації в TLA.
9. Принципи перевірки моделей, заданих в TLA, за допомогою SPIN.
10. Принципи перевірки моделей, заданих в Z.
11. Властивості Z-методу.
12. Принципи перевірки моделей, заданих в B.
13. Властивості B-методу.
14. Принципи перевірки моделей, заданих в RAISE.
15. Властивості RAISE-методу.

Рекомендована література: [1–5, 7–13].

Питання до іспиту

1. Які принципи є основними методологічними принципами теорії програмування?
2. Як формулюються принципи розвитку та гносеологічності?
3. Як визначається пентада основних понять програмування?
4. Як визначаються поняття користувача, проблеми, програми, обчислюваності, програмування?
5. Які властивості основних понять програмування?
6. Властивості основної пентади програмування.
7. Властивості програмної пентади.
8. Що таке часткова коректність програм? Яким чином доводиться часткова коректність програм?
9. Що таке повна коректність програм? Яким чином доводиться повна коректність програм?
10. Розкрийте зміст формалізації поняття програми.
11. Визначте різні класи функцій.
12. Визначте програмні системи різного рівня абстракції.
13. Дайте визначення класу номінативних даних.
14. Повний клас обчислюваних функцій над номінативними даними.
15. Як аспекти программ є головними?
16. На підставі яких принципів відбувається формалізація програмних понять?
17. Класи функцій, що використовуються для формалізації програм.
18. Як визначаються інтенсіональні та екстенсіональні аспекти програмних понять?
19. Поняття композиційно-номінативної системи.
20. Як визначаються системи різних рівнів абстракції?
21. Як визначаються мови специфікацій та програмування?
22. Якими методами описують предметні області?
23. Як методи використовують для специфікації вимог до програмних систем?
24. Які засади RAISE-методу розробки програм?
25. Які засади B-методу розробки програм?
26. Які засади Z-методу розробки програм?
27. Які засади TLA-методу розробки програм?
28. Які логічні формалізми використовують для специфікацій програм?
29. Як використовується класична та некласична логіка для специфікацій програм?

30. Як визначаються темпоральні та модальні логіки?
31. Як визначаються аксіоматичні методи специфікації програм?
32. Як визначається логіка Флойда-Хоара та які властивості вона має?
33. Повнота логіки Флойда-Хоара.
34. Які особливості має семантико-синтаксична технологія розробки програм?
35. Які особливості має метод послідовних уточнень?
36. Які особливості у розробку програм вносять об’єктно-орієнтовані методи?
37. Яка мета стандартів програмування?
38. Як використовують технологічні та інструментальні засоби специфікації та розробки програм?

Рекомендована література

Основна

1. М.С. Нікітченко, Теорія програмування: Частина 1.– Ніжин: Видавництво НДУ імені Миколи Гоголя, 2010.– 119 с.
2. М.С. Нікітченко, С.С. Шкільняк. Математична логіка та теорія алгоритмів. – К., 2008.
3. И.А. Басараб, Н.С.Никитченко, В.Н. Редько. Композиционные базы данных. - К., Либідь, 1992.– 182 с.
4. The RAISE specification language. Prentice Hall Int.– 1992.– 397 p.
5. С. Лавров. Программирование. Математические основы, средства, теория.– СПб.: БХВ-Петербург, 2001.– 320 с.
6. Бабенко Л.П., Лавріщева К.М. Основи програмної інженерії: Навч. посіб.–К.: Т-во "Знання", 2001.– 269 с.

Додаткова

7. Hoare C.A.R., Jifeng He. Unifying Theories of Programming.– London: Prentice Hall Europe, 1998.– 298 p.
8. Schneider K.: Verification of Reactive Systems. Formal Methods and Algorithms. Springer-Verlag Berlin Heidelberg (2004)
9. Clarke E.M., Grumberg O., Peled D.: Model Checking. MIT Press (1999)
10. Mike Spivey. The Z Notation: A Reference Manual, 2nd edition. Prentice Hall International Series in Computer Science, 1992.
11. Jim Davies and Jim Woodcock. Using Z: Specification, Refinement and Proof. Prentice Hall International Series in Computer Science, 1996.
12. Jean-Raymond Abrial. Assigning Programs to Meanings, Cambridge University Press, 1996. ISBN 0-521-49619-5.
13. Steve Schneider. The B-Method: An Introduction, Cornerstones of Computing series, 2001. ISBN 0-333-79284-X.
14. Leslie Lamport. Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers, 2002 Pearson Education Publ.

Додаток 1.

