

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Електронне видання

**СУЧАСНІ ТЕХНОЛОГІЇ АВТОМАТИЗОВАНОГО
ПРОЕКТУВАННЯ ТА ВЕРИФІКАЦІЇ ПРОГРАМ
ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Методичні вказівки до складання іспиту з дисципліни для студентів
Спеціальності 121 «Інженерія програмного забезпечення» Спеціалізації
«Інженерія програмного забезпечення комп'ютерних систем»
кафедри інформаційних систем та технологій всіх форм навчання

Рекомендовано

Вченою радою факультету

*інформатики та обчислювальної
техніки НТУУ «КПІ»*

Протокол №1 від 30.08.2021 р.

Київ НТУУ

«КПІ»

2021

Сучасні технології автоматизованого проектування та верифікації програм. Тестування програмного забезпечення. Методичні вказівки до складання іспиту з дисципліни. [Електронне видання] / Уклад.: Я.Ю. Дорогий, О.О. Дорога-Іванюк. – К.: НТУУ «КПІ», 2021. – 21 с.

Методичні вказівки призначені для студентів спеціальності 121 «Інженерія програмного забезпечення» кафедри інформаційних систем та технологій всіх форм навчання. В посібнику наведена методичні рекомендації щодо складання іспиту з дисципліни.

Укладачі	Я.Ю. Дорогий, к.т.н., доц. О.О. Дорога-Іванюк, аспірант
Відповідальний редактор	В.П. Полторак, к.т.н., доц.
Рецензент	М.М. Ткач, к.т.н., доцент кафедри технічної кібернетики

За редакцією укладачів

ЗМІСТ

ВСТУП.....	4
1 Мета та завдання дисципліни.....	5
2 Цілі й функції іспиту, допуск до іспиту.....	10
2.1 Цілі та функції іспиту	10
2.2 Допуск до екзамену.....	10
2.3 «Автомат» з іспиту.....	11
3 Загальні рекомендації та вимоги складання іспиту	12
3.1 Загальні рекомендації до складання іспиту.....	12
3.2 Рекомендації до самостійної роботи.....	12
3.3 Вимоги до складання іспиту	13
3.4 Структура екзаменаційного білету.....	13
3.5 Оцінювання результатів складання іспиту.....	13
СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ	14
ДОДАТОК 1 ПЕРЕЛІК ПИТАНЬ ДО ІСПИТУ	16
ПРИМІТКИ	21

ВСТУП

Методичні вказівки до складання іспиту мають на меті забезпечення студентів методикою підготовки та успішного складання іспиту з дисципліни «Сучасні технології автоматизованого проектування та верифікації програм» студентами всіх форм навчання.

Методичні вказівки містять практичний матеріал щодо структури, вимог, методів підготовки до іспиту, перелік питань, що виноситься на іспит, а також рекомендована література.

1 Мета та завдання дисципліни

Навчальна дисципліна «Сучасні технології автоматизації проектування та верифікації програм» (СТАПВП) відноситься до циклу загальної підготовки для здобуття глибинних знань зі спеціальності навчальної програми третього рівня вищої освіти підготовки за спеціальністю 121 «Інженерія програмного забезпечення».

Курс "Сучасні технології автоматизації проектування та верифікації програм" належить до циклу дисциплін «Програмування». Цей курс базується на таких забезпечуючих дисциплінах:

- «Основи програмування-1. Організація програм»;
- «Основи програмування-2. Програмування структурованих даних»;
- «Основи операційних систем»;
- «Об'єктно-орієнтоване програмування-1. Об'єктно-орієнтоване програмування»;
- «Сучасні методології і технології розробки програмного забезпечення-1. Концептуальне моделювання та автоматизація програмування».
- «Інтеграція застосувань на основі мікросервісного підходу».

