Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Кафедра автоматики та управління в технічних системах

# ТЕХНОЛОГІЇ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ І ВЕРИФІКАЦІЇ ПРОГРАМ

## Робоча програма навчальної дисципліни (Силабус)

### Реквізити навчальної дисципліни

| | |
|---|---|
| **Рівень вищої освіти** | ***Третій (PhD)*** |
| **Галузь знань** | *12 Інформаційні технології* |
| **Спеціальність** | *121 Інженерія програмного забезпечення* |
| **Освітня програма** | *Інженерія програмного забезпечення* |
| **Статус дисципліни** | *Нормативна* |
| **Форма навчання** | *Очна (денна, вечірня), заочна* |
| **Рік підготовки, семестр** | *2 курс, осінній семестр* |
| **Обсяг дисципліни** | *4 кредити ЄКТС (120 годин)* |
| **Семестровий контроль/ контрольні заходи** | *Екзамен, модульна контрольна робота, календарний контроль* |
| **Розклад занять** | *Згідно розкладу на осінній семестр поточного навчального року (rozklad.kpi.ua)* |
| **Мова викладання** | *Українська* |
| **Інформація про керівника курсу / викладачів** | Лектор: *к.т.н., доцент,* Дорогий Ярослав Юрійович, *y.dorogoy@kpi.ua, моб. 0970060025*<br>Комп'ютерний практикум: *к.т.н., доцент,* Дорогий Ярослав Юрійович, *y.dorogoy@kpi.ua, моб. 0970060025* |
| **Розміщення курсу** | Електронний кампус |

### Програма навчальної дисципліни

#### 1. Опис навчальної дисципліни, її мета, предмет вивчання та результати навчання

*Навчальна дисципліна «Технології автоматизованого проектування і верифікації програм» (ТАПВП) відноситься до нормативних дисциплін для здобуття глибинних знань зі спеціальності навчальної програми третього рівня вищої освіти за спеціальністю 121 Інженерія програмного забезпечення. Дисципліна сприяє розумінню та засвоєнню студентами багатьох аспектів аналізу інформації в різноманітних системах управління, комп'ютерних та телекомунікаційних системах та мережах.*

***Метою*** *навчальної дисципліни є формування у здобувачів вищої освіти таких компетентностей:*

- *Здатність розробляти якісне та надійне програмне забезпечення складних програмних комплексів та систем на основі новітніх технологій та стандартів розроблення програмного забезпечення.*
- *Здатність застосовувати формальні методи проєктування, розроблення та дослідження програмних систем та технологій у наукових дослідженнях.*
- *Здатність проводити експериментальні дослідження з оцінювання ефективності та безпечності програмного забезпечення.*
- *Здатність розробляти технічну документацію до наукових проєктів.*

- *Здатність виконувати оригінальні дослідження, досягати наукових результатів, які створюють нові знання в інженерії програмного забезпечення та дотичних до неї міждисциплінарних напрямах, і можуть бути опубліковані у провідних наукових виданнях з інформаційних технологій та суміжних галузей.*
- *Здатність розширювати межі знань використовуючи результати оригінальних досліджень.*
- *Здатність забезпечувати безперервний саморозвиток і самовдосконалення, відповідальність за розвиток інших у професійній галузі, дотримуючись педагогічної етики, правил академічної доброчесності у науково-педагогічній діяльності.*
- *Здатність використовувати адекватні методи ефективної взаємодії з представниками різних груп (соціальних, культурних і професійних).*
- *Здатність працювати в команді, формувати позитивні відносини з колегами, спілкуватися з широкою науковою спільнотою та громадськістю в сфері інженерії програмного забезпечення.*

***Предмет*** *навчальної дисципліни: 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.*

*В результаті вивчення навчальної дисципліни здобувачі вищої освіти набудуть таких програмних* ***результатів*** *навчання:*

