Хмельницький національний університет

ЗАТВЕРДЖУЮ
Декан факультету інформаційних технологій
_____ проф. Тетяна ГОВОРУЩЕНКО
«05» Вересня 2024 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Технології створення програмних продуктів

Галузь знань – 12 Інформаційні технології
Спеціальність – 122 Комп'ютерні науки
Рівень вищої освіти – Перший (бакалаврський)
Освітньо-професійна програма – Комп'ютерні науки
Обсяг дисципліни – 6 кредитів ЄКТС
Шифр дисципліни – ОПП.09
Мова навчання – українська
Статус дисципліни – обов'язкова (цикл професійної підготовки)
Факультет - Інформаційних технологій
Кафедра – Комп'ютерних наук

Форма навчання	Курс	Семестр	Обсяг дисципліни		Кількість годин						Курсовий проект	Курсова робота	Форма семестрового контролю	
					Аудиторні заняття				Індивідуальна Робота студента	Самостійна робота студентів				
			Кредити ЄКТС	Години	Всього	Лекції	Лабораторні роботи	Практичні заняття					Залік	Іспит
Д	4	7	6	180	85	34	51			95				+
Разом			**6**	**180**	**85**	**34**	**51**			**95**				**1**

Робоча програма складена на основі освітньо-професійної програми «Комп'ютерні науки» та стандарту вищої освіти зі спеціальності 122 – Комп'ютерні науки першого (бакалаврського) рівня вищої освіти.

Робоча програма складена _____ *д.т.н., доц. Едуард МАНЗЮК*

Схвалена на засіданні кафедри *комп'ютерних наук*

Протокол від *30 серпня 2024 р.* № *1* . Зав. кафедри _____ *проф. Олександр БАРМАК*

Робоча програма розглянута та схвалена вченою радою факультету інформаційних технологій

Голова вченої ради факультету _____ *проф. Тетяна ГОВОРУЩЕНКО*

Хмельницький 2024

ТЕХНОЛОГІЇ СТВОРЕННЯ ПРОГРАМНИХ ПРОДУКТІВ

Тип дисципліни	Обов'язкова професійної підготовки
Освітній рівень	Перший (бакалаврський)
Мова викладання	Українська
Семестр	7
Кількість встановлених кредитів ЄКТС	6,0
Форми навчання, для яких викладається дисципліна	Очна денна

Результати навчання. Студент, який успішно завершив вивчення дисципліни, повинен: вміло *використовувати* понятійний апарат; уміти *виконувати* розробку програмних систем із застосуванням шаблонних підходів; *визначати* необхідність застосування шаблонів проектування; *характеризувати* шаблони проектування; *підбирати* ефективні методи проектування програмних систем; *проектувати* програмні системи із використанням шаблонів проектування; *виконувати* розробку програмних систем.

Зміст навчальної дисципліни. Шаблони проектування. Класифікація шаблонів проектування, (шаблони GoF). Породжуючі шаблони. Поведінкові шаблони. Структурні шаблони. Шаблони Abstract Factory, Builder, Factory Method, Prototype, Singleton, Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Visitor, Template Method, Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy. Cache Management, Proxy, Object Pool, Dynamic Linkage,Single Threated Execution, Lock Object, Guarder Suspension, Balking, Scheduler, Read/Write Lock, Producer-Consumer, Two-Phase Termination, Double Buffering, Asynchronous Processing, Future. Принципи GRASP, Шаблони Low Coupling, High Cohesion, Information Expert, Creator, Pure Fabrication, Indirection, Polymorphism, Protected Variations, Controller, Domain Model, Service Layer, Table Modyle, Transaction Script, шаблони джерела даних Active Record, Data Mapper, Row Data Gateway, Table Data Gateway.

Пререквізити – вебтехнології, технології розподілених систем та паралельних обчислень, об'єктно-орієнтоване проектування, алгоритмізація та програмування, теорія алгоритмів.

Кореквізити – управління ІТ-проєктами, інформаційні технології хмарних обчислень.

Запланована навчальна діяльність: лекцій 34 год., лабораторних робіт 51 год., самостійної роботи 95 год.; разом 180 год.

Форми (методи) навчання: словесні (лекція, розповідь, дискусія, пояснення), практичні (лабораторна робота, індивідуальні завдання), наочні методи (ілюстрування зразків, демонстрування слайдів)

Форми оцінювання результатів навчання: усне опитування, захист лабораторних робіт.

