



КРОСПЛАТФОРМНЕ ПРОГРАМУВАННЯ

Робоча програма навчальної дисципліни (Силабус)

1. Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>3 рік підготовки, 6 семестр</i>
Обсяг дисципліни	<i>Лекції: 36 год., лабораторні роботи: 18 год., самостійна робота: 66 год.</i>
Семестровий контроль/ контрольні заходи	<i>Залік, модульна контрольна робота, календарний контроль</i>
Розклад занять	<i>Згідно розкладу на осінній семестр поточного навчального року (http://roz.kpi.ua/)</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: к.т.н., доцент, Цуркан В. В., v.v.tsurkan@gmail.com Лабораторні роботи: к.т.н., доцент, Цуркан В.В., v.v.tsurkan@gmail.com</i>
Розміщення курсу	<i>Google classroom: https://classroom.google.com/</i>

2. Програма навчальної дисципліни

3. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Вивчення дисципліни «Кросплатформне програмування» дозволяє сформувати у здобувачів освіти компетенції, необхідні для розв'язання практичних задач професійної діяльності, пов'язаної із розробленням кросплатформного програмного забезпечення.

Метою вивчення дисципліни «Кросплатформне програмування» є формування у здобувачів освіти здатностей самостійно проєктувати та розробляти кросплатформне програмне забезпечення.

Предметом дисципліни «Кросплатформне програмування» є методи, засоби моделювання і розроблення кросплатформного програмного забезпечення.

Вивчення дисципліни «Кросплатформне програмування» сприяє формуванню у здобувачів освіти **фахових компетентностей (ФК)**, необхідних для розв'язання практичних задач професійної діяльності, пов'язаної з розробленням, вдосконаленням та експлуатуванням програмного забезпечення:

ФК01 Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

ФК02 Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

ФК03 Здатність проектувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів.

ФК05 Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення.

ФК06 Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки (в тому числі кібербезпеки).

ФК11 Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розроблення програмного забезпечення.

ФК14 Здатність до алгоритмічного та логічного мислення.

ФК21 Здатність визначати, аналізувати та документувати вимоги до програмного забезпечення мультимедійних та інформаційно-пошукових систем.

Вивчення дисципліни «Кросплатформне програмування» сприяє формуванню у студентів наступних **програмних результатів навчання (ПРН)** за освітньою програмою:

ПРН03 Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПРН05 Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розроблення програмного забезпечення.

ПРН06 Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

ПРН09 Вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.

ПРН10 Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.

ПРН11 Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.

ПРН12 Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.

ПРН15 Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

ПРН17 Вміти застосовувати методи компонентної розробки програмного забезпечення.

ПРН21 Знати засоби, аналізувати, вибирати, кваліфіковано застосовувати засоби забезпечення інформаційної безпеки (в тому числі кібербезпеки) і цілісності даних відповідно до розв'язуваних прикладних завдань та створюваних програмних систем.

ПРН23 Вміти документувати та презентувати результати розроблення програмного забезпечення.

4. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Успішному вивченню дисципліни «Кросплатформне програмування» передують вивчення дисциплін «Основи програмування», «Алгоритми та структури даних», «Основи комп'ютерних систем і мереж», «Компоненти програмної інженерії», «Бази даних» та «Програмування» навчального плану підготовки бакалаврів за спеціальністю 121 Інженерія програмного забезпечення.

Отримані при засвоєнні дисципліни «Кросплатформне програмування» теоретичні знання та практичні уміння забезпечують успішне виконання курсових проєктів та магістерських дисертацій за спеціальністю 121 Інженерія програмного забезпечення.

5. Зміст навчальної дисципліни

Дисципліна «Кросплатформне програмування» передбачає вивчення таких тем:

Тема 1. Специфікування вимог до кросплатформного програмного забезпечення.

Тема 2. Створення логічної структури кросплатформного програмного забезпечення.

Тема 3. Створення фізичної структури кросплатформного програмного забезпечення.

Тема 4. Реалізування фізичної структури кросплатформного програмного забезпечення.