- *Знати сучасні технології автоматизованого проєктування і верифікації програм.*
- *Уміти застосовувати, розробляти та удосконалювати методи автоматизованого проєктування програмного забезпечення.*
- *Уміти застосовувати, розробляти та удосконалювати методи верифікації програмного забезпечення.*
- *Знати принципи побудови сценарних моделей та верифікації сценаріїв аналізу інформації.*
- *Мати передові концептуальні та методологічні знання з інженерії програмного забезпечення і на межі предметних галузей, а також дослідницькі навички, достатні для проведення наукових і прикладних досліджень на рівні сучасних світових досягнень з відповідного напряму, отримання нових знань та/або здійснення інновацій.*
- *Розробляти та досліджувати концептуальні, математичні і комп'ютерні моделі процесів і систем, ефективно використовувати їх для отримання нових знань та/або створення інноваційних продуктів у інженерії програмного забезпечення та дотичних міждисциплінарних напрямах.*
- *Розуміти теоретичні засади, що лежать в основі методів досліджень інформаційних систем та програмного забезпечення, методології проведення досліджень та обчислювальних експериментів.*
- *Уміти формулювати та вирішувати задачі оптимізації, адаптації, прогнозування, керування та прийняття рішень щодо процесів, засобів та ресурсів розроблення, впровадження, супроводу та експлуатації програмного забезпечення.*

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

*Успішному вивченню дисципліни «Технології автоматизованого проектування і верифікації програм» передує вивчення дисципліни «Основи програмування», «Основи операційних систем», «Об'єктно-орієнтоване програмування», «Сучасні методології і технології розробки програмного забезпечення» бакалаврської та магістерської підготовки. Також необхідним є вміння користуватись комп'ютером на рівні адміністратора, вміння програмувати, базові*

*знання в області теорії множин, вміння будувати Use-Case моделі для подальшої можливості аналізувати та прогнозувати майбутній функціонал застосування який буде проектуватись та тестуватись.*

*Отримані в результаті засвоєння дисципліни «Технології автоматизованого проектування і верифікації програм» теоретичні знання та практичні уміння можуть бути корисні для проведення наукових досліджень за темою дисертації, а також для засвоєння дисципліни «Методи реінжинірінгу програмного забезпечення».*

## 3. Зміст навчальної дисципліни

*Дисципліна «Технології автоматизованого проектування і верифікації програм» передбачає вивчення таких тем:*

*Тема 1. Сучасні технології автоматизованого проектування і верифікації програм як інженерна дисципліна*

*Тема 2. Тестування програмного коду*

*Модульна контрольна робота*

*Екзамен*

## 4. Навчальні матеріали та ресурси

***Базова література:***


*1. 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», 2020. – 87 с.*

*2. 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", 2020. – 60 с. 3. 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, 2020. – 89 с.*

*3. Лавріщева К.М. Програмна інженерія. – К. – 2008. – 319 с.*


***Додаткова література:***


*4. 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).*

*5. Test Automation Snake Oil [Електронний ресурс] – Режим доступа: http://www.satisfice.com/articles/test_automation_snake_oil.pdf. (Дата звернення: 20.01.2020).*

*6. Helles, The Cucumber Book: Behaviour-Driven Development for Testers and Developers / M. Wynne, A. Helleswy, The Pragmatic Bookshelf. - Jan. 2012. - 313 pp. - ISBN 978-1934356807*

*7. Corrka 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*

*8. Fewster M. "Common Mistakes in Test Automation.", Grove Consultants, 2001, pp.1-7.*

*9. Marik B. "When Should a Test Be Automated?", Testing Foundations, 1998, pp.1-20.*

*10. Success with Test Automation [Електронний ресурс] – Режим доступа: https://www.prismnet.com/~wazmo/succpap.htm. (Дата звернення: 20.01.2020).*

*11. Rossmiller M. "6 Tips to Getting Started with Automated Testing.", SmartBesar Software, 2011, pp.1-7.*

*12. Vihari. S. "A System of Humanizing Test Automation Outlay Efficiency." International Journal of Emerging Science and Engineering, May 2013, pp.74-78.*