Вид семестрового контролю: іспит.

Навчальні ресурси:

1. Wengner M._Practical Design Patterns for Java Developers. – Packt Publishing, 2023. – 266p.
2. Jain H., Semwal V. Design Patterns using Java Paperback. – Hemant Jain, 2023. – 283p.
3. Bass J.M. Agile Software Engineering Skills. – Springer International Publishing, 2023 – 317p.
4. Khononov V. Learning Domain-Driven Design. – O'Reilly Media, 2021 – 342p
5. Манзюк Е.А., Бармак О.В., Радюк П.М., Молчанова М.О., Пасічник О.А., Багрій Р.О. Технології створення програмних продуктів. Теоретичний курс та лабораторний практикум: навч. Посібник. Хмельницький: ХНУ, 2024р. 306с.
6. Модульне середовище для навчання MOODLE. Доступ до ресурсу: https://msn.khmnu.edu.ua
7. Електронна бібліотека університету. Доступ до ресурсу: http://library.khmnu.edu.ua

Викладачі: д.т.н., доцент Едуард МАНЗЮК.

Пояснювальна записка

Дисципліна «Технології створення програмних продуктів» є однією із фахових дисциплін і займає провідне місце у підготовці фахівців освітнього рівня «бакалавр» за спеціальністю та освітньо-професійною програмою «Комп'ютерні науки».

компетентності:

ФК 8. Здатність проєктувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об’єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

програмні результати навчання:

ПРН 9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв’язання задач в галузі комп’ютерних наук.

Пререквізити – вебтехнології, технології розподілених систем та паралельних обчислень, об'єктно-орієнтоване проектування, алгоритмізація та програмування, теорія алгоритмів.

Кореквізити – управління ІТ-проєктами, інформаційні технології хмарних обчислень.

Мета дисципліни. Вивчення технології програмування та створення програмних продуктів. Основна увага приділяється технології проектування програмного забезпечення, а також основним шаблонам, які застосовуються при розробці програмних систем.

Предмет дисципліни. Шаблони проектування програмних систем, принципи розробки програмних систем, архітектурні рішення побудови програмних продуктів, методи побудови програмних систем.

Завдання дисципліни. Надати студентам знання і практичні навички з проектування програмних систем із засуванням шаблонних методів та підходів щодо розробки архітектури та структури.

Структура залікових кредитів дисципліни

Назва теми	Кількість годин, відведених на:		
	Денна форма		
	лекції	лабор. роботи	СРС
Сьомий семестр			
Тема 1. Шаблони проектування. Вступ в шаблони проектування. Класифікація шаблонів проектування, (шаблони GoF). Породжуючі шаблони Abstract Factory, Builder.	2	8	6
Тема 2. Породжуючі шаблони Factory Method, Prototype, Singleton.	2	6	6
Тема 3. Поведінкові шаблони проектування Chain of Responsibility, Command, Interpreter.	2	6	6
Тема 4. Поведінкові шаблони Iterator, Mediator, Memento.	2	6	6
Тема 5. Поведінкові шаблони Observer, State, Strategy.	2	6	6
Тема 6. Поведінкові шаблони Visitor, Template Method, структурний шаблон Adapter.	2	6	6
Тема 7. Структурні шаблони Bridge, Composite, Decorator.-	2	6	6
Тема 8. Структурні шаблони Facade, Flyweight, Virtual Proxy.	2	7	6
Тема 9. Структурні шаблони Cache Management, Proxy, породжуючий шаблон Object Pool.	2	-	6
Тема 10. Структурний шаблон Dynamic Linkage, шаблони для конкуруючих операцій Single Threated Execution, Lock Object.	2	-	6
Тема 11. Шаблони для конкуруючих операцій Guarder Suspension, Balking, Scheduler.	2	-	6
Тема 12. Шаблони для конкуруючих операцій Read/Write Lock, Producer-Consumer, Two-Phase Termination.	2	-	6
Тема 13. Шаблони для конкуруючих операцій Double Buffering, Asynchronous Processing, Future	2	-	6
Тема 14. Проектування на базі обов'язків (Responsibility-Driven Design). Принципи GRASP. Шаблони Low Coupling, High Cohesion, Information Expert.	2	-	6
Тема 15. Шаблони Creator, Pure Fabrication, Indirection, Polymorphism, Protected Variations, Controller.	2	-	6
Тема 16. Шаблони домена Domain Model, Service Layer, Table Module, Transaction Script, шаблони джерела даних Active Record, Data Mapper	4	-	5
Разом:	**34**	**51**	**95**

