

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ
Декан ФІКТС

« 31 »

Савенко О.С.

2021 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Методології та технології розробки програмних систем

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Рівень вищої освіти – другий (магістерський)

Освітньо-професійна програма – інженерія програмного забезпечення

Статус дисципліни – обов'язкова

Факультет – Програмування та комп'ютерних і телекомунікаційних систем

Кафедра – Комп'ютерної інженерії та інформаційних систем

Форма навчання	Курс	Семестр	Загальне навантаження		Кількість годин							Форма семестрового контролю	
			Кредити ЄКТС	Години	Аудиторні заняття					Курсовий проєкт	Курсова робота	Залік	Іспит
					Разом	Лекції	Лабораторні роботи	Практичні заняття	Індивідуальна робота студента				
Д	1	1	5	150	51	17	34			99	+		+
ДФН			5	150	51	17	34			99	1		1

Робоча програма складена на основі освітньо-професійної програми підготовки магістрів

Програма складена

Підпис

Говорущенко Т.О.

Ініціали, прізвище викладача(ів)

Схвалена на засіданні кафедри Комп'ютерної інженерії та інформаційних систем

Протокол № 1 від 31 серпня 2021 р.

Зав. кафедри Комп'ютерної інженерії та інформаційних систем

Підпис

Говорущенко Т.О.

Ініціали, прізвище

Робоча програма розглянута та схвалена Вченою радою програмування та комп'ютерних і телекомунікаційних систем

Голова Вченої ради

Підпис

Савенко О.С.

Ініціали, прізвище

Хмельницький 2021

МЕТОДОЛОГІЇ ТА ТЕХНОЛОГІЇ РОЗРОБКИ ПРОГРАМНИХ СИСТЕМ

Тип дисципліни	Обов'язкова
Рівень вищої освіти	Другий (магістерський)
Мова викладання	Українська
Семестр	1
Кількість встановлених кредитів ЄКТС	5
Форми навчання, для яких викладається дисципліна	Денна

Результати навчання. Відповідно до Стандарту вищої освіти та освітньої програми дисципліна має забезпечити: **компетентності:** Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення. Здатність розробляти і реалізовувати наукові та/або прикладні проекти у сфері інженерії програмного забезпечення. Здатність проектувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів. Здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення. Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення. Здатність ефективно керувати фінансовими, людськими, технічними та іншими проектними ресурсами у сфері інженерії програмного забезпечення. Здатність критично осмислювати проблеми у галузі інформаційних технологій та на межі галузей знань, інтегрувати відповідні знання та розв'язувати складні задачі у широких або мультидисциплінарних контекстах. Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення. Здатність забезпечувати якість програмного забезпечення.

програмні результати навчання: Знати і застосовувати сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення. Оцінювати і вибирати ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу. Будувати і досліджувати моделі інформаційних процесів у прикладній області. Виявляти інформаційні потреби і класифікувати дані для проектування програмного забезпечення. Розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення. Розробляти і оцінювати стратегії проектування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проектних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів. Аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення. Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника. програмного забезпечення. Конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу. Здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника.

Зміст навчальної дисципліни. Основні особливості й проблеми проектування сучасних програмних засобів, сучасні технології проектування програмного забезпечення, сучасні методології розроблення програмних систем, етап визначення вимог та проектування ПЗ, формальні специфікації ПЗ, сертифікація ПЗ, методи та засоби колективного розроблення ПЗ, оцінка трудомісткості розроблення ПЗ, методи та засоби оцінки якості ПЗ.

Запланована навчальна діяльність: лекції – 17 год., лабораторні заняття – 34 год., самостійна робота – 99 год.; разом – 150 год.

Методи навчання: методи проблемного викладання, словесні, наочні (лекції); пояснювально-ілюстративні, частковопошукові (лабораторні заняття), дослідницькі, частково-пошукові (самостійна робота: індивідуальні завдання).

Форми і методи оцінювання результатів навчання: захист лабораторних робіт, усне опитування, письмові самостійні та контрольні роботи, захист курсового проекту, письмовий іспит **Вид семестрового контролю:** іспит, курсовий проект.