**Навчальний контент**

## 5. Методика опанування навчальної дисципліни (освітнього компонента)

| № з/п | Тип навчального заняття | Опис навчального заняття |
|---|---|---|
| *Тема 1. Сучасні технології автоматизованого проектування і верифікації програм як інженерна дисципліна* | | |
| *1* | *Лекція 1. Сучасні технології автоматизованого проектування та верифікації програм як інженерна дисципліна.* | *Загальна характеристика дисципліни. Головні завдання СТАПВП. Основні терміни та поняття.*<br>*Завдання на СРС: опрацювати матеріали електронного конспекту (Лекція 1).* |
| *2* | *Лекція 2. Характеристика областей знань з інженерії програмного забезпечення – SWEBOOK.* | *Інженерія вимог. Проектування програмного забезпечення. Конструювання програмного забезпечення.*<br>*Завдання на СРС: опрацювати матеріали електронного конспекту (Лекція 2).* |
| *3* | *Лекція 3. Характеристика областей знань з інженерії програмного забезпечення – SWEBOOK (продовження).* | *Тестування програмного забезпечення. Методи і інструментарій. Якість програмного забезпечення.*<br>*Завдання на СРС: опрацювати матеріали електронного конспекту (Лекція 3).* |
| *4* | *Лекція 4. Стандарт і моделі життєвого циклу.* | *Характеристика життєвого циклу стандарта ISO/IEC 12207. Формування прикладних моделей життєвого циклу. Типи моделей життєвого циклу ПЗ.*<br>*Завдання на СРС: опрацювати матеріали електронного конспекту (Лекція 4).* |
| *5* | *Лекція 5. Agile розробка програмних систем.* | *Модель керування програмними проектами SCRUM. Модель керування програмними проектами Kanban.*<br>*Завдання на СРС: опрацювати матеріали електронного конспекту (Лекція 5).* |
| *Тема 2. Тестування програмного коду* | | |
| *6* | *Лекція 6. Методи доведення, верифікації та тестування програм. Мови специфікації.* | *Тестування, верифікація і валідація – відмінності в поняттях. Класифікація проблем, що виникають при роботі програмних систем. Документація, що створюється на різних етапах життєвого циклу. Типи процесів тестування і верифікації і їх місце в різних моделях життєвого циклу. Методи доведення, верифікації та тестування програм. Мови специфікації. Мови специфікації програм і їхня класифікація.*<br>*Завдання на СРС: опрацювати матеріали електронного конспекту (Лекція 6).* |

| | | |
|---|---|---|
| *7* | *Лекція 7. Методи доведення, верифікації та тестування програм. Верифікація і валідація програм.* | *Методи доведення, верифікації та тестування програм. Базові методи доведення. Модель доведення програми за твердженнями. Верифікація і валідація програм. Валідація сценарію вимог. Верифікація об'єктних моделей. Верифікація композиції компонентів.*<br><br>*Завдання на СРС: опрацювати матеріали електронного конспекту (Лекція 7).* |
| *8* | *Лекція 8. Тестування програмних систем.* | *Завдання і цілі тестування програмного коду забезпечення повторюваності тестування при промисловій розробці програмного забезпечення. Методи тестування: чорний ящик; скляний (білий) ящик; тестування моделей; аналіз програмного коду (інспекції). Тестове оточення: драйвери і заглушки; тестові класи; генератори сигналів (подієво-керований код). Передумови для виконання тесту, налаштування тестового оточення, оптимізація послідовностей тестових прикладів. Залежність між тестовими прикладами, налаштування за умовчанням для тестових прикладів і їх груп. Статичні методи тестування. Динамічні методи тестування. Функціональне тестування.*<br><br>*Завдання на СРС: обробити матеріал електронного конспекту (Лекція 8).* |
| *9* | *Лекція 9. Модульне тестування, інтеграційне та системне тестування. Тестування призначеного для користувача інтерфейсу. Документація, що супроводжує процес верифікації і тестування.* | *Завдання і цілі модульного тестування. Поняття модуля і його меж. Тестування класів. Підходи до проектування тестового оточення. Організація модульного тестування.*<br>*Завдання та цілі інтеграційного тестування. Організація інтеграційного тестування: структурна класифікація методів інтеграційного тестування; тимчасова класифікація методів інтеграційного тестування. Планування інтеграційного тестування.*<br>*Завдання і цілі системного тестування. Види системного тестування. Системне тестування, приймально-здавальні та сертифікаційні випробування при розробці програмного забезпечення, що сертифікується.*<br>*Завдання і цілі тестування призначеного для користувача інтерфейсу. Функціональне тестування призначених для користувача інтерфейсів. Перевірка вимог до призначеного для користувача інтерфейсу. Повнота покриття призначеного для користувача інтерфейсу. Методи проведення тестування призначеного для користувача інтерфейсу, повторюваність тестування призначеного для користувача інтерфейсу.* |