Метою навчальної дисципліни є ознайомлення студентів з сучасними методами автоматизованого проектування та моделювання складних систем, методологіями об'єктно-орієнтованого аналізу та проектування, методами реалізації об'єктного підходу в мовах програмування високого рівня для проектування та моделювання складних систем, компонентами об'єктного підходу до аналізу та проектування складних систем, сучасними засобами підтримки об'єктно-орієнтованого підходу, а також засвоєння основних понять та визначень з галузі тестування, критеріїв вибору тестів, огляд різновидів тестування, аналіз особливостей процесу й технології індустріального тестування, придбання навичок застосування сучасних

інформаційних технологій для аналізу та тестуванню інформаційних систем, створення звітної тестової документації.

Предметом вивчення навчальної дисципліни є сучасні методи та технології автоматизованого проектування, тестування та керування якістю програмного забезпечення (ПЗ). Отриманні, в межах цього напрямку, теоретичні знання та практичні навички гармонічно поєднують в собі сучасні здобутки та погляди за окресленою проблематикою.

Дисципліна сприяє розумінню та засвоєнню студентами багатьох аспектів аналізу інформації в різноманітних системах управління, комп'ютерних та телекомунікаційних системах та мережах.

В результаті вивчення дисципліни студенти повинні

ЗНАТИ:

- архітектурні рішення і засоби проектування систем;
- вимоги до архітектури систем;
- стандарти моделювання;
- методи аналізу потреб;
- методи формальної специфікації програмних систем;
- модульне, інтеграційне, системне, регресійне тестування;
- засоби автоматизації тестування, тестового циклу, витрати тестування;
- планування тестування, підходи до розробки тестів, особливості ручної розробки й генерації тестів;
- документування тестування, огляди й метрики;
- аналіз методів відбору тестів, оцінка їх ефективності;
- принципи побудови сучасних програмних продуктів (ПП);

ВМІТИ:

- вибирати стратегії для планування життєвого циклу системи;
- розробляти формальні моделі ПС;
- розробляти шаблони і моделі;

- застосовувати моделі оцінки і параметри для аналізу вартості стосовно до фаз життєвого циклу програмного забезпечення;
- розробляти функціональні специфікації з урахуванням встановлених вимог.
- визначати організаційну, економічну, технічну та операційну здійсненність проекту;
- проектувати компоненти програмного забезпечення;
- розробляти проектну та робочу документації системи;
- використовувати стандарти якості при створенні ПП;
- розробляти технології забезпечення якості програмного забезпечення;
- обирати критерії вибору тестів;
- використовувати модульне, інтеграційне, системне, регресійне тестування;
- здійснювати планування тестування, розробку тестових сценаріїв та процедур, аналізувати особливості розробки й генерації тестів;
- здійснювати документування тестування, використовувати огляди й метрики;
- орієнтуватися в специфіці функціонування сучасних ПЗ;
- обґрунтовувати вимоги щодо основних характеристик перспективних ПЗ;
- виявляти потенційні вразливості ПЗ різного призначення;
- планувати стратегію організації комплексного тестування ПЗ.

Перелік компетенцій, сформованих навчальною дисципліною СТАПВП:

- **загально – професійні:**
 - здатність використовувати основні поняття, принципи та методи технологій автоматизованого проектування ПП;

- здатність використовувати основні поняття, принципи та методи технологій тестування ПП;
 - здатність використовувати стандарти якості при створенні ПП;
 - здатність розробляти технології забезпечення якості програмного забезпечення;
 - здатність обирати критерії вибору тестів;
 - здатність використовувати модульне, інтеграційне, системне, регресійне тестування;
 - здатність здійснювати планування тестування, розробку тестових сценаріїв та процедур, аналізувати особливості розробки й генерації тестів, здійснювати автоматизацію тестового циклу;
 - здатність здійснювати документування тестування, використовувати огляди й метрики.
- **технологічна** компетенція відображає здатність розв'язувати типові задачі з використанням інформаційно-комунікаційних технологій;
- **дослідницька** компетенція включає володіння науковими методами математичного моделювання; здатність будувати математичні моделі задач у пізнанні навколишнього світу; готовність систематизувати отримувані результати, встановлювати зв'язки з попередніми результатами, узагальнювати результати;
- **методологічна** компетенція відображає здатність оцінювати доцільність та особливість використання методів математичного моделювання для розв'язування професійних задач;
- **мотиваційна компетенція** – це сфера та характер задоволення пізнавальних та емоційно-ціннісних потреб особистості, характер спрямованості діяльності особистості на зміст або процес учіння, спрямованість ставлення людини до суспільства, самої себе, характер потреб, ціннісних орієнтацій, світогляду;