Навчальні ресурси:

1. Говорущенко Т. О. Методологія оцінювання достатності інформації для визначення якості програмного забезпечення: монографія. Хмельницький: Хмельницький національний університет, 2017. 310 с.
2. Бевз О. М. Проектування програмних засобів систем управління: навчальний посібник / О. М. Бевз, В. М. Папінов, Ю. А. Скидан. – Вінниця: ВНТУ, 2010. – 125 с.
3. Лаврищева Е. М. Software Engineering компьютерных систем. Парадигмы, технологии и CASE-средства программирования. – Киев: Наукова думка, 2013. – 283 с.
4. Говорущенко Т. О. Аналіз, дослідження та оцінювання програмних систем: навчальний посібник. – Хмельницький: Хмельницький національний університет, 2019. 358 с.

5. Електронна бібліотека університету. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/plage_lib.php. Викладач: доктор технічних наук, професор Говорущенко Т.О.

3. ПОЯСНЮВАЛЬНА ЗАПИСКА

Дисципліна "Методології та технології розробки програмних систем" є дисципліною професійної підготовки.

Мета дисципліни. Метою дисципліни «Методології та технології розробки програмних систем» є: 1) формування компетентностей, необхідних при проєктуванні програмних систем у складі колективу; 2) ознайомити студентів з сучасними парадигмами та технологіями проєктування програмних систем; 3) надати глибокі та міцні знання з проєктування програмних систем, необхідні для подальшої практичної інженерної та наукової діяльності; 4) ознайомити студентів з теоретичною базою, що використовується при вирішенні задач проєктування програмних систем; 5) виробити у студентів вміння використовувати набуті знання при проєктуванні програмних систем; 6) підготувати студентів до проєктування програмних систем у складних і непередбачуваних умовах, що потребує застосування нових підходів та породження нових ідей (креативності), самостійного пошуку помилок, оцінювання своєї поведінки та результатів мислення і постійного самовдосконалення; 7) підготувати студентів до провадження дослідницької та/або інноваційної діяльності в галузі проєктування програмних систем.

Предмет дисципліни. Методології та технології проєктування та виробництва високоякісних програмних систем.

Завдання дисципліни. Надати студентам знання і практичні навички з проєктування програмних систем; підготувати студентів до провадження дослідницької та/або інноваційної діяльності в галузі проєктування програмних систем, сформувані компетентності, необхідні при проєктуванні програмних систем у складі колективу.

Відповідно до Стандарту вищої освіти із зазначеної спеціальності та освітньої програми дисципліна має забезпечити:

компетентності:

ФК-1. Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення. ФК-2. Здатність розробляти і реалізовувати наукові та/або прикладні проєкти у сфері інженерії програмного забезпечення. ФК-3. Здатність проєктувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів. ФК-4. Здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення. ФК-5. Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення. ФК-6. Здатність ефективно керувати фінансовими, людськими, технічними та іншими проєктними ресурсами у сфері інженерії програмного забезпечення. ФК-7. Здатність критично осмислювати проблеми у галузі інформаційних технологій та на межі галузей знань, інтегрувати відповідні знання та розв'язувати складні задачі у широких або мультидисциплінарних контекстах. ФК-8. Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі

застосування сучасних моделей, методів та технологій розроблення програмного забезпечення. ФК-9. Здатність забезпечувати якість програмного забезпечення..

програмні результати навчання: ПРН-1 Знати і застосовувати сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення ПРН-2 Оцінювати і вибирати ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу.

ПРН-3 Будувати і досліджувати моделі інформаційних процесів у прикладній області. ПРН-4 Виявляти інформаційні потреби і класифікувати дані для проєктування програмного забезпечення. ПРН-5 Розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення. ПРН-6 Розробляти і оцінювати стратегії проєктування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проєктних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів. ПРН-7 Аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення. ПРН-8 Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника. програмного забезпечення. ПРН-13 Конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу. ПРН-15 Здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника.

