

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ



ЗАТВЕРДЖУЮ

Декан факультету Ретяна ГОВОРУЩЕНКО ФІТ
05.09.2024 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Технології проєктування програмних систем

Назва

Галузь знань 12 – Інформаційні технології
Спеціальність 123 – Комп'ютерна інженерія
Освітня програма Комп'ютерна інженерія та програмування (освітньо-наукова)
Статус дисципліни: обов'язкова, дисципліна науково-дослідної підготовки
Факультет – Інформаційних технологій
Кафедра – Комп'ютерної інженерії та інформаційних систем

Форма здобуття освіти	Курс	Семестр	Загальне навантаження		Кількість годин						Курсовий проєкт	Курсова робота	Форма семестрового контролю	
			Кредити ЄКТС	Години	Аудиторні заняття				Індивідуальна робота студента	Самостійна робота, в т.ч. ІРС			Залік	Іспит
					Разом	Лекції	Лабораторні роботи	Практичні заняття						
Д	1	1	5	150	51	17	34			99	+			+
	ДФН		5	150	51	17	34			99	1			1

Робоча програма складена на основі освітньо-наукової програми підготовки магістрів «Комп'ютерна інженерія та програмування»

Програма складена Говорущенко Т.О.
Підпис Ініціали, прізвище викладача(ів)

Схвалена на засіданні кафедри Комп'ютерної інженерії та інформаційних систем

Протокол № 2 від 30 серпня 2024 р.

В.о. зав. кафедри комп'ютерної інженерії та інформаційних систем Засорнова І.О.
Підпис Ініціали, прізвище

Робоча програма розглянута та схвалена Вченою радою факультету інформаційних технологій

Голова Вченої ради Говорущенко Т.О.
Підпис Ініціали, прізвище

Хмельницький 2024

ВСТУП

Мета викладання дисципліни. Дисципліна "Технології проектування програмних систем" є однією зі спеціальних профільюючих дисциплін і тому займає провідне місце у підготовці магістрів комп'ютерної інженерії.

Метою дисципліни «Технології проектування програмних систем» є: 1) формування компетентностей, необхідних при проектуванні програмних систем у складі колективу; 2) ознайомити студентів з сучасними парадигмами та технологіями проектування програмних систем; 3) надати глибокі та міцні знання з проектування програмних систем, необхідні для подальшої практичної інженерної та наукової діяльності; 4) ознайомити студентів з теоретичною базою, що використовується при вирішенні задач проектування програмних систем; 5) виробити у студентів вміння використовувати набуті знання при проектуванні програмних систем; 6) підготувати студентів до проектування 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 проектування програмних систем.

Предмет дисципліни. Методи та технології проектування та виробництва високоякісних програмних систем.

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

Компетентності, на формування яких спрямовано ОК:

Інтегральна - Здатність розв'язувати складні задачі і проблеми в галузі комп'ютерної інженерії або у процесі навчання, що передбачає проведення досліджень та/або здійснення інновацій та характеризується невизначеністю умов і вимог.

ЗК3. Здатність проводити дослідження на відповідному рівні

ЗК6. Здатність виявляти, ставити та вирішувати проблеми

ЗК8. Здатність спілкуватися іноземною мовою

ЗК9. Міжособистісні навички і вміння

ЗК10. Базові дослідницькі навички і вміння

СК1. Здатність до визначення технічних характеристик, конструктивних особливостей, застосування і експлуатації програмних, програмно-технічних засобів, комп'ютерних систем та мереж різного призначення

СК2. Здатність розробляти алгоритмічне та програмне забезпечення, компоненти комп'ютерних систем та мереж, Інтернет додатків, кіберфізичних систем з використанням сучасних методів і мов програмування, а також засобів і систем автоматизації проектування

СК5. Здатність будувати архітектуру та створювати системне і прикладне програмне забезпечення комп'ютерних систем та мереж

СК7. Здатність досліджувати, розробляти та обирати технології створення великих і надвеликих систем

СК8. Здатність забезпечувати якість продуктів і сервісів інформаційних технологій на протязі їх життєвого циклу

