

Національний університет водного господарства та
природокористування
Навчально-науковий інститут автоматики, кібернетики та
обчислювальної техніки

ЗАТВЕРДЖУЮ

Голова науково-методичної ради НУВГП
_____ Олег ЛАГОДНЮК

«___» _____ 20__

04-01-14S

СИЛАБУС

навчальної дисципліни

SYLLABUS

Практична підготовка з основ програмної інженерії	Practical training in the basics of software engineering
Шифр за ОП	<u>OK16</u> Code in Educational Program
Освітній рівень: бакалаврський (перший)	Educational level: Bachelor's (first)
Галузь знань: Інформаційні технології	Fields of knowledge: Information technologies
Спеціальність: Інженерія програмного забезпечення	Field of study: Software engineering
Спеціалізація: Інтернет речей	Specialization: Internet of things
Освітня програма: Інтернет речей	Educational Program: Internet of things

Силабус навчальної дисципліни **«Практична підготовка з основ програмної інженерії»** для здобувачів вищої освіти ступеня «бакалавр», які навчаються за освітньо-професійною програмою **«Інтернет речей», 121 Інженерія програмного забезпечення**. Рівне. НУВГП. 2020. 12 стор.

ОПП «Інтернет речей» на сайті університету:
http://ep3.nuwm.edu.ua/18444/1/opp_internet_rechey_2019.pdf

Розробник силабусу: *Демчук Олена Станіславівна, к.т.н., доцент кафедри комп'ютерних наук та прикладної математики*

Силабус схвалений на засіданні кафедри комп'ютерних наук та прикладної математики
Протокол № 2 від "6" жовтня 2020 року

В.о. завідувача кафедри: *Турбал Юрій Васильович, д.т.н., професор*
Керівник освітньої програми: *Жуковський Віктор Володимирович, к.т.н., доцент*

Схвалено науково-методичною радою з якості ННІ АКOT
Протокол № 3 від "29" грудня 2020 року

Голова науково-методичної ради з якості ННІ АКOT: *Мартинюк Петро Миколайович, д.т.н., професор*

СЗ №-249 в ЕДО від 25 січня 2021 року (70-75442569).

© Демчук Олена
Станіславівна,
2020
© НУВГП, 2020

ЗАГАЛЬНА ІНФОРМАЦІЯ*

Ступінь вищої освіти	бакалавр
Освітня програма	Інтернет речей
Спеціальність	121 Інженерія програмного забезпечення
Рік навчання, семестр	1 рік; 2 семестр
Кількість кредитів	6
Лекції:	
Лабораторні заняття:	60
Самостійна робота:	120
Курсова робота:	ні
Форма навчання	денна
Форма підсумкового контролю	залік
Мова викладання	українська

ІНФОРМАЦІЯ ПРО ВИКЛАДАЧА*

Лектор



Демчук Олена Станіславівна, к.т.н., доцент кафедри комп'ютерних наук та прикладної математики

Вікіситет

http://wiki.nuwm.edu.ua/index.php/Демчук_Олена_Станіславівна

ORCID

<http://orcid.org/0000-0002-8318-5009>

Як комунікувати

o.s.demchuk@nuwm.edu.ua

ПРО ДИСЦИПЛІНУ

Анотація навчальної дисципліни, в т.ч. мета та цілі

Програма нормативної навчальної дисципліни «Практична підготовка з основ програмної інженерії» складена відповідно до освітньо-професійної програми «Інтернет речей» підготовки бакалавра за спеціальністю «Інженерія програмного забезпечення».

Міждисциплінарні зв'язки: дисципліна є складовою частиною блоку фундаментальної підготовки та відноситься до навчальних дисциплін циклу фахової підготовки студентів.

Дисципліна «Практична підготовка з основ програмної інженерії» спрямована на вивчення основних етапів життєвого циклу програмного забезпечення: аналізу предметної області, формулювання вимог, проектування,

конструювання, верифікації і тестування, оцінювання якості та керування проектом тощо.

Мета: формування у майбутніх інженерів-програмістів сучасного рівня інформаційної та програмістської культури, оволодіння основними принципами програмної інженерії; набуття практичних навичок самостійної розробки професійного програмного забезпечення і використання сучасних інформаційних технологій для розв'язання практичних задач.