Програма навчальної дисципліни

Зміст лекційного матеріалу

№ п/п	Перелік змістових модулів, тем лекцій, їх анотації	Годин
1	Породжуючі шаблони Abstract Factory, Builder.	2

	Літ.: [1] с. 65 – 100; [2] с. 24 – 65; [5] с. 9-23.	
2	Породжуючі шаблони Factory Method, Prototype, Singleton. Літ.: [1] с. 65 - 100 ; [2] с. 24 – 65; [5] с. 23-37.	2
3	Поведінкові шаблони проектування Chain of Responsibility, Command, Interpreter. Літ.: [1] с. 139 – 176; [2] с. 113 – 192; [5] с. 38-63.	2
4	Поведінкові шаблони Iterator, Mediator, Memento. Літ.: [1] с. 139 – 176; [2] с. 113 – 192; [5] с. 63-72.	2
5	Поведінкові шаблони Observer, State, Strategy. Літ.: [1] с. 139 – 176; [2] с. 113 – 192; [5] с. 72-83.	2
6	Поведінкові шаблони Visitor, Template Method. Структурний шаблон Adapter. Літ.: [1] с. 139 – 176; [2] с. 113 – 192; [5] с. 83-94.	2
7	Структурні шаблони Bridge, Composite, Decorator. Літ.: [1] с. 101 – 138; [2] с. 65 – 104; [5] с. 94-102.	2
8	Структурні шаблони Facade, Flyweight, Proxy. Літ.: [1] с. 101 – 138; [2] с. 65 – 104; [5] с. 102-114.	2
9	Структурні шаблони Cache Management, Proxy, породжуючий шаблон Object Pool Літ.: [1] с. 101 – 138; [2] с. 65 – 104; [5] с. 121-141.	2
10	Структурний шаблон Dynamic Linkage, шаблони для конкуруючих операцій Single Threated Execution, Lock Object Літ.: [1] с. 101 – 138; [2] с. 65 – 104; [5] с. 141-160.	2
11	Шаблони для конкуруючих операцій Guarder Suspension, Balking, Scheduler Літ.: [1] с. 179 – 213; [5] с. 160-175.	2
12	Шаблони для конкуруючих операцій Read/Write Lock, Producer-Consumer, Two-Phase Termination Літ.: [1] с. 179 – 213; [5] с. 175-185.	2
13	Шаблони для конкуруючих операцій Double Buffering, Asynchronous Processing, Future Літ.: [1] с. 179 – 213; [5] с. 185-202.	2
14	Проектування на базі обов'язків (Responsibility-Driven Design). Шаблони Low Coupling, High Cohesion, Information Expert Літ.: [3] с. 111 – 128; [5] с. 204-218.	2
15	Шаблони Creator, Pure Fabrication, Indirection, Polymorphism, Protected Variations, Controller. Літ.: [3] с. 111 – 128; [5] с. 218-236.	2
16	Шаблони домена Domain Model, Service Layer, Table Module, Transaction Script, шаблони джерела даних Active Record, Data Mapper Літ.: [4] с. 171 – 185; [5] с. 238-258.	4
	Разом:	**34**

Зміст лабораторних занять

№ п/п	Тема лабораторного заняття	Годин
1	Практична реалізація шаблонів Abstract Factory, Builder. Літ.: [1] с. 71 – 78; [2] с. 24 – 38; [5] с. 264-267.	8
2	Практична реалізація шаблонів Factory Method, Prototype, Singleton. Літ.: [1] с. 67 – 71, 78 – 85; [2] с. 38 – 57; [5] с. 267-271.	6
3	Практична реалізація шаблонів Chain of Responsibility, Command, Interpreter. Літ.: [1] с. 143 – 150; [2] с. 114 – 142; [5] с. 271-276.	6