Модульна контрольна робота

Залік

6. Навчальні матеріали та ресурси

Базова література:

1. OMG® Unified Modeling Language® (OMG UML®). Version 2.5.1. URL: <https://www.omg.org/spec/UML/2.5.1/PDF> (accessed on: 01.06.2022).
2. OMG Systems Modeling Language (OMG SysML™). Version 1.6. URL: <https://sysml.org/.res/docs/specs/OMGSysML-v1.6-19-11-01.pdf> (accessed on: 01.06.2022).
3. Percival H., Gregory B. *Architecture Patterns with Python: Enabling Test-Driven Development, Domain-Driven Design, and Event-Driven Microservices*. Sebastopol, USA : O'Reilly Media, 2020. 304 p.
4. Aniche M. *Effective Software Testing: A developer's guide*. Shelter Island, USA: Manning, 2022. 328 p.
5. *Threat Modeling. Process*. URL: https://owasp.org/www-community/Threat_Modeling_Process (accessed on: 01.06.2022).
6. Shostack A. *Threat Modeling: Designing for Security*. Indianapolis: John Wiley & Sons, 2014. 590 p.
7. ISO/IEC 27005:2018. *Information technology. Security techniques. Information security risk management*. [Valid from 2018-06-10]. URL: <https://www.iso.org/standard/75281.html> (accessed on: 01.06.2022).

Додаткова література:

8. Dennis A., Wixom B., Tegarden D. *Systems Analysis and Design : An Object-Oriented Approach with UML*. Hoboken, USA : Wiley, 2020. 544 p.
9. Holt J, Perry S. *SysML for Systems Engineering: A model-based approach (Computing and Networks)*. London, United Kingdom : The Institution of Engineering and Technology, 2019. 880 p.
10. *Python documentation*. URL: <https://docs.python.org/3/> (accessed on: 01.06.2022).
11. Myers G., Sandler C., Badgett T. *The Art of Software Testing*. Hoboken, USA : Wiley, 2011. 256 p.
12. *Threat Modeling*. URL: <https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling> (accessed on: 01.06.2022).
13. Tarandach I., Coles M. J. *Threat Modeling. A Practical Guide for Development Teams*. Sebastopol: O'Reilly Media, 2020, 201 p.
14. MITER ATT&CK. URL: <https://attack.mitre.org/> (accessed on: 01.06.2022).

Використати для опанування практичних умінь дисципліни. Матеріали знаходяться у вільному доступі в Інтернеті.

7. Навчальний контент

8. Методика опанування навчальної дисципліни (освітнього компонента)

№ з/п	Тип навчального заняття	Опис навчального заняття
<i>Тема 1. Специфікування вимог до кросплатформного програмного забезпечення</i>		
1	<i>Лекція 1. Зміст курсу, вступ до кросплатформного програмування</i>	<i>Огляд змісту курсу. Поняття кросплатформності як властивості програмного забезпечення. Різновиди кросплатформності програмного забезпечення. Життєвий цикл розроблення кросплатформного програмного забезпечення. Способи та засоби досягнення кросплатформності програмного забезпечення. Завдання на СРС: п. 6 № 1.</i>
2	<i>Лекція 2. Методи визначення функційних вимог до кросплатформного програмного забезпечення</i>	<i>Поняття функційних вимог до кросплатформного програмного забезпечення. Процес визначення функційних вимог до кросплатформного програмного забезпечення. Етапи визначення функційних вимог до кросплатформного програмного забезпечення. Представлення варіантів використання кросплатформного програмного забезпечення. Завдання на СРС: п. 6 № 2.</i>
3	<i>Лабораторна робота 1. Специфікування функційних вимог до кросплатформного програмного забезпечення</i>	<i>Завдання: специфікувати функційні вимоги до кросплатформного програмного забезпечення. Завдання на СРС: п. 6 № 3.</i>
4	<i>Лекція 3. Методи визначення нефункційних вимог до кросплатформного програмного забезпечення</i>	<i>Забезпечення безпеки як нефункційна вимога до кросплатформного програмного забезпечення. Процес визначення нефункційних вимог до кросплатформного програмного забезпечення. Етапи визначення нефункційних вимог до кросплатформного програмного забезпечення. Завдання на СРС: п. 6 № 4.</i>
5	<i>Лекція 4. Методи моделювання загроз безпеці кросплатформного програмного забезпечення</i>	<i>Поняття моделі загроз безпеці кросплатформного програмного забезпечення. Процес моделювання загроз безпеці кросплатформного програмного забезпечення. Етапи моделювання загроз безпеці кросплатформного програмного забезпечення. Характеристика комбінованого методу STRIDE-DREAD. Характеристика методу оцінювання ризиків інформаційної безпеки та кібербезпеки. Завдання на СРС: п. 6 № 5.</i>
6	<i>Лабораторна робота 2. Специфікування нефункційних вимог до кросплатформного програмного забезпечення</i>	<i>Завдання: специфікувати нефункційні вимоги (безпеки) до кросплатформного програмного забезпечення. Завдання на СРС: п. 6 № 6.</i>