Завдання: сформувані у студентів теоретичні знання та виробити практичні уміння у сфері розробки програмного забезпечення на всіх етапах життєвого циклу; підготувати студентів до ефективного використання основ програмної інженерії при вивченні спеціальних дисциплін, підготувати студентів до використання отриманих знань і навиків при розв'язуванні практичних задач, а також при написанні кваліфікаційних та магістерських робіт.

У результаті вивчення навчальної дисципліни студент повинен

знати:

1. Основні типи інструментарію для розробки ПЗ.
2. Основи управління вимогами до розробки ПЗ.
3. Види архітектури ПЗ.
4. Принципи та моделі розробки ПЗ, методології програмування.
5. Основні принципи і методи тестування ПЗ.
6. Принципи оцінювання якості та керування програмним проектом.

вміти:

- а) проводити аналіз і моделювання предметної області;
- б) формулювати вимоги до програмного продукту та керувати ними;
- в) розробляти структуру програмного проекту;
- г) конструювати програмну систему;
- д) розробляти документацію до програмного проекту;
- е) тестувати програмну систему.

Посилання на розміщення навчальної дисципліни на навчальній платформі Moodle
Компетентності

<https://exam.nuwm.edu.ua/course/view.php?id=3128>

ЗК02. Здатність застосовувати знання у практичних ситуаціях.

ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

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.

ФК05. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.

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

ФК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

Програмні результати навчання

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

ПРН04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.

ПРН07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПРН14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.

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

Перелік соціальних, «м'яких» навичок (soft skills)

Відкритість, взаємодія з людьми, вміння працювати в команді, здатність до навчання, здатність логічно обґрунтовувати позицію, знаходити вихід з складних ситуацій, знаходити час на відпочинок, комунікаційні якості, навички міжособистісних відносин, навички усного спілкування, саморозвиток, творчі здібності, чесність.

Структура навчальної дисципліни

Лабораторні – 60 год., Самостійна робота – 120 год.

МОДУЛЬ 1 ЗМІСТОВИЙ МОДУЛЬ 1

Стандарт і моделі життєвого циклу програмного забезпечення. Визначення вимог до ПЗ. Методи об'єктного аналізу і моделювання. Проектування архітектури ПЗ.

Вступ.

Поняття інженерії програмного забезпечення. Предмет і зміст дисципліни. Основні визначення та базові складові програмної інженерії.

Тема 1. Стандарт і моделі життєвого циклу ПЗ.

Характеристика життєвого циклу стандарту ISO/IEC 12207. Основні процеси життєвого циклу ПЗ: придбання, постачання, розроблення, експлуатація, супроводження. Процеси

підтримки та організаційні процеси розробки ПС. Типи моделей життєвого циклу: каскадна, інкрементна, спіральна, еволюційна моделі.

Тема 2. Визначення вимог до програмних систем: загальні підходи.

Поняття вимог до ПЗ. Основні типи загальних вимог: вимоги до продукту, вимоги користувачів, вимоги до ПЗ (системні, функціональні, нефункціональні). Аналіз і збирання вимог. Інженерія вимог. Фіксація вимог: специфікація вимог до ПЗ, валідація та верифікація вимог. Трасування вимог.

Тема 3. Об'єктно-орієнтований підхід до визначення вимог до програмних систем.

Візуальний підхід. Сценарій як варіант подання моделі виконання системи. Актори. Графічна нотація UML для побудови моделі сценаріїв. Відношення між сценаріями. Текстовий підхід. Поняття прецеденту, екземпляру. Середовище Rational Rose для формування вимог за допомогою прецедентів.

Тема 4. Методи об'єктного аналізу і моделювання.

Основні поняття об'єктно-орієнтованих методів аналізу та зв'язки між ними. Предметна область. Простір проблем. Об'єктна модель ПрО та метод OOAS Шлеєра-Меллора її побудови. Інформаційна модель системи. Зв'язки між об'єктами та їх типи. Відношення між об'єктами. Модель станів. Дії і події. Діаграми переходу станів. Таблиці переходу станів. Схема взаємодії моделей поведінки об'єктів. Модель процесів. Потоки керування і даних. Діаграми дій потоків даних.

Тема 5. Проектування архітектури програмних систем. Загальні підходи.

Поняття архітектури системи. Стандартизований підхід до проектування архітектури ПС. Особливості об'єктного підходу. Рівні архітектур програмних систем. Розподілена архітектурна схема. Архітектурна схема клієнт–сервер. Багаторівнева архітектурна схема.