4	Практична реалізація шаблонів Iterator, Mediator, Memento. Літ.: [1] с. 150 – 160; [2] с. 142 – 162; [5] с. 276-280.	6
5	Практична реалізація шаблонів Observer, State, Strategy. Літ.: [1] с. 160 – 169; [2] с. 162 – 179; [5] с. 280-284.	6
6	Практична реалізація шаблонів Visitor, Template Method, Adapter. Літ.: [1] с. 169 – 176, 102 – 106; [2] с. 66 – 72, 186 – 202; [5] с. 284-287.	6
7	Практична реалізація шаблонів Bridge, Composite, Decorator. Літ.: [1] с. 106 – 115; [2] с. 72 – 94; [5] с. 287-291.	6
8	Практична реалізація шаблонів Facade, Flyweight, Proxy. Літ.: [1] с. 115 – 135; [2] с. 94 – 113; [5] с. 291-294.	7
	Разом:	**51**

Зміст самостійної (у т.ч. 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, виконанні індивідуальних завдань тощо.

Тема	Сам. роб.
Тема 1. Шаблони проектування. Вступ в шаблони проектування. Класифікація шаблонів проектування, (шаблони GoF). Породжуючі шаблони Abstract Factory, Builder.	6
Тема 2. Породжуючі шаблони Factory Method, Prototype, Singleton.	6
Тема 3. Поведінкові шаблони проектування Chain of Responsibility, Command, Interpreter.	6
Тема 4. Поведінкові шаблони Iterator, Mediator, Memento.	6
Тема 5. Поведінкові шаблони Observer, State, Strategy.	6
Тема 6. Поведінкові шаблони Visitor, Template Method, структурний шаблон Adapter.	6
Тема 7. Структурні шаблони Bridge, Composite, Decorator.	6
Тема 8. Структурні шаблони Facade, Flyweight, Virtual Proxy.	6
Тема 9. Структурні шаблони Cache Management, Proxy, породжуючий шаблон Object Pool	6
Тема 10. Структурний шаблон Dynamic Linkage, шаблони для конкуруючих операцій Single Threated Execution, Lock Object.	6
Тема 11. Шаблони для конкуруючих операцій Guarder Suspension, Balking, Scheduler.	6
Тема 12. Шаблони для конкуруючих операцій Read/Write Lock, Producer-Consumer, Two-Phase Termination.	6
Тема 13. Шаблони для конкуруючих операцій Double Buffering, Asynchronous Processing, Future	6
Тема 14. Проектування на базі обов’язків (Responsibility-Driven Design). Принципи GRASP. Шаблони Low Coupling, High Cohesion, Information Expert.	6
Тема 15. Шаблони Creator, Pure Fabrication, Indirection, Polymorphism, Protected Variations, Controller.	6
Тема 16. Шаблони домена Domain Model, Service Layer, Table Module, Transaction Script, шаблони джерела даних Active Record, Data Mapper.	5
Разом	95

Керівництво самостійною роботою та контроль за виконанням індивідуального завдання здійснює викладач згідно з розкладом консультацій у позаурочний час.

Технології навчання

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних методів. Зокрема, лекції проводяться в основному словесними методами, візуалізації, а лабораторні заняття проводяться з використанням інформаційних технологій і мають за мету – набуття студентами практичних навичок з проектування програмних систем, які базуються на шаблонах, самостійна робота.

Методи контролю

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. При цьому при виведенні остаточної оцінки враховуються результати поточного контролю.

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 використовувати їх при вирішенні різних питань при проектуванні інформаційних систем, вміє розширити їх.

Перед вивченням дисципліни, як правило, проводиться вхідний контроль знань з дисциплін, що їй передують і забезпечують.

Використовуються такі методи поточного контролю:

– усне опитування перед допуском до лабораторного заняття;

– захист лабораторних робіт;

– проміжний контроль теоретичного матеріалу з теми.

При отримані підсумкової семестрової оцінки враховуються результати як поточного контролю так і захисту лабораторних робіт.

Оцінювання результатів навчання студентів у семестрі

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. Підсумкова оцінка з дисципліни виставляється як середньоарифметична за усі семестри. 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 та індивідуального завдання 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 програмного коду; вміння студента обґрунтувати прийняті рішення; своєчасний захист лабораторної 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.

Примітка: для практичних чи семінарських занять викладач встановлює обов’язковий мінімум оцінок, які має отримати студент впродовж семестру, щоб виконати програму дисципліни.

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 і

конструктивні рішення. Відмінна оцінка передбачає грамотний, логічний виклад відповіді (як в усній, так і в письмовій формі), якісне зовнішнє оформлення. Студент повинен набути практичних навичок з розробки різних видів програмних систем.