СК9. Здатність представляти результати власних досліджень та/або розробок у вигляді презентацій, науково-технічних звітів, статей і доповідей на науково-технічних конференціях

СК10. Здатність ідентифікувати, класифікувати та описувати роботу програмно-технічних засобів, комп'ютерних систем, мереж та їхніх компонентів

СК11. Здатність обирати ефективні методи розв'язування складних задач комп'ютерної інженерії, критично оцінювати отримані результати та аргументувати прийняті рішення

СК12. Здатність використовувати методи аналізу, ідентифікації й синтезу комп'ютерних систем та мереж, кіберфізичних систем, засобів Інтернету речей та ІТ-інфраструктур

СК13. Здатність проектувати системи та їхні компоненти з урахуванням усіх аспектів їх життєвого циклу та поставленої задачі, включаючи створення, налаштування, експлуатацію, технічне обслуговування та утилізацію.

СК14. Здатність проектувати та розробляти інтерфейси користувача програмних систем, враховуючи їх призначення та зручність використання.

СК17. Здатність досліджувати проблему в галузі комп'ютерних та інформаційних технологій, визначати їх обмеження.

Програмні результати навчання, на забезпечення яких спрямовано ОК:

ПРН1. Застосовувати загальні підходи пізнання, методи математики, природничих та інженерних наук до розв'язання складних задач комп'ютерної інженерії

ПРН4. Застосовувати спеціалізовані концептуальні знання, що включають сучасні наукові здобутки у сфері комп'ютерної інженерії, необхідні для професійної діяльності, оригінального мислення та проведення досліджень, критичного осмислення проблем інформаційних технологій та на межі галузей знань.

ПРН5. Розробляти і реалізовувати проекти у сфері комп'ютерної інженерії та дотичні до неї міждисциплінарні проекти з урахуванням інженерних, соціальних, економічних, правових та інших аспектів.

ПРН6. Аналізувати проблематику, ідентифікувати та формулювати конкретні проблеми, що потребують вирішення, обирати ефективні методи їх вирішення.

ПРН8. Застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп'ютерних систем та мереж для вирішення складних задач комп'ютерної інженерії та дотичних проблем.

ПРН9. Розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем

ПРН12. Вільно спілкуватись усно і письмово українською мовою та однією з іноземних мов (англійською, німецькою, італійською, французькою, іспанською) при обговоренні професійних питань, досліджень та інновацій в галузі інформаційних технологій.

ПРН13. Зрозуміло і недвозначно доносити власні знання, висновки та аргументацію з питань інформаційних технологій і дотичних міжгалузевих питань до фахівців і нефахівців, зокрема, до осіб, які навчаються.

ПРН14. Планувати і виконувати наукові дослідження в сфері комп'ютерної інженерії, формулювати і перевіряти гіпотези, обирати методики та інструменти, аналізувати результати, обґрунтовувати висновки.

ПРН15. Вміти виконувати експериментальні дослідження за професійною тематикою

ПРН17. Вміти проектувати та розробляти системне та прикладне програмне забезпечення, включаючи інтерфейси користувача програмних систем, враховуючи його призначення, необхідний рівень якості та зручність використання (usability)