4. СТРУКТУРА ЗАЛІКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Назва теми	Кількість годин, відведених на:		
	Денна форма		
	Лекції	Лабораторні роботи	СРС
<i>Перший семестр</i>			
Тема 1. Основні особливості й проблеми проєктування сучасних програмних засобів	2		11
Тема 2. Сучасні технології проєктування програмного забезпечення (ПЗ)	2	4	11
Тема 3. Етапи визначення вимог та проєктування ПЗ	4	12	22
Тема 4. Формальні специфікації ПЗ. Сертифікація ПЗ	2		14
Тема 5. Методи та засоби колективного розроблення ПЗ	2	8	11
Тема 6. Оцінка трудомісткості розроблення ПЗ	2		11
Тема 7. Методи та засоби оцінки якості ПЗ на етапі проєктування	2	8	11
Тема 8. Підсумкове заняття	2	4	8
Разом за 1-й семестр:	17*	34*	99

Примітка.

* по чисельнику – 18 годин, по знаменнику – 16 годин (розрахунок здійснюється відповідно до розкладу занять) ** по чисельнику – 36 годин, по знаменнику – 32 години (розрахунок здійснюється відповідно до розкладу занять)

5. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

5.1. Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
	<i>Перший семестр</i>	
1	<p>Лекція 1. Сучасний стан сфери виробництва програмних засобів. Порівняльний аналіз та вибір життєвого циклу розроблення програмного забезпечення. Розповсюджені процеси та етапи розроблення програмних систем</p> <p>Сучасний стан сфери виробництва програмних засобів. Огляд та порівняльний аналіз моделей життєвого циклу ПЗ. Вибір прийнятної моделі життєвого циклу ПЗ. Розповсюджені процеси та етапи розроблення програмних систем.</p> <p>Літ.: [1-18]</p>	2
2	<p>Лекція 2. Основні фази, стандарти та засоби розроблення програмного забезпечення. Сучасні технології проєктування програмного забезпечення</p> <p>Основні фази розроблення ПЗ: формулювання вимог, формулювання цілей проєкту, аналіз прикладної галузі, створення функційної специфікації, проєктування, реалізація. Стандарти в галузі розроблення ПЗ. Програмні засоби підтримки життєвого циклу. Визначення технології проєктування програмного забезпечення (ТППЗ). Загальні вимоги, пропонувані до ТППЗ. Приклади ТППЗ. Моделі систем. Прототипування програмних систем.</p> <p>Літ.: [1-4, 24-29]</p>	2
3	<p>Лекція 3. Етап визначення вимог до програмної системи</p> <p>Методи визначення вимог. Планування етапу визначення вимог. Формалізація вимог: виділення вимог за допомогою прецедентів. Формалізація вимог: псевдокод, кінцеві автомати, графічні дерева рішень, візуальне подання вимог за допомогою діаграм UML. Завдання та результати етапу аналізу вимог.</p> <p>Літ.: [1-4, 25, 30-34]</p>	2
4	<p>Лекція 4. Архітектура програмних систем. Управління ризиками при розробленні програмного забезпечення</p> <p>Планування архітектури. Проєктування архітектури. Документування архітектури. Аналіз архітектури. Поняття ризику. Управління ризиками.</p> <p>Літ.: [1-4, 11, 12, 35-38]</p>	2
5	<p>Лекція 5. Формальні специфікації програмного забезпечення. Сертифікація й оцінювання процесів створення програмного забезпечення</p> <p>Формальні специфікації як засіб підвищення якості ПЗ. Специфікування інтерфейсів. Специфікація поведінки систем. Мови розроблення формальних специфікацій. Сертифікація процесів створення ПЗ.</p> <p>Оцінювання процесів створення ПЗ.</p> <p>Літ.: [1-4, 11, 12, 19-23, 38]</p>	2

