

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

1. Загальна інформація про навчальну дисципліну

Повна назва навчальної дисципліни	Технологія створення програмних продуктів
Повна офіційна назва закладу вищої освіти	Сумський державний університет
Повна назва структурного підрозділу	Факультет електроніки та інформаційних технологій. Кафедра інформаційних технологій
Розробник(и)	Нагорний Володимир В`ячеславович
Рівень вищої освіти	Перший рівень вищої освіти, НРК – 6 рівень, QF-LLL – 6 рівень, FQ-EHEA – перший цикл
Семестр вивчення навчальної дисципліни	16 тижнів протягом 4-го семестру
Обсяг навчальної дисципліни	Обсяг становить 5 кред. ЄКТС, 150 год. Для денної форми навчання 56 год. становить контактна робота з викладачем (24 год. лекцій, 32 год. лабораторних занять), 94 год. становить самостійна робота.
Мова викладання	Українська

2. Місце навчальної дисципліни в освітній програмі

Статус дисципліни	Обов`язкова навчальна дисципліна для всіх освітніх програм спеціальності 122 "Комп`ютерні науки"
Передумови для вивчення дисципліни	Вивчення курсу "Об'єктно-орієнтоване програмування"
Додаткові умови	Додаткові умови відсутні
Обмеження	Обмеження відсутні

3. Мета навчальної дисципліни

Метою навчальної дисципліни є досягнення студентами поглибленого розуміння процесів, методів та інструментів, необхідних для успішного створення високоякісних програмних продуктів, розвиток навичок аналізу, проектування, реалізації та тестування програм з використанням сучасних технологій розробки, а також на засвоєння принципів командної роботи, ефективного управління проектами та розуміння важливості дотримання стандартів програмування та кращих практик.

4. Зміст навчальної дисципліни

Тема 1 Вступ у технології програмування. Життєвий цикл програмного забезпечення

Розгляд фундаментальних аспектів програмування, від основних понять до ключових концепцій, що лежать в основі розробки ПЗ. Дослідження різноманітних категорій ПЗ, включаючи системне, веб-додатки, мобільні додатки та ігри. Опис програмування як постійного процесу створення, тестування та підтримки програмного забезпечення. Аналіз еволюції мов програмування, від старих до сучасних, з фокусом на їхню роль у розробці. Розгляд різних підходів та парадигм програмування, включаючи об'єктно-орієнтоване та функціональне програмування. Дослідження історії та основних характеристик перших мов програмування. Опис етапів розробки ПЗ, від аналізу вимог до підтримки та обслуговування. Порівняння різних методологій управління проектами та розробкою ПЗ, включаючи Agile, Waterfall та DevOps.

Тема 2 Планування та керування

Аналіз потреб користувачів та визначення функціональних та нефункціональних вимог для ефективної розробки програмного продукту. Створення документації, включаючи схеми архітектури, базу даних, інтерфейси та плани реалізації. Огляд основних форм планування робіт, структурування завдань, ресурсного планування та розподіл ролей в команді. Визначення критеріїв якості, тестування, аудит коду та оцінка відповідності стандартам якості. Розгляд Agile, Waterfall, DevOps тощо, для вибору оптимального методу відповідно до проекту та командних можливостей.

Тема 3 Проектування програмного забезпечення

Використання методологій IDEF0, ARIS та BPMN для створення візуальних моделей бізнес-процесів для оптимізації діяльності підприємства та підвищення його ефективності. Аналіз структури мови UML для розробки діаграм, що спрощують процес розробки програмних продуктів та підвищують їхню якість. Розгляд принципів створення користувацького інтерфейсу, зокрема UX-дизайну та UI-розробки, для забезпечення зручності та задоволення користувачів програмних продуктів.

Тема 4 Формалізовані методології розробки програмних продуктів

Використання формалізованих методологій для систематизації процесів розробки ПЗ, зменшення помилок та підвищення продуктивності команди. Уніфікований процес розроблення ПЗ - Rational Unified Process (RUP): гнучкість у виборі підходу до проекту та підтримка усього життєвого циклу ПЗ. Стандарт СММІ на зрілість процесів розробки ПЗ для сприяння покращення якості продукту та управлінню проектом. Огляд дисципліни керування ризиками для передбачення та управління негативними впливами на проект та його успішне завершення. Огляд дисципліни керування проектом для забезпечення ефективного планування, виконання та контролю проектних завдань.