— **предметна компетенція** – розуміння місця кожної науки у системі знань людства, діалектики, отримання нових теоретичних знань та їх використання на практиці, незалежне оперування предметними знаннями та їх критичне осмислення з позицій практики інших наук.

Вивчення дисципліни СТАПВП спрямоване на оволодіння студентами системою вмінь, які визначені освітньо-кваліфікаційною характеристикою доктора філософії спеціальності підготовки 121 “Інженерія програмного забезпечення” та забезпечують вище визначені компетенції.

2 Цілі й функції іспиту, допуск до іспиту

2.1 Цілі та функції іспиту

Іспит з дисципліни «Сучасні технології автоматизації проектування та верифікації програм» повинен визначати рівень засвоєння студентами матеріалів дисципліни, вміння самостійно аналізувати складні явища та процеси, будувати правильні програми та проводити їх тестування, застосовувати широкі можливості засобів автоматизованого проектування та верифікації програмних систем, активно використовувати набуті знання у своїй професійній діяльності при вирішенні різноманітних завдань.

Цілі іспиту зумовлюють і його функції. Головною з них є контроль та оцінка рівня знань, отриманих студентом протягом навчання у семестрі.

Реалізація цієї функції припускає перевірку знань з методологічних та теоретичних принципів, проблем і положень дисципліни, а також вміння їх використовувати в вирішенні конкретних завдань.

Важливе значення має функція виявлення навичок вирішення практичних завдань, конкретного аналізу проблемних ситуацій професійного спрямування.

Іспит виконує ще й важливу виховну функцію.

У конкретному вираженні вона виявляється у вмінні студента самостійно, логічно й послідовно висловлювати свої переконання, здійснювати самоконтроль та критично оцінювати власні знання і навички.

Таким чином, іспит в значній мірі, є засобом всебічного впливу на особистість студента і має контрольну, пізнавальну та виховну функції.

2.2 Допуск до екзамену

Рейтинг студента з дисципліни складається з балів, що він отримує за:

- 1) виконання та захист 7 лабораторних робіт;
- 2) виконання семестрової самостійної роботи;
- 3) відповіді на екзамені.

Для отримання допуску до екзамену студент денної форми повинен:

- захистити 7 лабораторних робіт;
- виконати та захистити семестрову самостійну роботу.

При цьому студент повинен набрати не менше 60 рейтингових балів.

2.3 «Автомат» з іспиту

Студенти, які набрали протягом семестру більше 60 рейтингових балів мають можливість отримати «автомат».

3 Загальні рекомендації та вимоги складання іспиту

3.1 Загальні рекомендації до складання іспиту

Згідно з учбовим планом, студенти всіх форм навчання складають іспит з дисципліни. Перед складанням іспиту студент має детально вивчити лекційний матеріал з курсу “Сучасні технології автоматизації проектування та верифікації програм”, підібрати відповідну навчальну, методичну та спеціальну літературу за рекомендованим списком (а також самостійно підбраною).

3.2 Рекомендації до самостійної роботи

У процесі підготовки до іспиту студенти повинні не просто керуватися навчальною програмою, а виробити власне розуміння опрацьованої теми чи окремого питання. Для цього їм необхідно детально спланувати свою роботу, розподілити теми на питання, окремі питання – на підпитання, що дасть можливість краще засвоїти матеріал.