| | | |
|---|---|---|
| | | *Технологічні процеси верифікації і ролі в проекті, документація, що створюється в ході життєвого циклу проекту, її призначення. Стратегія і плани верифікації. Тест-вимоги: Технологічні ланцюжки і ролі учасників проекту, що використовують тест-вимоги; Зв'язок тест-вимог з іншими типами проектної документації. Властивості тест-вимог. Тест-плани.*<br>*Завдання на СРС: обробити матеріал лекції самостійно.* |
| *Модульна контрольна робота* | | |

*Комп'ютерний практикум:*

| *№ з/п* | *Тип навчального заняття* | *Опис навчального заняття* |
|---|---|---|
| *1* | *Комп'ютерний практикум 1. Види тестування. Планування тестування.* | *Вивчити класифікацію видів тестування, розробити перевірки для різних видів тестування, навчитися планувати тестові активності в залежності від особливостей продукції, що поставляється на тестування функціональності.* |
| *2* | *Комп'ютерний практикум 2. Розробка вимог.* | *Виявити й описати призначені для користувача вимоги у вигляді варіантів використання (Use Cases).* |
| *3* | *Комп'ютерний практикум 3. Тестування вимог.* | *Вивчити критерії якості вимог, виконати тестування вимог до програмного забезпечення.* |
| *4* | *Комп'ютерний практикум 4. Тестування програмного забезпечення. Розробка тестів.* | *Розробити робочу тестову документацію для тестування web-застосунку.* |
| *5* | *Комп'ютерний практикум 5. Пошук і документування дефектів.* | *Протестувати web-застосунок і описати знайдені дефекти.* |
| *6* | *Комп'ютерний практикум 6. Документування результатів тестування.* | *Скласти підсумковий звіт про результати тестування web-застосунку.* |
| *7* | *Комп'ютерний практикум 7. Тестування usability.* | *Вивчити та реалізувати на практиці експертний і призначений для користувача підходи юзабіліті тестування.* |

## 6. Самостійна робота аспіранта

*Дисципліна Технології автоматизованого проектування і верифікації програм» ґрунтується на самостійній підготовці до аудиторних занять на теоретичні теми.*

| *№ з/п* | *Назва теми, що виноситься на самостійне опрацювання* | *Кількість годин* | *Література* |
|---|---|---|---|