Тема 2. Створення логічної структури кросплатформного програмного забезпечення

7	<i>Лекція 5. Статична логічна структура кросплатформного програмного забезпечення</i>	<i>Поняття статичної логічної структури кросплатформного програмного забезпечення. Способи представлення статичної логічної структури кросплатформного програмного забезпечення. Представлення сутностей предметної області класами. Завдання на СРС: п. 6 № 7.</i>
8	<i>Лекція 6. Представлення статичної логічної структури кросплатформного програмного забезпечення</i>	<i>Визначення класів, їх атрибутів і методів (операцій). Різновиди відношень між класами. Визначення відношень між класами. Графічні нотації представлення класів і відношень між ними. Приклади представлення класів і відношень між ними. Завдання на СРС: п. 6 № 8.</i>
9	<i>Лабораторна робота 3. Створення статичної логічної структури кросплатформного програмного забезпечення</i>	<i>Завдання: створити статичну логічну структуру кросплатформного програмного забезпечення. Завдання на СРС: п. 6 № 9.</i>
10	<i>Лекція 7. Динамічна логічна структура кросплатформного програмного забезпечення</i>	<i>Поняття динамічної логічної структури кросплатформного програмного забезпечення. Способи представлення динамічної логічної структури кросплатформного програмного забезпечення. Специфікування поведінки кросплатформного програмного забезпечення. Завдання на СРС: п. 6 № 10.</i>
11	<i>Лекція 8. Представлення динамічної логічної структури кросплатформного програмного забезпечення</i>	<i>Представлення поведінки кросплатформного програмного забезпечення діяльністю. Визначення вузлів і дуг діяльності. Визначення вузлів управління діяльністю. Графічні нотації представлення діяльності. Приклади представлення діяльності. Завдання на СРС: п.6 № 11.</i>
12	<i>Лабораторна робота 4. Створення динамічної логічної структури кросплатформного програмного забезпечення</i>	<i>Завдання: створити динамічну логічну структуру кросплатформного програмного забезпечення. Завдання на СРС: п.6 № 12.</i>
13	<i>Лекція 9. Взаємодія елементів кросплатформного програмного забезпечення</i>	<i>Поняття взаємодії елементів кросплатформного програмного забезпечення. Способи представлення взаємодії елементів кросплатформного програмного забезпечення. Специфікування взаємодії елементів кросплатформного програмного забезпечення. Завдання на СРС: п. 6 № 13.</i>
14	<i>Лекція 10. Представлення взаємодії елементів</i>	<i>Представлення взаємодії елементів кросплатформного програмного забезпечення лініями життя і повідомленнями між ними.</i>