Оцінка "відмінно" виставляється студенту, який глибоко засвоїв методи та підходи щодо розробки програмних систем із застосуванням шаблонів. 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 в 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 має будуватись на основі самостійного мислення.

Оцінку „добре" отримує студент за правильну відповідь з двома-трьома 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 практичні навички з проектування програмних систем, але допустив неточності, не має чіткого поняття необхідність та доцільність використання шаблонів. 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 з розроблення програмних систем.

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 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.

Окремі теми курсу можуть бути зараховані у випадку отримання студентом **результатів навчання у неформальній освіті**, що підтверджені відповідним документом (сертифікат, свідоцтво, освітня програма тощо) та відповідно до Положення про порядок визнання та перезарахування результатів навчання здобувачів вищої освіти у ХНУ (https://khmnu.edu.ua/wp-content/uploads/normatyvni-dokumenty/polozhennya/pro-poryadok-vyznannya-ta-perezarahuvannya-rezultativ-navchannya.pdf). Перелік лабораторних робіт, що можуть бути перезараховані та приклади відповідних курсів на освітніх ресурсах:

Лабораторні роботи 1-3: Design Patterns – https://www.coursera.org/learn/design-patterns

Лабораторні роботи 1-6: Software Design and Architecture – https://www.coursera.org/specializations/software-design-architecture

Лабораторні роботи 1-3: Software Design Patterns: Best Practices for Software Developer– https://www.educative.io/courses/software-design-patterns-best-practices

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

<table>
<tr><td colspan="8">Аудиторна робота</td><td>Семестровий контроль
(іспит)</td></tr>
<tr><td colspan="9">7 семестр</td></tr>
<tr><td colspan="8">Лабораторні роботи №:</td><td>Підсумковий контрольний захід</td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td></td></tr>
<tr><td colspan="8">ВК: 0,6</td><td>0,4</td></tr>
</table>

Умовні позначення: ВК – ваговий коефіцієнт.

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

Питання для самоконтролю студентів

1. Для чого використовуються GoF шаблони?
2. Де можна використовувати породжуючі шаблони?
3. Що таке шаблон Abstract Factory і як його використовують?
4. Для чого використовують шаблон Builder?
5. Яке призначення шаблону Factory Method і де його можна застосовувати?
6. Що таке шаблон Prototype і як він використовується?
7. Як працює шаблон Singleton і де його можна застосовувати?
8. Для чого потрібний шаблон Chain of Responsibility і де його використовують?
9. Що таке шаблон Command і в яких сценаріях його варто застосовувати?
10. Який зміст шаблону Interpreter і де його можна застосовувати?
11. Де можна застосовувати поведінкові шаблони?
12. Як використовується шаблон Iterator і де його можна застосувати?
13. Для чого використовується шаблон Mediator і в яких областях його можна використовувати?
14. Що таке шаблон Memento і як його застосовують?
15. Де застосовують шаблон Observer і як він працює?
16. Яким чином працює шаблон State і де його можна використовувати?
17. Для чого використовується шаблон Strategy і в яких сферах його можна використовувати?
18. Що таке шаблон Visitor і як він використовується?
19. Яким чином використовується шаблон Template Method і де його можна застосовувати?
20. Де можна застосовувати структурні шаблони?
21. Що таке шаблон Adapter і в яких сценаріях його варто використовувати?
22. Для чого потрібний шаблон Bridge і в яких областях його використовують?
23. Як використовується шаблон Composite і де його можна застосовувати?
24. Що таке шаблон Decorator і в яких сферах його використовують?
25. Для чого використовується шаблон Facade і де його можна використовувати?
26. Як працює шаблон Flyweight і в яких сферах його можна застосовувати?

27. Для чого використовується шаблон Proxy і в яких сценаріях його варто використовувати?

28. Як використовується діаграма класів у шаблонах проектування?

29. Як використовується діаграма кооперації у шаблонах проектування?

30. Як використовується діаграма послідовності у шаблонах проектування?

31. Що означає шаблон Cache Management і в яких сферах його використовують?

32. Як використовується шаблон Virtual Proxy і в яких сценаріях його варто використовувати?

33. Для чого використовується шаблон Object Pool і де його можна застосовувати?