6	<p>Лекція 6. Методи та засоби колективного розроблення програмного забезпечення</p> <p>Методологія розроблення ПЗ Microsoft Solutions Framework (MSF). Принципи створення бібліотеки MSF. Модель команди в MSF, ролеві кластери, масштабованість команд та керування компромісами у MSF. Гнучкий підхід до створення ПЗ, основні принципи гнучкого розроблення. Реалізація концепції керування програмним проектом на всіх етапах життєвого циклу у Visual Studio 2012. Функціональні можливості та архітектура TeamFoundationServer 2012 (TFS). Способи розгортання TFS на одному або декількох серверах, в одному домені, робочі групи або в декількох доменах. Шаблони командних проектів TFS, області керування командними проектами. Питання створення командного проекту, зміст програмної інфраструктури проекту, склад і призначення робочих елементів. Аналіз методології Scrum, робочі елементи шаблону MicrosoftVisualStudioScrum 2.2. Організація колективу у методології Scrum. Літ.: [1-4, 11, 12, 38]</p>	2
7	<p>Лекція 7. Оцінка трудомісткості розроблення програмного забезпечення</p> <p>Методика оцінки трудомісткості розробки ПЗ на основі функціональних точок. Алгоритмічне моделювання трудомісткості розробки ПЗ. Літ.: [1-4, 11, 12, 38]</p>	2
8	<p>Лекція 8. Методи та засоби оцінки якості ПЗ на етапі проектування</p> <p>Поняття та модель якості ПЗ. Основні принципи метричного аналізу. Вибір метрик, придатних до використання на етапі проектування ПЗ. Дослідження результатів метричного аналізу. Засоби статичного аналізу програмного коду. Дослідження результатів статичного аналізу. Літ.: [1-4, 11-14, 38]</p>	2
9	Підсумкове заняття	2
Разом за перший семестр:		17*

Примітка. * по чисельнику – 18 годин, по знаменнику – 16 годин

5.2 Зміст лабораторних занять

№ з/п	Тема лабораторного заняття	Кількість годин
<i>Перший семестр</i>		
1	Розроблення технічного завдання. Методологія об'єктноорієнтованого аналізу й проектування Літ.: [1-4]	2
2	Розроблення технічного завдання. Методологія об'єктноорієнтованого аналізу й проектування Літ.: [1-4]	2
3	Розроблення діаграм варіантів використання і діаграм класів. Розроблення діаграм станів і діаграм діяльності Літ.: [1-4]	2
4	Розроблення діаграм варіантів використання і діаграм класів. Розроблення діаграм станів і діаграм діяльності Літ.: [1-4]	2
5	Розроблення діаграм послідовності і діаграм кооперації. Розроблення діаграм компонентів і діаграм розгортання Літ.: [1-4]	2
6	Управління розробкою програмних проєктів Літ.: [15, 33-38]	2
7	Управління розробкою програмних проєктів Літ.: [15, 33-38]	2
8	Командна розробка програмних проєктів у Microsoft Team Foundation Server Літ.: [15, 33-38]	2
9	Командна розробка програмних проєктів у Microsoft Team Foundation Server Літ.: [15, 33-38]	2
10	Командна розробка програмних проєктів у Microsoft Team Foundation Server Літ.: [15, 33-38]	2
11	Командна розробка програмних проєктів у Microsoft Team Foundation Server Літ.: [15, 33-38]	2
12	Верифікація вихідного коду програмного забезпечення (статичний аналіз) Літ.: [33-38]	2
13	Верифікація вихідного коду програмного забезпечення (статичний аналіз) Літ.: [33-38]	2
14	Верифікація вихідного коду програмного забезпечення (динамічний аналіз) Літ.: [33-38]	2
15	Підсумкове заняття	2

Номер тижня	Вид самостійної роботи	К-ть годин
<i>Перший семестр</i>		
1-2	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №1. Робота над курсовим проєктом	11
3-4	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №1. Підготовка до лабораторної роботи №2. Робота над курсовим проєктом	11
5-6	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №2. Підготовка до лабораторної роботи №3. Робота над курсовим проєктом	11
7-8	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №3. Підготовка до лабораторної роботи №4. Робота над курсовим проєктом	11
9-10	Підготовка до тестування за темами 1-4. Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №4. Підготовка до лабораторної роботи №5. Робота над курсовим проєктом	14
11-12	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №5. Робота над курсовим проєктом	11
13-14	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №5. Підготовка до лабораторної роботи №6. Робота над курсовим проєктом	11
15-16	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №6. Підготовка до лабораторної роботи №7. Підготовка до комплексної контрольної роботи. Підготовка до захисту курсового проєкту	11
17	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи №7. Підготовка до захисту курсового проєкту	8
Разом за 1-й семестр:		99

6. МЕТОДИ НАВЧАННЯ

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних методів: методи проблемного викладання, словесні, наочні (лекції); пояснювально-ілюстративні, частковопошукові (лабораторні заняття), дослідницькі, частково-пошукові (самостійна робота: індивідуальні завдання).