	<i>кросплатформного програмного забезпечення</i>	<i>Визначення ліній життя. Визначення повідомлень між лініями життя. Графічні нотації представлення ліній життя і повідомлень між ними Приклади представлення ліній життя і повідомлень між ними. Завдання на СРС: п.6 № 14.</i>
<i>Тема 3. Створення фізичної структури кросплатформного програмного забезпечення</i>		
15	<i>Лекція 11. Фізична структура кросплатформного програмного забезпечення</i>	<i>Поняття фізичної структури кросплатформного програмного забезпечення. Способи представлення фізичної структури кросплатформного програмного забезпечення. Специфікування фізичної структури кросплатформного програмного забезпечення. Завдання на СРС: п. 6 № 15.</i>
16	<i>Лекція 12. Представлення фізичної структури кросплатформного програмного забезпечення</i>	<i>Представлення фізичної структури кросплатформного програмного забезпечення компонентами. Визначення портів і з'єднувачів компонентів. Визначення відношень між компонентами. Графічні нотації представлення фізичної структури. Приклади представлення фізичної структури. Завдання на СРС: п. 6 № 16.</i>
17	<i>Лабораторна робота 5. Створення фізичної структури кросплатформного програмного забезпечення</i>	<i>Завдання: створити фізичну структуру кросплатформного програмного забезпечення. Завдання на СРС: п.6 № 17.</i>
18	<i>Лекція 13. Фізична конфігурація кросплатформного програмного забезпечення</i>	<i>Поняття фізичної конфігурації кросплатформного програмного забезпечення. Способи представлення фізичної конфігурації кросплатформного програмного забезпечення. Специфікування фізичної конфігурації кросплатформного програмного забезпечення. Завдання на СРС: п. 6 № 18.</i>
19	<i>Лекція 14. Представлення фізичної конфігурації кросплатформного програмного забезпечення</i>	<i>Представлення фізичної конфігурації кросплатформного програмного забезпечення вузлами. Різновиди фізичних вузлів. Стереотипи фізичних вузлів. Визначення фізичних вузлів і відношень між ними. Графічні нотації представлення фізичної конфігурації. Приклади представлення фізичної конфігурації. Завдання на СРС: п. 6 № 19.</i>
20	<i>Лабораторна робота 6. Створення фізичної конфігурації кросплатформного програмного забезпечення</i>	<i>Завдання: створити фізичну конфігурацію кросплатформного програмного забезпечення. Завдання на СРС: п.6 № 20.</i>

<i>Тема 4. Реалізування фізичної структури кросплатформного програмного забезпечення</i>		
21	<i>Лекція 15. Засоби реалізування фізичної структури кросплатформного програмного забезпечення</i>	<i>Функційне призначення кросплатформного програмного забезпечення. Описання логіки кросплатформного програмного забезпечення. Вибір засобів реалізування фізичної структури кросплатформного програмного забезпечення. Завдання на СРС: п. 6 № 21.</i>
22	<i>Лекція 16. Логіка кросплатформного програмного забезпечення</i>	<i>Структура кросплатформного програмного забезпечення Елементи кросплатформного програмного забезпечення. Функції елементів кросплатформного програмного забезпечення. Відношення між елементами кросплатформного програмного забезпечення. Логіка елементів кросплатформного програмного забезпечення. Завдання на СРС: п. 6 № 22.</i>
23	<i>Модульна контрольна робота. Створення робочого проєкту кросплатформної програми</i>	<i>Завдання: створити робочий проєкт кросплатформного програмного забезпечення. Завдання на СРС: п.6 № 23.</i>
24	<i>Лекція 17. Тестування кросплатформного програмного забезпечення</i>	<i>Поняття тестування кросплатформного програмного забезпечення. Життєвий цикл кросплатформного програмного забезпечення. Методи тестування кросплатформного програмного забезпечення. Особливості тестування кросплатформного програмного забезпечення. Завдання на СРС: п. 6 № 24.</i>
25	<i>Лекція 18. Тест-проєктування кросплатформного програмного забезпечення</i>	<i>Поняття тест-проєктування кросплатформного програмного забезпечення. Поняття тест-кейсу. Атрибути тест-кейсів. Життєвий цикл тест-кейсів. Створення тест-кейсів. Якість тест-кейсів. Завдання на СРС: п. 6 № 25.</i>
26	<i>Лабораторна робота 7. Демонстрування робочого проєкту кросплатформного програмного забезпечення</i>	<i>Завдання: продемонструвати робочий проєкт кросплатформного програмного забезпечення. Завдання на СРС: п. 6 № 26.</i>
27	<i>Залік</i>	<i>Завдання на СРС: п. 6 № 27.</i>

9. Самостійна робота студента/аспіранта

Дисципліна «Кросплатформне програмування» ґрунтується на самостійних підготовках до аудиторних занять на теоретичні та практичні теми.