ЗМІСТОВИЙ МОДУЛЬ 2

Прикладні і теоретичні методи програмування

Тема 6. Прикладне (систематичне) програмування. Структурне проектування і програмування.

Прикладні і теоретичні методи програмування. Огляд методів систематичного програмування. Суть структурного підходу до розробки ПС. Структурне проектування. Метод функціонального моделювання SADT. Метод SSADM. Метод IDEF1. Інструментальна підтримка SSADM.

Тема 7. Об'єктно-орієнтоване програмування.

Поняття об'єкту, класу об'єктів. Операції над об'єктами. Об'єктно-орієнтована модель програмної системи. Життєвий цикл розробки моделі системи у середовищі ООП. Типи моделей системної архітектури: статична і динамічна моделі. Модель оточення та модель використання системи.

Тема 8. UML-метод моделювання.

Мова візуального моделювання систем UML. Діаграма класів. Моделювання поведінки системи: діаграми послідовності, співпраці, діяльності, станів, реалізації, компонентів, розміщення. Схема моделювання і проектування програмної системи в UML.

Тема 9. Компонентне програмування.

Суть компонентного програмування. Поняття компонента, артефакту, реалізації та інтерфейсу. Розгортання компонента. Типи компонентних структур. Поняття шаблону. Компонентна модель, компонентне середовище. Каркаси та їх типи. Функції та призначення контейнера. Інтерфейс компонентів. Компоненти повторного використання. Методологія компонентної розробки ПС.

Тема 10. Аспектно-орієнтоване програмування.

Поняття аспекту. Приклади аспектів. Фільтр композиції як один з механізмів композиції компонентів і аспектів. Технологічна схема проектування програмної системи засобами АОП. Середовища для розробки аспектно-орієнтованих програм.

Тема 11. Генерувальне програмування.

Суть генерувального програмування. Предметно-орієнтована мова DSL та її особливості. Генерувальна модель домену GDM та її структура: простір проблеми і простір рішень. Технологія розробки сімейства програм для ПрО.

Тема 12. Сервісно-орієнтоване програмування.

Поняття сервісу, приклади сервісів. Сервісно-орієнтована архітектура SOA. Веб-сервіси та їх компоненти. Основні засоби опису веб-сервісів.

Тема 13. Теоретичне програмування.

Українська теоретична школа програмування В.М. Глушкова. Алгебраїчне та інсерційне програмування (А.А.Летичевський і ін.). Експлікативне та композиційно-номінативне програмування (В.Н.Редько, М.С.Нікітченко). Алгебро-алгоритмічне програмування (Г.О.Цейтлін).

ЗМІСТОВИЙ МОДУЛЬ 3

Методи доведення, верифікації і тестування програмних систем. Моделі якості та надійності програмних систем

Тема 14. Доведення правильності програм. Формальні специфікації.

Формальні специфікації, методи доведення, верифікація і тестування. Поняття часткової і повної правильності програми. Мови специфікації програм і їхня класифікація. Універсальні мови специфікації VDM, Z, RAISE, Spec#. Мови специфікації предметних областей (доменів) у програмуванні. Формальні мови специфікації мов програмування. Форми Бекуса–Наура для опису синтаксису мови програмування. Визначення семантики програм: композиційно-номінативний, денотаційний, операційний і аксіоматичний підходи.

Тема 15. Базові методи доведення правильності програм.

Метод Флойда. Метод Хоара. Метод Маккарті. Метод Дейкстри.

Тема 16. Верифікація і валідація програм.

Призначення верифікації і валідації. Процес верифікації. Процес валідації. Підхід до валідації сценарію вимог. Верифікація об'єктних моделей.

Тема 17. Тестування програмних систем.

Мета та функції тестування. Статичні та динамічні методи тестування. Функціональне тестування. Класифікація помилок і методи їхнього пошуку. Процес тестування за життєвим циклом.

Тема 18. Моделі якості та надійності програмних систем.

Стандартні показники та метрики якості. Стандартна оцінка показників якості. Керування якістю програмних систем. Моделі оцінки надійності програмних систем. Ґрунтовні поняття проблематики надійності. Класифікація моделей надійності.