Лекції проводяться в основному словесними методами, а лабораторні заняття проводяться з використанням інформаційних технологій, майстер-класів, практикумів і мають за мету – Надати студентам знання і практичні навички з проєктування програмних систем; підготувати студентів до провадження дослідницької та/або інноваційної діяльності в галузі проєктування програмних систем, сформувати компетентності, необхідні при проєктуванні програмних систем у складі колективу.

7. ФОРМИ І МЕТОДИ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Поточний контроль здійснюється під час лекційних та лабораторних занять. Семестровий контроль проводиться у формі іспиту. При виведенні остаточної оцінки враховуються результати поточного контролю та письмового іспиту.

Процес оцінювання підготовленості студента можна розділити на етапи:

Перший етап оцінювання направлений на визначення знань інформаційного мінімуму. Якщо студент твердо засвоїв визначену навчальним планом суму формальних знань, то це означає, що він вміє використати їх при вирішенні різних питань при проєктуванні програмних систем, вміє розширити їх.

Перед вивченням дисципліни, як правило, проводиться вхідний контроль знань з дисциплін, що їй передують і забезпечують. При цьому необхідно встановити рівні та критерії сформованості знань щодо змісту навчальних елементів. Такими рівнями є:

Ознайомчо-орієнтовний (ОО) – особа має орієнтовне уявлення щодо понять, які вивчаються, здатна: програмувати основні елементи програмних систем різними мовами програмування, обирати сучасні методології та технології проєктування програмного забезпечення, обґрунтовано використовувати сучасні середовища розроблення програмного забезпечення для розроблення програмних систем.

Понятійно-аналітичний (ПА) – особа має чітке уявлення щодо навчального об'єкту, здатна перенести раніше засвоєні знання на типові ситуації.

Продуктивно-синтетичний (ПС) – особа має глибоке розуміння щодо навчального об'єкту, здатна здійснювати синтез, генерувати нові ідеї та уявлення, переносити раніше засвоєні знання на нетипові, нестандартні ситуації.

Кожний вид роботи з дисципліни оцінюється за *чотирибальною* шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих *позитивно* з врахуванням коефіцієнта вагомості. Вагові коефіцієнти змінюються залежно від структури дисципліни і важливості окремих її видів робіт. Студент, який набрав позитивний середньозважений бал за поточну роботу і не здав контрольний захід (іспит), вважається невстигаючим.

При оцінюванні знань студентів використовуються різні засоби контролю, зокрема: усне опитування перед допуском до виконання лабораторної роботи – здійснюється на її початку; засвоєння теоретичного матеріалу з тем перевіряється тестовим контролем; якість виконання, набуття теоретичних знань і практичних навичок перевіряється шляхом захисту кожної лабораторної роботи згідно з робочою програмою дисципліни і робочим навчальним планом.

Оцінка, яка виставляється за *лабораторне заняття*, складається з таких елементів: усне опитування студентів перед допуском до виконання лабораторної роботи; знання теоретичного матеріалу з теми; якість оформлення протоколу і графічної частини; вміння студента обґрунтувати прийняті конструктивні рішення; своєчасний захист лабораторної роботи. Для виконання програми дисципліни студент повинен отримати 7 оцінок за лабораторні роботи.

Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті. За несвоєчасний захист лабораторної роботи з неповажної причини студент за позитивну відповідь отримує оцінку «задовільно».

Пропущене лабораторне заняття студент повинен відпрацювати в лабораторіях кафедри у встановлений викладачем термін з реєстрацією у відповідному журналі кафедри, але не пізніше, ніж за два тижні до кінця теоретичних занять у семестрі.

При оцінюванні знань студентів викладач керується такими критеріями.

Оцінку „відмінно” отримує студент за глибоке і повне опанування змісту навчального матеріалу, в якому він легко орієнтується, понятійного апарату, за уміння зв'язувати теорію з практикою, вирішувати практичні завдання, висловлювати і обґрунтовувати свої судження. Відмінна оцінка передбачає грамотний, логічний виклад відповіді (як в усній, так і в письмовій формі), якісне зовнішнє оформлення. Студент повинен набути практичних навичок із проєктування та програмної реалізації програмних систем.