<i>№ з/п</i>	<i>Назва теми, що виноситься на самостійне опрацювання</i>	<i>Кількість годин</i>	<i>Література</i>
1	<i>Підготовка до лекції 1</i>	1,5	1–3; 8; 9

2	Підготовка до лекції 2	1,5	1; 2; 8; 9
3	Підготовка до лабораторної роботи 1	4	1; 2; 8; 9
4	Підготовка до лекції 3	1,5	1; 2; 5–9; 12–14
5	Підготовка до лекції 4	1,5	1; 2; 5–9; 12–14
6	Підготовка до лабораторної роботи 2	4	1; 2; 5–9; 12–14
7	Підготовка до лекції 5	1,5	1–3; 8; 9
8	Підготовка до лекції 6	1,5	1–3; 8; 9
9	Підготовка до лабораторної роботи 3	4	1–3; 8; 9
10	Підготовка до лекції 7	1,5	1–3; 8; 9
11	Підготовка до лекції 8	1,5	1–3; 8; 9
12	Підготовка до лабораторної роботи 4	4	1–3; 8; 9
13	Підготовка до лекції 9	1,5	1–3; 8; 9
14	Підготовка до лекції 10	1,5	1–3; 8; 9
15	Підготовка до лекції 11	1,5	1–3; 8; 9
16	Підготовка до лекції 12	1,5	1–3; 8; 9
17	Підготовка до лабораторної роботи 5	4	1–3; 8; 9
18	Підготовка до лекції 13	1,5	1–3; 8; 9
19	Підготовка до лекції 14	1,5	1–3; 8; 9
20	Підготовка до лабораторної роботи 6	4	1–3; 8; 9
21	Підготовка до лекції 15	1,5	1–3; 8–10
22	Підготовка до лекції 16	1,5	1–3; 8–10
23	Підготовка до модульної контрольної роботи	6	1–3; 8–10
24	Підготовка до лекції 17	1,5	3, 4, 11
25	Підготовка до лекції 18	1,5	3, 4, 11.
26	Підготовка до лабораторної роботи 7	3	3, 4, 11
27	Підготовка до заліку	6	1-14

10. Політика та контроль

11. Політика навчальної дисципліни (освітнього компонента)

Відвідування лекційних занять є обов'язковим.

Відвідування занять лабораторних робіт може бути епізодичним та за потреби консультації/захисту лабораторних робіт.

Правила поведінки на заняттях: активність, повага до присутніх, відключення телефонів.

Дотримання політики академічної доброчесності.

Правила захисту лабораторних робіт: роботи повинні бути виконані відповідно до поставлених завдань та згідно з обраним студентом варіантом.

12. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Протягом семестру студенти виконують 7 лабораторних робіт. **Максимальна кількість балів за кожну лабораторну роботу: 12 балів.**

Бали нараховуються за якість виконання і захист лабораторних робіт: 0-12 балів.

Критерії оцінювання якості виконання і захисту:

11–12 балів – робота виконана якісно, в повному обсязі, відповіді повні, добре аргументовані;

9–10 балів – робота виконана якісно, в повному обсязі, але має недоліки, відповіді з незначними помилками;

7–8 балів – робота виконана достатньо якісно, в повному обсязі, але містить суттєві недоліки, відповіді зі суттєвими помилками;

0 балів – робота виконана не якісно, не в повному обсязі, відповідей або немає, або невірні.

Максимальна кількість балів за виконання та захист лабораторних робіт:

12 балів × 7 лабораторних робіт = 84 бали.

Завданням **модульної контрольної роботи** є створення робочого проекту кросплатформного програмного забезпечення. Відповідь оцінюється 16 балами.