ПРН 19. Вміти ефективно працювати як індивідуально, так і у складі команди

ТЕХНОЛОГІЇ ПРОЄКТУВАННЯ 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. Студент, який успішно завершив вивчення дисципліни, повинен: застосовувати загальні підходи пізнання, методи математики, природничих та інженерних наук до розв'язання складних задач комп'ютерної інженерії; застосовувати спеціалізовані концептуальні знання, що включають сучасні наукові здобутки у сфері комп'ютерної інженерії, необхідні для професійної діяльності, оригінального мислення та проведення досліджень, критичного осмислення проблем інформаційних технологій та на межі галузей знань; розробляти і реалізовувати проекти у сфері комп'ютерної інженерії та дотичні до неї міждисциплінарні проекти з урахуванням інженерних, соціальних, економічних, правових та інших аспектів; аналізувати проблематику, ідентифікувати та формулювати конкретні проблеми, що потребують вирішення, обирати ефективні методи їх вирішення; застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп'ютерних систем та мереж для вирішення складних задач комп'ютерної інженерії та дотичних проблем; розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем; вільно спілкуватись усно і письмово українською мовою та однією з іноземних мов (англійською, німецькою, італійською, французькою, іспанською) при обговоренні професійних питань, досліджень та інновацій в галузі інформаційних технологій; зрозуміло і недвозначно доносити власні знання, висновки та аргументацію з питань інформаційних технологій і дотичних міжгалузевих питань до фахівців і нефахівців, зокрема, до осіб, які навчаються; планувати і виконувати наукові дослідження в сфері комп'ютерної інженерії, формулювати і перевіряти гіпотези, обирати методи та інструменти, аналізувати результати, обґрунтовувати висновки; вміти виконувати експериментальні дослідження за професійною тематикою; вміти проєктувати та розробляти системне та прикладне програмне забезпечення, включаючи інтерфейси користувача програмних систем, враховуючи його призначення, необхідний рівень якості та зручність використання (usability); вміти ефективно працювати як індивідуально, так і у складі команди.

Зміст навчальної дисципліни. Основні особливості й проблеми проєктування сучасних програмних засобів, сучасні технології проєктування програмного забезпечення, сучасні методології розроблення програмних систем, етап визначення вимог та проєктування 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: словесні, наочні, проблемно-пошукові (лекції); пояснювально-ілюстративні, практичні, проблемно-пошукові, частково-пошукові (лабораторні заняття), практичні, дослідницькі, частково-пошукові (самостійна робота: індивідуальні завдання).

Форми і методи оцінювання результатів навчання: захист лабораторних робіт, тестування, захист курсового проєкту, підсумковий контрольний захід.

Вид семестрового контролю: іспит, курсовий проєкт.

Навчальні ресурси:

1. Говорущенко Т. О. Аналіз, дослідження та оцінювання програмних систем: навчальний посібник. – Хмельницький: Хмельницький національний університет, 2019. 358 с.
2. Говорущенко Т.О. Технологія проєктування програмних систем: Методичні вказівки для виконання лабораторних робіт для магістрів спеціальності 123 “Комп’ютерна інженерія” денної форми навчання. – Хмельницький: ХНУ, 2021. – 90 с.
3. Говорущенко Т.О. Технології проєктування програмних систем: Методичні вказівки до курсового проєктування для здобувачів вищої освіти за освітньо-науковою програмою “Комп’ютерна інженерія та програмування” спеціальності 123 “Комп’ютерна інженерія” другого (магістерського) рівня вищої освіти. – Хмельницький: ХНУ, 2022. – 65 с.
4. C. Hu. An Introduction to Software Design: Concepts, Principles, Methodologies, and Techniques 1st ed. 2023 Edition
5. M. Chemuturi. Software Design: A Comprehensive Guide to Software Development Projects 1st Edition, 2018.
6. Модульне середовище для навчання MOODLE. Доступ до ресурсу: <https://msn.khnu.edu.ua>.
7. Електронна бібліотека університету. Доступ до ресурсу: <http://lib.khmnu.edu.ua>.

Викладач: доктор технічних наук, професор Говорущенко Т.О.

1. 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)

2. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

2.1. Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
	<i>Перший семестр</i>	
1	<p>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.: [1-25]</p>	2
2	<p>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, пропоновані до ТППЗ. Приклади ТППЗ. Моделі систем. Прототипування програмних систем.</p> <p>Літ.: [1-25]</p>	2
3	<p>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.: [1-25]</p>	2
4	<p>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.: [1-25]</p>	2
5	<p>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. Оцінювання процесів створення ПЗ.</p> <p>Літ.: [1-25]</p>	2
6	<p>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 проєктом на всіх етапах життєвого циклу у Visual Studio. Функціональні можливості та архітектура</p>	2