34. Що означає шаблон Dynamic Linkage і в яких сферах його використовують?

35. Для чого використовується шаблон Single Threated Execution і в яких сценаріях його варто використовувати?

36. Що таке шаблон Lock Object і в яких сферах його використовують?

37. Як використовується шаблон Guarder Suspension і в яких сценаріях його варто використовувати?

38. Для чого використовується шаблон Balking і в яких областях його можна застосовувати?

39. Що означає шаблон Scheduler і в яких сферах його використовують?

40. Як використовується шаблон Read/Write Lock і в яких сценаріях його варто використовувати?

41. Для чого використовується шаблон Producer-Consumer і в яких сферах його використовують?

42. Що означає шаблон Two-Phase Termination і в яких сценаріях його варто використовувати?

43. Як використовується шаблон Double Buffering і в яких сферах його використовують?

44. Для чого використовується шаблон Asynchronous Processing і в яких областях його можна застосовувати?

45. Що таке шаблон Future і як він використовується?

46. Для чого потрібен шаблон Low Coupling і в яких сферах його використовують?

47. Як використовується шаблон High Cohesion і в яких сценаріях його варто використовувати?

48. Для чого використовується шаблон Information Expert і в яких областях його можна застосовувати?

49. Що означає шаблон Creator і в яких сферах його використовують?

50. Для чого використовується шаблон Pure Fabrication і в яких сценаріях його варто використовувати?

51. Що означає шаблон Indirection і в яких сферах його використовують?

52. Як використовується шаблон Polymorphism і в яких областях його можна застосовувати?

53. Для чого використовується шаблон Protected Variations і в яких сценаріях його варто використовувати?

54. Що таке шаблон Controller і в яких сферах його використовують?

55. Для чого потрібен шаблон Domain Model і в яких областях його використовують?

56. Як використовується шаблон Service Layer і в яких сценаріях його варто використовувати?

57. Що означає шаблон Table Module і в яких сферах його використовують?

58. Для чого використовується шаблон Transaction Script і в яких сценаріях його варто використовувати?

59. Як використовується шаблон Active Record і в яких областях його використовують?

60. Для чого використовується шаблон Data Mapper і в яких сценаріях його варто використовувати?

Методичне забезпечення

Навчальний процес з дисципліни «Технології створення програмних продуктів» повністю і

в достатній кількості забезпечений необхідною навчально-методичною літературою. Зокрема, викладачами кафедри підготовлені і видані такі роботи:

1. Електронний освітній ресурс "Дистанційний курс" №1106 " Технологія створення програмних продуктів"/ Манзюк Е.А., кафедра інформаційних технологій проектування/ дата розміщення в інформаційній підсистемі Хмельницького національного університету "Дистанційне навчання" затверджено рішенням Науково-методичної ради університету протокол №6 від 21 червня 2014 р., сертифікат № 914.
2. Манзюк Е.А., Бармак О.В., Радюк П.М., Молчанова М.О., Пасічник О.А., Багрій Р.О. Технології створення програмних продуктів. Теоретичний курс та лабораторний практикум: навч. Посібник. Хмельницький: ХНУ, 2024р. 306с.

Рекомендована література

1. Wengner M._Practical Design Patterns for Java Developers. – Packt Publishing, 2023. – 266p.
2. Jain H., Semwal V. Design Patterns using Java Paperback. – Hemant Jain, 2023. – 289p.
3. Bass J.M. Agile Software Engineering Skills. – Springer International Publishing, 2023 – 317p.
4. Khononov V. Learning Domain-Driven Design. – O'Reilly Media, 2021 – 342p
5. Манзюк Е.А., Бармак О.В., Радюк П.М., Молчанова М.О., Пасічник О.А., Багрій Р.О. Технології створення програмних продуктів. Теоретичний курс та лабораторний практикум: навч. Посібник. Хмельницький: ХНУ, 2024р. 306с.

Інформаційні ресурси

1. Модульне середовище для навчання. Доступ до ресурсу: https://msn.khnu.km.ua.
2. Модульний курс для дистанційної форми навчання Доступ до ресурсу: http://dn.khnu.km.ua/dn/k_list.aspx?bk=T.
3. Електронна бібліотека університету. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/p1age_lib.php.
4. Репозитарій ХНУ. Доступ до ресурсу: http://elar.khnu.km.ua/jspui/?locale=uk.