Теми лабораторних робіт

Етапи життєвого циклу ПЗ у стандарті ISO/IEC 12207.

Побудова моделей життєвого циклу ПЗ: каскадна, інкрементна, спіральна, еволюційна моделі.

Формування загальних вимог до програмних систем.

Застосування об'єктно-орієнтованого підходу до формування вимог до програмних систем.

Побудова інформаційної моделі програмної системи.

Модель станів. Побудова діаграми переходу станів, таблиці переходу станів.

Модель процесів. Потоки керування і даних. Побудова діаграми дій потоків даних.

Проектування архітектури програмної системи.

Елементи структурного програмування. Конструкції

послідовного виконання, розгалуження і циклу.
Основи об'єктно-орієнтованого програмування (Java).
Побудова діаграми класів (UML).
Побудова форм Бекуса–Наура для опису синтаксису мови програмування.
Композиційно-номінативний підхід до доведення правильності програм.
Верифікація і валідація програм.
Тестування програмних систем.
Оцінка якості та надійності програмних систем.

Завдання для самостійної роботи

Інтерфейси, взаємодія, еволюція програм і даних.
Інженерія виробництва програмних продуктів.
Методи керування програмним проектом.

Методи оцінювання та структура оцінки

Оцінювання навчальних досягнень студентів за усіма видами навчальних робіт проводиться за поточним та підсумковим контролем. Поточний контроль знань та контроль самостійної роботи студентів з навчальної дисципліни проводиться:

- за допомогою тестів: контрольні завдання включають тестові питання трьох рівнів складності;
- з лабораторних робіт – за допомогою перевірки виконаних завдань та теоретичної підготовки до занять.

Усі контрольні заходи включено до 100-бальної шкали оцінювання.

Підсумковий семестровий контроль знань відбувається у вигляді заліку, є обов'язковим і проводиться у формі тестування лише для тих студентів, які, з тих чи інших причин, не пройшли модульний контроль. Основними критеріями, що характеризують рівень компетентності студента при оцінюванні результатів поточного та підсумкового контролю з навчальної дисципліни «Практична підготовка з основ програмної інженерії» є:

- виконання всіх видів навчальної роботи, що передбачені робочою програмою навчальної дисципліни;
- глибина і характер знань навчального матеріалу за змістом навчальної дисципліни.
- характер відповідей на поставлені питання (чіткість, лаконічність, логічність, послідовність тощо);
- обґрунтування вибору методу для розв'язання задачі;
- рівень вміння аналізувати одержані результати.

Оцінювання результатів усіх форм контролю передбачено у 100-бальній шкалі.

Критерії оцінювання результатів поточної роботи (завдань, що виконуються на практичних заняттях, результати самостійної роботи студентів) проводиться у % від кількості балів, виділених на завдання, із заокругленням до цілого числа:

0% – завдання не виконано;

40% – завдання виконано частково та містить суттєві

помилки методичного або розрахункового характеру;
 60% – завдання виконано повністю, але містить суттєві помилки у розрахунках або в методиці;
 80% – завдання виконано повністю і вчасно, проте містить окремі несуттєві недоліки (розмірності, висновки, оформлення тощо);
 100% – завдання виконано правильно, вчасно і без зауважень.