При вивченні літературних джерел рекомендується скласти для себе так званий поширений план, який включає вузлові питання теми, підпитання окремих питань, виділяє найбільш істотну інформацію, цифровий та статистичний матеріал, посилання на джерела. Крім того, необхідно забезпечити єдність теми, взаємозв'язок її окремих питань (включаючи ті, що розглянуті в аудиторії з викладачем) та підпитань; логічність викладання інформації.

Таким чином добре спланована самостійна робота студента допоможе глибоко осмислити тему, успішно підготуватись до іспиту.

Необхідною умовою глибокого та міцного засвоєння матеріалу по предмету є конспект, у якому стисло викладений зміст предмета, занотовані основні думки у формі плану, тез, виписок, цитат тощо.

Для перевірки якості засвоєння теоретичного матеріалу, а також застосування набутих знань у майбутній професійній діяльності після

вивчення теоретичного матеріалу необхідно потренуватись у розв'язанні практичних завдань.

3.3 Вимоги до складання іспиту

При складанні іспиту необхідно дотримуватись таких вимог:

- теоретичні питання та умови задач наводяться перед відповіддю та відповідним рішенням;
- розв'язок задач має супроводжуватись поясненням, при необхідності, формулами тощо.
- відповіді на питання білету повинні бути охайно оформлені, розбірливо написані без скорочення слів (крім загальновідомих скорочень);
- лістинги програм повинні бути добре прокоментовані.

3.4 Структура екзаменаційного білету

Екзаменаційний білет складається з двох частин:

- 2-х теоретичних питань згідно переліку питань на іспит (Додаток 1).
Відповідь повинна бути викладена у трактовці виконавця. Треба виказати своє відношення до цієї проблеми. Виклад теоретичного питання можна супроводжувати малюнками, схемами, таблицями, графіками.
- 2-х практичних завдань, що визначаються викладачем.

3.5 Оцінювання результатів складання іспиту

Оцінювання результатів складання іспиту відбувається на базі РСО, що є невід'ємною частиною робочої навчальної програми кредитного модуля.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Лавріщева К.М. Програмна інженерія. – К. – 2008. – 319 с.
2. Automation Testing Using Cucumber Tool and Selenium [Електронний ресурс] – Режим доступа: <http://www.softwaretestinghelp.com/cucumber-bdd-tool-selenium-tutorial-30/>. (Дата звернення: 20.01.2020).
3. Test Automation Snake Oil [Електронний ресурс] – Режим доступа: http://www.satisfice.com/articles/test_automation_snake_oil.pdf. (Дата звернення: 20.01.2020).
4. Hellesпу, The Cucumber Book: Behaviour-Driven Development for Testers and Developers / M. Wynne, A. Hellesпу, The Pragmatic Bookshelf. - Jan. 2012. - 313 pp. - ISBN 978-1934356807.
5. Corrca de Oliveira J., Costa Gouveia C., Quidute Filho R. “A way of Improving Test Automation Cost-Effectiveness.”, C.E.S.A.R., 2006, pp.1-5.
6. Fewster M. “Common Mistakes in Test Automation.”, Grove Consultants, 2001, pp.1-7.
7. Marik B. “When Should a Test Be Automated?”, Testing Foundations, 1998, pp.1-20.
8. Success with Test Automation [Електронний ресурс] – Режим доступа: <https://www.prismnet.com/~wazmo/succrap.htm>. (Дата звернення: 20.01.2020).
9. Rossmiller M. “6 Tips to Getting Started with Automated Testing.”, SmartBesar Software, 2011, pp.1-7.
10. Vihari. S. “A System of Humanizing Test Automation Outlay Efficiency.” International Journal of Emerging Science and Engineering, May 2013, pp.74-78.
11. Винниченко, И. Автоматизация процессов тестирования / И. Винниченко. - СПб.: питерб 2005. - 203с. - ISBN 5-469-00798-7.