Оцінка "відмінно" виставляється студенту, який глибоко засвоїв основні принципи проєктування програмних систем та вміє їх раціонально застосувати, знає методики та вміє ними користуватися при розробленні програмного забезпечення. Студент не повинен вагатися при видозміні запитання, повинен робити детальні та узагальнюючі висновки.

Оцінку „добре” отримує студент за повне засвоєння навчального матеріалу, володіння понятійним апаратом, орієнтування у вивченому матеріалі, свідоме використання знань для вирішення практичних завдань, грамотний виклад відповіді, але у змісті і формі відповіді мали місце окремі неточності (похибки), нечіткі формулювання закономірностей тощо. Відповідь студента повинна будуватись на основі самостійного мислення.

Оцінку „добре” отримує студент за правильну відповідь з однією-двома суттєвими помилками.

Оцінки "задовільно" заслуговує студент, який виявив знання основного навчально-програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, що справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь студента будується на рівні репродуктивного мислення, студент слабо знає структуру курсу, допускає помилки у відповіді, засвоїв і набув практичних навичок у проєктуванні та реалізації програмних систем, але припустився неточностей. Вагається при відповіді на видозмінене запитання, разом з тим студент володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.

Оцінки "задовільно" заслуговує студент за неповне опанування програмного матеріалу, але отримані знання і набуті практичні навички із проєктування та розроблення програмного забезпечення.

Оцінка „незадовільно” виставляється, коли студент має розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань. Як правило, оцінка "незадовільно" виставляється студенту, який не може продовжити навчання без додаткових знань з курсу.

На основі результатів поточного контролю і іспиту виставляється підсумкова семестрова оцінка. На основі аналізу контролю знань викладач удосконалює курс лекцій, звертаючи особливу увагу на ті розділи, чи теми, з яких було найбільше неточних відповідей, що свідчить про методичні чи інші недоліки при висвітленні вказаних тем або розділів.

Аналогічно вносяться корективи в методичні посібники для лабораторних робіт, детальніше розглядаються принципові питання при виконанні лабораторних робіт та їх захисті.

Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів денної форми навчання у семестрі за ваговими коефіцієнтами

Аудиторна робота	Самостійна, індивідуальна робота	Іспит
<i>I семестр</i>		

Лабораторні роботи №:							Контроль:		Іспит
1	2	3	4	5	6	7	ТК Т 1-4	ТК Т 5-8	
ВК: 0,4							0,2		0,4

Умовні позначення: ТК – тестовий контроль; Т – тема дисципліни; ВК – ваговий коефіцієнт.

Структурування курсового проєктування для дисципліни за видами робіт і оцінювання результатів курсового проєктування студентів у семестрі за ваговими коефіцієнтами

Самостійна, індивідуальна робота				
І семестр				
Роботи (контрольні точки) №			Захист	
1	2	3		
ВК: 0,8			0,2	

Умовні позначення: ВК – ваговий коефіцієнт.

Оцінювання тестових завдань. Тематичний тест для кожного студента складається з двадцяти тестових завдань, кожне з яких оцінюється одним балом. Максимальна сума балів, яку може набрати студент, складає 20.

Оцінювання здійснюється за чотирибальною шкалою.

Відповідність набраних балів за тестове завдання оцінці, що виставляється студенту, представлена у нижченаведеній таблиці.

Сума балів за тестове завдання	1–11	12–14	15–18	19-20
Оцінка	2	3	4	5

На тестування відводиться 20 хвилин. Тестування проводиться з використанням модульного середовища для навчання MOODLE. Правильні відповіді студент реєструє в он-лайн режимі в модульному середовищі MOODLE. Через 20 хвилин студенти завершують тестування та надсилають свої відповіді на сервер. Викладач оголошує результати тестування згідно журналу оцінок модульного середовища MOODLE.

Якщо студент отримав негативну оцінку, то він має перездати її в установленому порядку, але обов'язково до терміну наступного контролю. У випадку, коли студент не виконав індивідуальний план з дисципліни у заплановані терміни без поважних причин, то під час відпрацювання заборгованості при позитивній відповіді йому виставляється оцінка „задовільно”.

Підсумкова семестрова оцінка за національною шкалою і шкалою ЄКТС встановлюється в автоматизованому режимі після внесення усіх оцінок до електронного журналу. Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС наведені у наступній таблиці.

Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	Інтервальна шкала балів	Вітчизняна оцінка	
A	4,75–5,00	5	Відмінно – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навиків

B	4,25–4,74	4	<i>Добре</i> – повне знання навчального матеріалу з кількома незначними помилками
C	3,75–4,24	4	<i>Добре</i> – в загальному правильна відповідь з двома-трьома суттєвими помилками
D	3,25–3,74	3	<i>Задовільно</i> – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією
E	3,00–3,24	3	<i>Задовільно</i> – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00–2,99	2	<i>Незадовільно</i> – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00–1,99	2	<i>Незадовільно</i> – необхідна серйозна подальша робота і повторне вивчення дисципліни

8. ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ РЕЗУЛЬТАТІВ НАВЧАННЯ

1. Огляд та порівняльний аналіз моделей життєвого циклу ПЗ.
2. Вибір прийнятної моделі життєвого циклу ПЗ.
3. Сучасний стан сфери виробництва програмних засобів.
4. Розповсюджені процеси та етапи розроблення програмних систем.
5. Сертифікація процесів створення ПЗ.
6. Оцінювання процесів створення ПЗ.
7. Визначення технології проєктування програмного забезпечення (ТППЗ).
8. Загальні вимоги, пропоновані до ТППЗ.
9. Приклади ТППЗ.
10. Моделі програмних систем.
11. Прототипування програмних систем.
12. Основні фази розроблення ПЗ: формулювання вимог, формулювання цілей проєкту, аналіз прикладної галузі, створення функційної специфікації, проєктування, реалізація.
13. Стандарти в галузі розроблення ПЗ.
14. Програмні засоби підтримки життєвого циклу.
15. Методи визначення вимог.
16. Планування етапу визначення вимог.
17. Формалізація вимог: виділення вимог за допомогою прецедентів.
18. Формалізація вимог: псевдокод, кінцеві автомати, графічні дерева рішень, візуальне подання вимог за допомогою діаграм UML.
19. Завдання та результати етапу аналізу вимог.
20. Планування архітектури.
21. Проєктування архітектури.
22. Документування архітектури.
23. Аналіз архітектури.
24. Поняття ризику.
25. Управління ризиками.
26. Формальні специфікації як засіб підвищення якості ПЗ.
27. Специфікування інтерфейсів.
28. Специфікація поведінки систем.

29. Мови розроблення формальних специфікацій.
30. Методологія розроблення ПЗ Microsoft Solutions Framework (MSF).
31. Принципи створення бібліотеки MSF.
32. Модель команди в MSF, ролеві кластери, масштабованість команд та керування компромісами у MSF.
33. Гнучкий підхід до створення ПЗ, основні принципи гнучкого розроблення.
34. Реалізація концепції керування програмним проектом на всіх етапах життєвого циклу у Visual Studio 2012.
35. Функціональні можливості та архітектура TeamFoundationServer 2012 (TFS).
36. Способи розгортання TFS на одному або декількох серверах, в одному домені, робочі групи або в декількох доменах.
37. Шаблони командних проєктів TFS, області керування командними проєктами.
38. Питання створення командного проєкту, зміст програмної інфраструктури проєкту, склад і призначення робочих елементів.
39. Аналіз методології Scrum, робочі елементи шаблону MicrosoftVisualStudioScrum. Організація колективу у методології Scrum.
40. Методика оцінки трудомісткості розробки ПЗ на основі функціональних точок.
41. Алгоритмічне моделювання трудомісткості розробки ПЗ.
42. Поняття та модель якості ПЗ.
43. Основні принципи метричного аналізу.
44. Вибір метрик, придатних до використання на етапі проєктування ПЗ.
45. Дослідження результатів метричного аналізу.
46. Засоби статичного аналізу програмного коду.
47. Дослідження результатів статичного аналізу.

9. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Навчальний процес з дисципліни «Методології та технології розробки програмних систем» повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою. Зокрема, викладачами кафедри підготовлені і видані:

1. Говорущенко Т. О. Аналіз, дослідження та оцінювання програмних систем: навчальний посібник. – Хмельницький: Хмельницький національний університет, 2019. 358 с.

2. Говорущенко Т. О. Методологія оцінювання достатності інформації для визначення якості програмного забезпечення: монографія. Хмельницький: Хмельницький національний університет, 2017. 310 с.