<p>Місце навчальної дисципліни в освітній траєкторії здобувача вищої освіти</p>	<p>Дисципліни, що вивчаються супутньо із зазначеною дисципліною: «Програмування», «Операційні системи», «Математична логіка, теорія алгоритмів та програмування» та є базою для вивчення таких дисциплін як, «Практична підготовка з програмування і підтримки веб-застосувань», «Практична підготовка з баз даних», «Людино-машинна взаємодія та основи комп'ютерного дизайну» «Компонентно-орієнтоване програмування», «Сучасні та спеціалізовані мови програмування» та ін.</p>
<p>Інформаційні ресурси</p>	<ol style="list-style-type: none"> 1. Лавріщева К. М. Програмна інженерія: Навчальний посібник /Лавріщева К. М. – К.: Академперіодика, 2008. – 319 с. 2. Sommerville I. Software Engineering – 9th ed. / Ian Sommerville. – Addison-Wesley, 2011. – 773 p. 3. Липаев В.В. Программная инженерия. Методологические основы. / Липаев В.В. – М.: ТЕИС, 2006. – 608 с. 4. Орлов С. А. Программная инженерия. Технология разработки программного обеспечения : учебник / С.А.Орлов. – 5-е изд., обнов. и доп. – Санкт-Петербург : Питер, 2016. - 640 с. 5. Вигерс К., Битти Д. Разработка требований к программному обеспечению. 3-е изд., дополненное / Пер. с англ. — М. : Издательство «Русская редакция»; СПб. : БХВПетербург, 2014. — 736 с. 6. Блэк Р. Ключевые процессы тестирования. Планирование, подготовка, проведение, совершенствование. Пер. с англ./ Р. Блэк. – М: «Лори», 2011. – 456 с. 7. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений. Пер. с англ./ Г. Буч, Р. Максимчук, М. Энгл. – 3-е изд. – М.:ООО «И.Д.Вильямс», 2008. – 720 с. 8. Затонский А. В. Информационные технологии. Разработка информационных моделей и систем. Учебное пособие / А. В. Затонский. – Москва: ИНФРА-М, 2014. – 344 с. 9. Software Engineering - Guide to the Software Engineering Body of Knowledge (SWEBOOK) TECHNICAL REPORT КОЛЕС TR 19759 IEEE First edition 2005-09-15. 10. Герберт Шилдт. Java 8. Полное руководство; 9-е изд.: Пер. с англ. – М. : ООО "И.Д. Вильямс", 2015. – 1376 с. 11.Ткаченко О.М., Каплун В.А. Об'єктно-орієнтоване програмування мовою Java. Навчальний посібник. –

Вінниця: ВНТУ, 2006. – 107 с.

12. Власюк А.П. та ін. Лабораторний практикум з програмування / А.П.Власюк, П.М.Мартинюк, О.В.Прищеп, І.А.Філатова, М.С.Філатов, А.М.Рощенюк, О.С.Демчук, О.Р.Мічута, Т.П.Цветкова, Н.А.Федорчук. – Рівне: Вид-во НУВГП, 2011. – 494 с.

ПРАВИЛА ТА ВИМОГИ (ПОЛІТИКА)*

Дедлайни та перекладання

Перездача тестових завдань перевірки засвоєння теоретичного матеріалу здійснюється згідно з правилами

ННЦНО

<http://nuwm.edu.ua/struktturnipidrozdzili/navch-nauk-tsentr-nezaleznoho-otsiniuvanniaznan/dokumenty>

Студенти повинні вчасно виконувати та здавати завдання лабораторних робіт. Якщо, без вагомої причини, завдання здане невчасно, то бали за нього можуть зніматися. Студент може доздавати завдання на консультаціях та інших парах.

Правила академічної доброчесності

Студенти повинні самостійно виконувати завдання лабораторних робіт. Кожен студент несе індивідуальну відповідальність за виконання поставлених перед ним завдань.

Студентам рекомендується вивчати довідкову літературу та мережу Інтернет.

Студенти, які порушують Кодекс честі університету, не отримують бали за це завдання, а в разі грубих порушень, курс не буде їм зараховано, студенти будуть направлені на повторне вивчення. В цілому студенти та викладачі мають дотримуватись:

- Положення про запобігання плагиату випускних кваліфікаційних робіт здобувачів вищої освіти зі змінами та доповненнями

- Кодекс честі студентів

- Кодекс честі наукових, науково-педагогічних і педагогічних працівників НУВГП

Положення про виявлення та запобігання академічного плагиату в НУВГП – всі документи тут: <http://nuwm.edu.ua/sp/akademichna-dobrochesnistj>

Вимоги до відвідування

Студент повинен відвідувати кожне заняття. У разі пропуску, на це повинна бути поважна причина. В такому разі студент бере завдання у викладача і виконує його самостійно або на консультації.

ДОДАТКОВО

Правила отримання зворотної інформації про дисципліну*

На перших заняттях студентам буде запропоновано відповісти на низку питань щодо врахування в поточному курсі їх побажань.

Оновлення*

Викладач періодично оновлює зміст даної навчальної дисципліни на основі сучасних практик та опитування випускників кафедри

Навчання осіб з інвалідністю

Організація навчання людей з інвалідністю проводиться за дотриманням вимог нормативних документів, розроблених в НУВГП:
<http://nuwm.edu.ua/sp/dlja-osib-z-invalidnistju>

Лектор

Демчук Олена Стеніславівна, к.т.н.