12. Котляров, В. П. Основы тестирования программного обеспечения: учебное пособие / В.П. Котляров, Т.В. Коликова. - М.: Интернет-Ун-т Информ. Технологий, 2006. -- 288с. - ISBN 5-94774-406-4.

13. Сучасні технології автоматизованого проектування та верифікації програм. Тестування програмного забезпечення. Методичні вказівки до виконання лабораторних робіт. [Електронне видання] / Уклад.: Я.Ю. Дорогий, О.О. Дорога-Іванюк. – К.: НТУУ «КПІ», 2021. – 87 с.

14. Методичні вказівки до організації самостійної роботи студентів (СРС) з дисципліни «Сучасні технології автоматизованого проектування та верифікації програм». Мова моделювання UML / уклад. Я.Ю. Дорогий, О.О. Дорога-Іванюк, – Київ.: НТУУ “КПІ ім. І. Сікорського”, 2021. – 60 с.

15. Сучасні технології автоматизованого проектування і верифікації програм: Конспект лекцій [Електронний ресурс] : навч. посіб. для студ. спеціальності 121 «Інженерія програмного забезпечення», спеціалізації «Інженерія програмного забезпечення комп’ютерних систем» / КПІ ім. Ігоря Сікорського; уклад.: Я. Ю. Дорогий, О. О. Дорога-Іванюк. – Електронні текстові дані (1 файл: 3,9 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 89 с.

ДОДАТОК 1 ПЕРЕЛІК ПИТАНЬ ДО ІСПИТУ

1. Назвіть основні існуючі методи проектування програмного забезпечення. Коротко охарактеризуйте кожний з них.
2. Назвіть основні нотації структурного аналізу і проектування (SADT). Коротко охарактеризуйте кожну з них.
3. Охарактеризуйте модельно-орієнтований підхід. Назвіть основні підходи, що сформувалися на базі модельно-орієнтованого підходу.
4. Назвіть галузі знань SWEBOOK інженерії розроблення програмного продукту.
5. Наведіть базові поняття SWEBOOK.
6. Які моделі створюються в рамках використання MDA. Коротко охарактеризуйте кожну з них.
7. Назвіть принципи, стандарти та технології перетворення моделей в рамках MDA.
8. Дайте загальну характеристику гнучких методологій розробки. Назвіть переваги та недоліки гнучких методологій розробки. Назвіть основні концептуальні засади екстремального програмування.
9. Коротко охарактеризуйте методологію SCRUM. Назвіть та поясніть основні поняття методології SCRUM.
10. Коротко охарактеризуйте методологію Kanban. Назвіть та поясніть основні поняття методології Kanban.
11. Що собою представляє життєвий цикл ПЗ? З яких основних етапів він складається? Який основний нормативний документ регламентує ЖЦ ПЗ? Які процеси він описує?
12. Який принцип організації каскадної моделі життєвого циклу ПЗ? Назвіть її основні фази, переваги, недоліки та випадки використання.
13. Який принцип організації V-подібної моделі життєвого циклу ПЗ? Назвіть її переваги, недоліки та випадки використання.
14. Опишіть ітеративну (інкрементальну) модель та модель швидкої розробки прикладних програм RAD?