| | | | |
|---|---|---|---|
| *1* | *Підготовка до лекції 1. Вивчення архітектури, візуальних інтерфейсів та інструментальних засобів CASE-системи Visual Paradigm.* | *6* | *1-3* |
| *2* | *Підготовка до лекції 2. Аналіз вимог та розробка UML-діаграм концептуального проектування програмної системи* | *6* | *1-3.* |
| *3* | *Підготовка до лекції 3. Розробка UML-діаграм логічного рівня проектування програмної системи: моделювання статичних аспектів.* | *6* | *1-3* |
| *4* | *Підготовка до лекції 4. Розробка UML-діаграм логічного рівня проектування програмної системи: моделювання динамічних аспектів.* | *6* | *1-3* |
| *5* | *Підготовка до лекції 5. Розробка UML-діаграм фізичного рівня проектування програмної системи.* | *6* | *1-3* |
| *6* | *Підготовка до лекції 6. Методи доведення, верифікації та тестування програм. Мови специфікації.* | *10* | *1-2, 4-12.* |
| *7* | *Підготовка до лекції 7. Тестування програмних систем.* | *20* | *1, 4-12.* |
| *8* | *Підготовка до лекції 8. Модульне тестування, інтеграційне та системне тестування. Тестування призначеного для користувача інтерфейсу. Документація, що супроводжує процес верифікації і тестування.* | *20* | *1, 4-12.* |
| *9* | *Підготовка до лекції 9. Підготовка до іспиту по всьому матеріалу модуля.* | *4* | *1-12* |

## Політика та контроль

### 7. Політика навчальної дисципліни (освітнього компонента)

- *відвідування лекційних та комп'ютерних практикумів є обов'язковою складовою вивчення матеріалу;*
- *на лекції викладач користується власним презентаційним матеріалом; використовує гуглдиск для викладання матеріалу поточної лекції, додаткових ресурсів, лабораторних робіт та інше; викладач відкриває доступ до певної директорії гугл-диска для скидання електронних лабораторних звітів та відповідей на МКР;*
- *на лекції заборонено відволікати викладача від викладання матеріалу, усі питання, уточнення та ін. студенти задають в кінці лекції у відведений для цього час;*
- *лабораторні роботи захищаються у два етапи – перший етап: студенти виконують завдання на допуск до захисту лабораторної роботи; другий етап – захист лабораторної роботи. Бали за лабораторну роботу враховуються лише за наявності електронного звіту;*
- *модульні контрольні роботи пишуться на лекційних заняттях без застосування допоміжних засобів (мобільні телефони, планшети та ін.); результат пересилається у файлі до відповідної директорії гугл-диску;*

*Заохочувальні бали нараховуються за:*

*- активну участь на лекціях; участь у факультетських та інститутських олімпіадах з навчальних дисциплін, участь у конкурсах робіт, підготовка оглядів наукових праць; презентацій по одній із тем СРС дисципліни тощо. Кількість заохочуваних балів не більше 10;*
*- модернізація лабораторних робіт. Кількість заохочуваних балів не більше 10;*
*- виконання завдань із удосконалення дидактичних матеріалів з дисципліни. Кількість заохочуваних балів не більше 10;*

*Штрафні бали нараховуються за:*
*- невчасну здачу лабораторної роботи. Кількість штрафних балів 1;*
*- відсутність на лабораторному занятті без поважної. Кількість штрафних балів 2.*

## 8. Види контролю та рейтингова система оцінювання результатів навчання (РСО)

*Протягом семестру аспіранти виконують 7 комп'ютерних практикумів. Максимальна кількість балів за кожний комп'ютерний практикум: 5 балів.*

*Бали нараховуються за:*
*- якість виконання комп'ютерного практикуму: 0-2 бали;*
*- відповідь під час захисту комп'ютерного практикуму: 0-2 бали;*
*- своєчасне представлення роботи до захисту: 0-1 бали.*

*Критерії оцінювання якості виконання:*
*2 бали – робота виконана якісно, в повному обсязі;*
*1 бал – робота виконана в повному обсязі, але містить незначні помилки;*
*0 балів – робота виконана не в повному обсязі, або містить суттєві помилки.*

*Критерії оцінювання відповіді:*
*2 бали – відповідь повна, добре аргументована;*
*1 бал – в цілому відповідь правильна, але має недоліки або незначні помилки;*
*0 балів – немає відповіді або відповідь неправильна.*

*Критерії оцінювання своєчасності представлення роботи до захисту:*
*2 бали – робота представлена до захисту не пізніше вказаного терміну;*
*0 балів – робота представлена до захисту пізніше вказаного терміну.*

*Максимальна кількість балів за виконання та захист комп'ютерних практикумів:*
*5 балів × 7 комп. практ. = 35 балів.*