Завдання
для самостійної роботи з елементами дистанційного навчання з дисципліни «Формальні методи розробки програмних систем» на період з 24 січня до 28 лютого 2018 р. для студентів 1 курсу магістратури освітньої програми «Інформатика»

Викладач: проф. Нікітченко М.С.
(електронна пошта nikitchenko@unicyb.kiev.ua**,** nikitchenko@knu.ua **)**

Види та форми контрольних заходів з перевірки самостійної роботи студентів

Контроль за виконанням самостійної роботи студентами здійснюється у двох формах:

- у січні-лютому за допомогою електронних засобів (електронною поштою),
- у березні – шляхом проведення письмової контрольної роботи.

Впродовж січня-лютого (24 січня – 20 лютого 2018 р.) студенти мають вивчити запропоновані питання визначених тем на базовому рівні. Для підтвердження виконання завдання студенти мають надіслати відповіді на 3 теоретичних питання та виконану лабораторну роботу із звітом не пізніше **15 лютого 2018 р**.

Завдання першого етапу, які мають бути виконані та надіслані на електронну пошту викладача (nikitchenko@knu.ua), подано нижче.

На контрольну роботу за підсумками самостійної роботи виносяться всі зазначені нижче теоретичні питання.

Теоретичні питання для самостійного опрацювання

1. Предметні області та методи їх опису
2. Мови програмування та мови специфікацій
3. Формальні моделі мов програмування та мов специфікацій
4. Структури даних та класи функцій у мовах програмування та мовах специфікацій
5. Класи композицій у мовах програмування та мовах специфікацій
6. Часткова та повна коректність програм
7. Методи розробки коректних програм
8. Метод Флойда-Хоара доведення властивостей програм

Примітка: за узгодженням з викладачем можливий вибір інших теоретичних питань.

Лабораторна робота

Обрати одну з нижченаведених систем та метод специфікації і побудувати специфікацію цієї системи за допомогою обраного методу.

Системи для специфікації:

1. Банкомат
2. Кавовий автомат
3. Склад
4. Е-магазин
5. Мікрохвильова піч
6. Телефон
7. Електроплита
8. Трамвай
9. Ліфт
10. Літак
11. Смарт-система освітлення будинку

Примітка: за узгодженням з викладачем можливий вибір іншої системи.

Методи специфікації програмних систем:

1. B
2. TLA
3. RAISE
4. ASM
5. Z

Примітка: за узгодженням з викладачем можливий вибір іншого методу специфікації систем.

Критерії оцінювання

Викладач оцінює виконані завдання в категоріях «**зараховано**» або «**не зараховано**». Щоб отримати оцінку «зараховано» потрібно набрати 4 і більше балів за відповіді на теоретичні питання та виконати лабораторну роботу.

Відповіді на теоретичні питання оцінюються наступним чином: відповіді немає — 0 балів; відповідь є, але не повна — 1 бал; відповідь повна — 2 бали. Якщо студент отримає оцінку «не зараховано», у нього є час до **20 лютого** переробити завдання та надіслати їх викладачу повторно.

Контрольна робота оцінюється максимум в **10 балів**. Вона включає в себе 5 тестових питань з проблематики, винесеної на самостійну роботу, та одне практичне завдання. Правильна відповідь на кожне тестове завдання оцінюється в 1 бал. За розв'язання задачі студент може отримати від 1 до 5 балів.

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

Рекомендована література

1. М.С. Нікітченко, Теорія програмування: Частина 1.– Ніжин: Видавництво НДУ імені Миколи Гоголя, 2010.– 119 с.
2. М.С. Нікітченко, С.С. Шкільняк. Математична логіка та теорія алгоритмів. – К., 2008.
3. The RAISE specification language. Prentice Hall Int.– 1992.– 397 p.
4. Hoare C.A.R., Jifeng He. Unifying Theories of Programming.– London: Prentice Hall Europe, 1998.– 298 p.

5. Jean-Raymond Abrial. Assigning Programs to Meanings, Cambridge University Press, 1996. ISBN 0-521-49619-5
6. Steve Schneider. The B-Method: An Introduction, Cornerstones of Computing series, 2001. ISBN 0-333-79284-X.
7. Leslie Lamport. Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers, 2002 Pearson Education Publ.
8. Schneider K.: Verification of Reactive Systems. Formal Methods and Algorithms. Springer-Verlag Berlin Heidelberg (2004)
9. Clarke E.M., Grumberg O., Peled D.: Model Checking. MIT Press (1999)
10. Mike Spivey. The Z Notation: A Reference Manual, 2nd edition. Prentice Hall International Series in Computer Science, 1992.
11. Jim Davies and Jim Woodcock. Using Z: Specification, Refinement and Proof. Prentice Hall International Series in Computer Science, 1996.