15. Який принцип організації спіральної моделі життєвого циклу ПЗ? Назвіть її переваги, недоліки та випадки використання.
16. В чому полягає принцип розробки адаптованих моделей? Наведіть приклади. Якими є загальні вимоги до методології та технології розробки ПЗ?
17. Що таке програмні вимоги? Наведіть приклад структури вимог за рівнями.
18. Які вимоги належать до групи функціональних вимог? Які вимоги належать до групи нефункціональних вимог?
19. Перелічіть основи програмних вимог, дайте коротке пояснення кожному пункту. Що включає в себе процес роботи вимогами? Які основні типи ролей беруть участь в процесі роботи з вимогами?
20. Яким чином формуються вимоги до програмного продукту? Які етапи включає процес аналізу вимог?
21. Назвіть основні методи перевірки вимог, дайте характеристику кожному з них.
22. Якими властивостями мають володіти програмні вимоги? Наведіть приклади формалізації вимог.
23. Діаграма варіантів використання UML. Основні елементи, їх графічне позначення. Типи відношень, їх характеристика та графічне представлення.
24. Діаграма класів UML. Поняття «клас», його характеристика.
25. Діаграма класів UML. Базові відношення між класами в UML, їх графічне позначення. Різниця між агрегацією та композицією.
26. Діаграма об'єктів UML. Поняття «об'єкт». В яких випадках доцільно розробляти діаграму об'єктів?
27. Параметризований клас (шаблон). Характеристика.
28. Діаграма станів UML. Автомат в UML, обов'язкові умови його існування.

29. Діаграма станів UML. Поняття «стан», графічне позначення стану в UML.
30. Діаграма станів UML. Основні види станів, приклади їх графічного позначення.
31. Охороні умови. Паралельний перехід, види паралельного переходу. Реалізація переходу між складеними станами.
32. Діаграми діяльності UML. Стани дій, переходи та доріжки на діаграмах діяльності, графічне позначення.
33. Діаграми діяльності UML. Приклади (з графічним позначенням) основних видів станів в діаграмі діяльності.
34. Діаграма послідовності UML. Лінія життя об'єкту, графічне позначення.
35. Діаграма послідовності UML. Фокус управління, графічне позначення.
36. Повідомлення на діаграмі послідовності UML. Приклади основних повідомлень з графічним позначенням.
37. Комбіновані фрагменти діаграми послідовності UML, приклади.
38. Діаграма розгортання в UML. Основні елементи. Приклад.
39. Діаграма синхронізації в UML. Основні елементи діаграми синхронізації. Приклад.
40. Аспекти визначення якості та її атрибути.
41. Парадигма «вбудови» якості в інженерії програмного забезпечення. Матриця «дім якості»
42. Вимоги до якості програмного забезпечення.
43. Стандарти в інженерії якості. CMM / CMMI. Стандарти ISO. Six. SIGMA. ITIL.
44. Сертифікація програмного забезпечення в Україні.
45. Метрики якості.
46. Лос- оцінки.
47. Метрики стилістики й зрозумілості програм.

48. Об'єктно-орієнтовані метрики.
49. Метрики Холстеда.
50. Метрики циклічної складності за Мак-Кейбом.
51. Метрики Чепіна.
52. Попередня оцінка якості на основі статистичних методів в залежності від етапу розробки програми.
53. Попередня оцінка складності програми на етапі розробки специфікацій вимог до програми.
54. Попередня оцінка складності програми на етапі визначення архітектури.
55. Класифікація моделей якості.
56. Модель якості МакКола.
57. Модель якості Боема.
58. Модель якості FURPS.
59. Модель якості Друмі.
60. Модель якості ISO 9126.
61. Моделі CMM.
62. Універсальна модель якості.
63. Якість процесу в інженерії ПЗ.
64. Техніки управління якістю.
65. Формальні методи доведення програм .
66. Модель формального доведення конкретності програм.
67. Методи перегляду структури програм.
68. Процес верифікації.
69. Процес валідації.
70. Процеси колективного аналізу.
71. Інспекції.
72. Рецензування коду.
73. Аудити.
74. Поняття дефекту, збою та відмови.

75. Кількісні оцінки при тестуванні.
76. Ефективність тестування.
77. Види тестування.
78. Рівні тестування.
79. Тестування методами «білого ящика», «сірого ящика» та «чорного ящика».
80. Техніки тестування.
81. Процес тестування.
82. Документування тестування.
83. Аналіз результатів тестування.
84. Управління процесом тестування.
85. Створення групи з тестування.
86. Розробка тест-стратегії.
87. Документування тестів й робочого продукту.
88. Звіти про дефекти.
89. Журнал тестування.

ПРИМІТКИ