	TeamFoundationServer (TFS). Способи розгортання TFS на одному або декількох серверах, в одному домені, робочі групи або в декількох доменах. Шаблони командних проєктів TFS, області керування командними проєктами. Питання створення командного проєкту, зміст програмної інфраструктури проєкту, склад і призначення робочих елементів. Аналіз методології Scrum, робочі елементи шаблону MicrosoftVisualStudioScrum. Організація колективу у методології Scrum. Літ.: [1-25]	
7	Оцінка трудомісткості розроблення програмного забезпечення Методика оцінки трудомісткості розробки ПЗ на основі функціональних точок. Алгоритмічне моделювання трудомісткості розробки ПЗ. Літ.: [1-25]	2
8	Методи та засоби оцінки якості ПЗ на етапі проєктування Поняття та модель якості ПЗ. Основні принципи метричного аналізу. Вибір метрик, придатних до використання на етапі проєктування ПЗ. Дослідження результатів метричного аналізу. Засоби статичного аналізу програмного коду. Дослідження результатів статичного аналізу. Літ.: [1-25]	2
9	Підсумкове заняття	2
	Разом за перший семестр:	17*

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

2.2 Зміст лабораторних занять

№ з/п	Тема лабораторного заняття	Кількість годин
<i>Перший семестр</i>		
1	Розроблення технічного завдання. Методологія об'єктно-орієнтованого аналізу й проєктування Літ.: [1-4]	4
2	Розроблення діаграм варіантів використання і діаграм класів. Розроблення діаграм станів і діаграм діяльності Літ.: [1-4]	4
3	Розроблення діаграм послідовності і діаграм кооперації. Розроблення діаграм компонентів і діаграм розгортання Літ.: [1-4]	4
4	Управління розробкою програмних проєктів Літ.: [1-4]	4
5	Командна розробка програмних проєктів у Microsoft Team Foundation Server Літ.: [1-4]	8
6	Верифікація вихідного коду програмного забезпечення (статичний аналіз) Літ.: [1-4]	4
7	Верифікація вихідного коду програмного забезпечення (динамічний аналіз) Літ.: [1-4]	4
8	Підсумкове заняття	4
Разом за 1-й семестр		34*

Примітка. * по чисельнику – 36 годин, по знаменнику – 32 години

2.3 Зміст самостійної (індивідуальної) роботи

Самостійна робота студентів денної форми навчання полягає у систематичному опрацюванні програмного матеріалу, підготовці до виконання і захисту лабораторних робіт, тестування з теоретичного матеріалу, тощо.

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

3. МЕТОДИ НАВЧАННЯ

Лекції проводяться, в основному, з використанням словесних, наочних, проблемно-пошукових методів; лабораторні заняття проводяться пояснювально-ілюстративними методами, практичними, проблемно-пошуковими та частково-пошуковими методами; самостійна робота передбачає виконання індивідуальних завдань із залученням практичних, дослідницьких, частково-пошукових методів.

4. ФОРМИ І МЕТОДИ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Поточний контроль здійснюється під час лекційних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочим планом дисципліни. Семестровий контроль проводиться у формі іспиту (підсумкового контрольного заходу). При цьому при виведенні остаточної оцінки враховуються результати поточного контролю.

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 проектування 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 підсумковий 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.

При оцінюванні знань студентів викладач керується такими критеріями.

Оцінку „відмінно”, за шкалою ECTS – А, 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 проектування та програмної реалізації програмних систем. Оцінка "відмінно" виставляється студенту, який глибоко засвоїв основні принципи проектування програмних систем та вміє їх раціонально застосувати, знає методики та вміє ними користуватися при розробленні програмного забезпечення. Студент не повинен вагатися при відозміні запитання, повинен робити детальні та узагальнюючі висновки.

Оцінку „добре”, за шкалою ECTS – В, 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”, за шкалою ECTS – C, отримує студент за правильну відповідь з однією-двома суттєвими помилками.

Оцінки "задовільно", за шкалою ECTS – D, 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 проектуванні 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", за шкалою ECTS – E, заслуговує студент за неповне опанування програмного матеріалу, але отримані знання і набуті практичні навички із проектування та розроблення програмного забезпечення.

Оцінка „незадовільно”, за шкалою ECTS – FX, 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", за шкалою ECTS – F, виставляється студенту, який не може продовжити навчання без додаткових знань з курсу.