*Протягом семестру на лекціях відбувається **опитування за темою поточного заняття**. Максимальна кількість балів за опитування, яку можна отримати протягом семестру: 2 бали.*

*Максимальна кількість балів за опитування за темою поточного заняття:*
*2 балів × 8 лекцій = 16 балів.*

*Завдання на **модульну контрольну роботу** оцінюється 29 балами.*

*Критерії оцінювання кожного запитання контрольної роботи:*
*14-20 балів – повна відповідь, правильні відповіді на всі питання, гарна презентація результатів;*
*10-13 балів – достатньо повна відповідь, правильні відповіді на більшість поставлених питань, робочий макет;*
*5-9 балів – неповна відповідь, неправильні відповіді на деякі питання, частково працюючий макет;*
*0-4 балів – незадовільна відповідь, неправильні відповідь на більшість поставлених питань, відсутність розробленого макету;*
*0 балів – немає відповіді або відповідь неправильна.*

*Рейтингова шкала з дисципліни дорівнює:*
$R_c = R_{ком.практ} + R_{опитув} + R_{МКР} = 35$ *балів + 16 бали + 29 балів = 80 балів.*

*$R_E$=20 балів можна отримати за іспит. Тоді розмір рейтингової шкали з дисципліни складає:*

*$R = R_C + R_E$= 100 балів*

*Календарний контроль: провадиться 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 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 50 % від максимальної кількості балів, яку може отримати студент до другої атестації.*

*Семестровий контроль: **екзамен***

*Умови допуску до екзамену:*
*- зарахування всіх комп'ютерних практикумів;*
*- здача усіх розділів лекційного курсу;*
*- виконання модульної контрольної роботи.*

*Склад та критерії оцінювання екзамену:*

*Завдання на **екзамен** складається з 4 запитань – 2 теоретичних та 2 практичних. Відповідь на кожне теоретичне запитання оцінюється 3 балами, а відповідь на практичне запитання оцінюється 7 балами.*

*Критерії оцінювання кожного теоретичного запитання екзамену:*
*3 балів – повна відповідь, правильні відповіді на всі питання;*
*2-3 балів – достатньо повна відповідь, правильні відповіді на більшість питань;*
*1-2 балів – у відповіді є незначні помилки;*
*0-1 бали – у відповіді є суттєві помилки;*
*0 балів – немає відповіді або відповідь неправильна.*

*Критерії оцінювання практичного запитання екзамену:*
*6-7 балів – відповідь правильна, розрахунки виконані у повному обсязі;*
*4-5 бали – відповідь правильна, але не дуже добре підкріплена розрахунками;*
*2-3 балів – в цілому відповідь правильна, але має недоліки;*
*0-1 балів – у відповіді є незначні помилки;*
*0 балів – немає відповіді або відповідь неправильна.*

*Максимальна кількість балів за екзамен:*
*3 балів × 2 теоретичних запитань + 7 балів × 2 практичних запитань = 20 балів.*

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 |

## 9. Додаткова інформація з дисципліни (освітнього компонента)

*- Перелік запитань, які виносяться на семестровий контроль, наведено у Додатку 1.*

*- Передбачена можливість закривати частину лабораторного та лекційного матеріалу шляхом здобування сертифікатам по online курсам (наприклад, COURSERA) відповідних розділів та тем дисципліни.*

**Робочу програму навчальної дисципліни (силабус):**

**Складено** к.т.н., доц., Дорогий Ярослав Юрійович

**Ухвалено** кафедрою АУТС (протокол № 1 від 27.08.2020 р.)

**Погоджено** Методичною комісією (протокол № 1 від 02.09.2020 р.)

***Додаток 1***

***Перелік теоретичних питань на екзамен***

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 SWEBOK інженерії розроблення програмного продукту.
5. Наведіть базові поняття SWEBOK.
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. Охоронні 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. Loc- 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 ящику», «сірого ящику» та «чорного ящику».
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. Журнал тестування.