Критерії оцінювання модульної контрольної роботи:

15–16 балів – відповідь вірна, повна, добре аргументована;

12–14 балів – в цілому відповідь вірна, але має недоліки;

10–11 балів – у відповіді є суттєві помилки;

0 балів – немає відповіді або відповідь невірна.

Максимальна кількість балів за модульну контрольну роботу:

16 балів × 1 завдання = 16 балів.

Рейтингова шкала з дисципліни дорівнює:

$R = R_C = R_{\text{лабор. робіт}} + R_{\text{МКР}} = 84 \text{ бали} + 16 \text{ балів} = 100 \text{ балів.}$

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

На першій атестації (8-й тиждень) студент отримує «Зараховано», якщо його поточний рейтинг не менше 20 балів (50 % від максимальної кількості балів, яку може отримати студент до першої атестації).

На другій атестації (14-й тиждень) студент отримує «Зараховано», якщо його поточний рейтинг не менше 40 балів (50 % від максимальної кількості балів, яку може отримати студент до другої атестації).

Семестровий контроль: залік

Умови допуску до семестрового контролю:

При семестровому рейтингу (R_C) не менше 60 балів студент отримує залік «автоматом» відповідно до таблиці (Таблиця відповідності рейтингових балів оцінкам за університетською шкалою). В іншому випадку ним виконується залікова контрольна робота.

Якщо студент не погоджується з оцінкою «автоматом», то може спробувати підвищити свою оцінку шляхом написання залікової контрольної роботи. При цьому його бали, отримані за семестр, зберігаються, а з двох отриманих студентом оцінок виставляється краща («м'яка» система оцінювання).

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно

64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

13. Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, які виносяться на семестровий контроль, наведено у Додатку 1.

Робочу програму навчальної дисципліни (силабус):

Складено к.т.н., доцент Цуркан В.В.

Ухвалено кафедрою ПЗКС (протокол №8 від 25.01.2023)

Погоджено Методичною комісією факультету прикладної математики (протокол №6 від 27.01.2023)

Додаток 1. Перелік питань, які виносяться на семестровий контроль

- 1. Охарактеризувати поняття кросплатформності як властивості програмного забезпечення.*
- 2. Охарактеризувати різновиди кросплатформності програмного забезпечення.*
- 3. Охарактеризувати способи досягнення кросплатформності програмного забезпечення.*
- 4. Охарактеризувати засоби досягнення кросплатформності програмного забезпечення.*
- 5. Охарактеризувати методи визначення функційних вимог до кросплатформного програмного забезпечення.*
- 6. Охарактеризувати представлення варіантів використання кросплатформного програмного забезпечення.*
- 7. Охарактеризувати методи визначення нефункційних вимог до кросплатформного програмного забезпечення.*
- 8. Охарактеризувати методи моделювання загроз безпеці кросплатформного програмного забезпечення.*
- 9. Охарактеризувати поняття статичної логічної структури кросплатформного програмного забезпечення.*
- 10. Охарактеризувати представлення статичної логічної структури кросплатформного програмного забезпечення.*
- 11. Охарактеризувати поняття динамічної логічної структури кросплатформного програмного забезпечення.*
- 12. Охарактеризувати представлення динамічної логічної структури кросплатформного програмного забезпечення.*
- 13. Охарактеризувати поняття взаємодії елементів кросплатформного програмного забезпечення.*
- 14. Охарактеризувати представлення взаємодії елементів кросплатформного програмного забезпечення.*
- 15. Охарактеризувати поняття фізичної структури кросплатформного програмного забезпечення.*
- 16. Охарактеризувати представлення фізичної структури кросплатформного програмного забезпечення.*
- 17. Охарактеризувати поняття фізичної конфігурації кросплатформного програмного забезпечення.*
- 18. Охарактеризувати представлення фізичної конфігурації кросплатформного програмного забезпечення.*
- 19. Охарактеризувати реалізування фізичної структури кросплатформного програмного забезпечення.*
- 20. Охарактеризувати процес тестування кросплатформного програмного забезпечення.*