На основі результатів поточного контролю і підсумкового контрольного заходу 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	Семестровий контроль (іспит)
<i>I семестр</i>								
Лабораторні роботи №:							Контроль:	Підсумковий контрольний захід
1	2	3	4	5	6	7	ТК Т 1-4	
ВК: 0,4							0,2	0,4

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

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

Самостійна, індивідуальна робота			Підсумковий контроль	
<i>I семестр</i>				
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

Тестування проводиться з використанням модульного середовища для навчання MOODLE. Правильні відповіді студент реєструє в он-лайн режимі в модульному середовищі MOODLE. Викладач виставляє результати тестування згідно журналу оцінок модульного середовища MOODLE.

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	<i>Відмінно</i> – 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

5. ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ СТУДЕНТІВ ФОРМ 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 проєктом на всіх етапах життєвого циклу у Visual Studio.
35. Функціональні можливості та архітектура TeamFoundationServer (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.

6. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Освітній процес з дисципліни «Технології проєктування 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. Говорущенко Т.О. Технологія проєктування програмних систем: Методичні вказівки для виконання лабораторних робіт для магістрів спеціальності 123 “Комп’ютерна інженерія” денної форми навчання. – Хмельницький: ХНУ, 2021. – 90 с.

4. Говорущенко Т.О. Технології проєктування програмних систем: Методичні вказівки до курсового проєктування для здобувачів вищої освіти за освітньо-науковою програмою “Комп’ютерна інженерія та програмування” спеціальності 123 “Комп’ютерна інженерія” другого (магістерського) рівня вищої освіти. – Хмельницький: ХНУ, 2022. – 65 с.

7. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

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. Говорущенко Т.О. Технологія проектування програмних систем: Методичні вказівки для виконання лабораторних робіт для магістрів спеціальності 123 “Комп’ютерна інженерія” денної форми навчання. – Хмельницький: ХНУ, 2021. – 90 с.
4. Говорущенко Т.О. Технології проектування програмних систем: Методичні вказівки до курсового проектування для здобувачів вищої освіти за освітньо-науковою програмою “Комп’ютерна інженерія та програмування” спеціальності 123 “Комп’ютерна інженерія” другого (магістерського) рівня вищої освіти. – Хмельницький: ХНУ, 2022. – 65 с.
5. C. Hu. An Introduction to Software Design: Concepts, Principles, Methodologies, and Techniques 1st ed. 2023 Edition
6. M. Chemuturi. Software Design: A Comprehensive Guide to Software Development Projects 1st Edition, 2018.
7. N. Ford, M. Richards, P. Sadalage, Z. Dehghani. Software Architecture: The Hard Parts. Modern Trade-Off Analyses for Distributed Architectures. 1st Ed., 2022.
8. C. Fournier. The Manager's Path. O'Reilly Media, Inc., 2019.
9. M. Keeling. Design It!: From Programmer to Software Architect. 1st Ed. 2020.
10. D. Jakson. The Essence of Software: Why Concepts Matter for Great Design. 2023.
11. G. Marten Bonnema, K. Th. Veenvliet, J.F. Broenink. Systems Design and Engineering: Facilitating Multidisciplinary Development Projects. 2022.
12. L. Bas, P. Clements, R. Kazman. Software Architecture in Practice (SEI Series in Software Engineering) 4th Edition. 2021.
13. A. Xu. System Design Interview – An insider's guide. 2020.
14. ISO/IEC 25000 series of standards. Web-site. URL: <http://iso25000.com/index.php/en/iso-25000-standards> (Last accessed: December 7, 2017).
15. Why the Waterfall Model Doesn't Work // <http://www.infoq.com/resource/articles/scaling-software-agility/en/resources/ch02.pdf>

8. ІНФОРМАЦІЙНІ РЕСУРСИ

Електронний університет:

1. Модульне середовище для навчання (розміщені усі необхідні матеріали з дисципліни, в тому числі тестові завдання для поточного та семестрового контролю знань).
2. Електронна бібліотека університету