3. Мищенко В. О. CASE-оценка критических программных систем: в 3 т. Т. 1. Качество: монография / Мищенко В. О., Поморова О. В., Говорущенко Т. А. ; под ред. В. С. Харченка. Харьков: Нац. аэрокосмический университет «ХАИ», 2012. 201 с.

4. Поморова О.В., Говорущенко Т.О. Технологія проєктування програмних систем: конспект лекцій - Хмельницький: ХНУ, 2014. - 292 с.

5. Говорущенко Т.О. Технологія проєктування програмних систем: Методичні вказівки для виконання лабораторних робіт для студентів освітнього ступеня «магістр» спеціальності 123 “Комп’ютерна інженерія” денної форми навчання. – Хмельницький: ХНУ, 2017. – 90 с.

10. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Бевз О. М. Проектування програмних засобів систем управління: навчальний посібник / О. М. Бевз, В. М. Папінов, Ю. А. Скидан. – Вінниця: ВНТУ, 2010. – 125 с.
2. Sommerville I. Software Engineering. London: Pearson, 2015. 816 p.
3. Maedche A., Botzenhardt A., Neer L. Software for people: fundamentals, trends and best practices (Management for professionals). Springer-Verlag Berlin Heidelberg, 2012. 293 p.
4. Munch J., Schmid K. Perspectives on the future of software engineering. Springer-Verlag Berlin Heidelberg, 2013. 365 p.
5. Bird Ch., Menzies T., Zimmermann Th. The art and science of analyzing software data. Morgan Kaufmann, 2015. 672 p.
6. Diaz V. G., Lovelle J. M., Garcia-Bustelo B. C. Handbook of research on innovations in systems and software engineering. Hershey, 2015. 745 p.
7. Levenson N. G. Engineering a safer world: systems thinking applied to safety. MIT Press, 2012. 560 pp.
8. Wiegers K., Beatty J. Software requirements: 3rd edition. Washington: MS Press, 2013. 640 p.
9. Chen A., Beatty J. Visual models for software requirements. Washington: MS Press, 2012. 444 p.
10. Relating system quality and software architecture / I. Mistrik and others. Elsevier Inc., 2014. 379 p.
15. Abran A. Software project estimation: the fundamentals for providing high quality information to decision makers. Wiley-IEEE Computer Society Press, 2015. 288 p.
16. Fenton N., Bieman J. Software metrics: a rigorous and practical approach. CRC Press, 2014. 617 p.
17. ISO/IEC/IEEE 24765:2010. Systems and software engineering. Vocabulary. [Introduced 15.12.2010]. Geneva (Switzerland), 2010. 410 p. (International standard).
18. ISO/IEC TR 19759:2015. Software Engineering. Guide to the software engineering body of knowledge (SWEBOOK). [Introduced 01.10.2015]. Geneva (Switzerland), 2015. 336 p. (International standard).
19. ISO/IEC TR 12182:2015. Systems and software engineering. Framework for categorization of IT systems and software, and guide for applying it. [Introduced 01.05.2015]. Geneva (Switzerland), 2015. 20 p. (International standard).
20. ISO/IEC/IEEE CD 12207:2016. Systems and software engineering. Software life cycle processes. [Introduced 21.09.2016]. Geneva (Switzerland), 2016. 162 p. (International standard).
21. ISO/IEC/IEEE DIS 15939:2016. Systems and software engineering. Measurement process. [Introduced 19.10.2016]. Geneva (Switzerland), 2016. 36 p. (International standard).
22. ISO/IEC 25000 series of standards. Web-site. URL: <http://iso25000.com/index.php/en/iso25000-standards> (Last accessed: December 7, 2017).
23. Why the Waterfall Model Doesn't Work // <http://www.infoq.com/resource/articles/scalingsoftware-agility/en/resources/ch02.pdf>

11. ІНФОРМАЦІЙНІ РЕСУРСИ

1. Модульне середовище для навчання. Доступ до ресурсу: <https://msn.khnu.km.ua>.
2. Електронна бібліотека університету. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/plage_lib.php.

3. Репозитарій ХНУ. Доступ до ресурсу: <http://elar.khnu.km.ua/jspui/?locale=uk>.