Тема 5 Гнучкі методології розробки програмних продуктів

Гнучка розробка (Agile software development), як підхід до розробки програм, який розвивається на основі співпраці, ітерацій та швидкої адаптації до змін. Методи та фреймворки Agile, такі як Scrum, Kanban, Lean, щоб покращити процес розробки. Стадії управління гнучкою розробкою - планування, розробка, тестування, виправлення та постійне вдосконалення. Метод Extreme Programming - методологія розробки програм, яка акцентує на спільній роботі команди та тестуванні. Практики у Extreme Programming: парне програмування, тестування першого, постійна інтеграція та інші. Scrum, як фреймворк для управління проектами, який використовує ітерації та регулярні зустрічі для вдосконалення процесу. Процес автоматичного збирання, тестування та впровадження програмного продукту для швидкого розгортання.

Тема 6 Стандартизація ІТ-галузі

Галузеві стандарти. IEEE. SWEBOOK. GDPR. ДСТУ. CMM. Tickit. ISO. Класифікація стандартів розроблення програмних систем (Systems and software engineering). Організації- розробники міжнародних стандартів. Державні стандарти України. Стандарт СММІ на зрілість процесів розробки ПЗ. Бібліотека інфраструктури інформаційних технологій (ITIL). Регламент захисту персональних даних (GDPR).

Тема 7 Атестація та верифікація

Верифікація, як процес перевірки відповідності програмного продукту вимогам і специфікаціям із включенням в себе аналізу коду та документації. Атестація/валідація, як процес перевірки коректності роботи ПЗ в реальному середовищі, перевірка на відповідність специфікації, перевірка задоволення потреб користувачів. Підходи до забезпечення стійкості та відсутності помилок у програмному продукті протягом усього життєвого циклу. Постійний процес виправлення помилок та підвищення надійності ПЗ. Аналіз результатів верифікації та валідації для забезпечення якості продукту. Види тестування: функціональне, навантажувальне, безпеки, тестування сумісності та інші види тестування для виявлення помилок. Test-Driven Development (TDD), як метод розробки: спершу тести, а потім функціональність для їхнього пройдення.

5. Очікувані результати навчання навчальної дисципліни

Після успішного вивчення навчальної дисципліни здобувач вищої освіти зможе:

PH1	Створювати програмні моделі, які відображають реальні предметні об'єкти та їх взаємодію з використанням відповідних інструментів та методологій.
PH2	Вміти аналізувати, планувати та виконувати всі стадії життєвого циклу програмного продукту, включаючи збір та аналіз вимог, проектування, розробку, тестування, впровадження та підтримку, а також створення необхідної проектної документації.
PH3	Використовувати CASE-інструменти для аналізу, проектування і управління програмними проектами, визначати взаємозв'язки між компонентами системи і розробляти програмні рішення, що відповідають вимогам цих систем, таким чином підвищуючи ефективність розробки.

6. Роль навчальної дисципліни у досягненні програмних результатів

Програмні результати навчання, досягнення яких забезпечує навчальна дисципліна.

Для спеціальності 122 Комп'ютерні науки:

ПР9	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.
ПР11	Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).
ПР15	Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

7. Роль освітнього компонента у формуванні соціальних навичок

Загальні компетентності та соціальні навички, формування яких забезпечує навчальна дисципліна:

СН1	Здатність вчитися і оволодівати сучасними знаннями.
СН2	Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
СН3	Здатність застосовувати знання у практичних ситуаціях.
СН4	Здатність приймати обґрунтовані рішення
СН5	Здатність працювати в команді.

8. Види навчальних занять

Тема 1. Вступ у технології програмування. Життєвий цикл програмного забезпечення	
Лк1 "Життєвий цикл та моделі життєвого циклу програмного забезпечення"	Вступ у технології програмування. Базові поняття. Види програмного забезпечення. Інтегроване середовище розроблення програмного забезпечення. Поняття технології програмування як процесу. Розвиток мов. Стилї та технологій програмування. Ранні мови програмування. Імперативне програмування (Imperative programming). Декларативне програмування (Declarative programming). Компонентне програмування (Component-based programming). Життєвий цикл програмного забезпечення. Моделі життєвого циклу ПЗ. Каскадна модель ЖЦ (Waterfall model). Ітеративна модель (Iterative model). Еволюційна модель. V-подібна модель. Адаптивний ЖЦ (Adaptive Life Cycle).
Лб1 "Планування роботи над проектом у формі діаграми Ганта"	Визначення переліку робіт над проектом із створення програмного продукту у вигляді ієрархічної структури. побудова діаграми Ганта.
Тема 2. Планування та керування	

<p>Лк2 "Визначення вимог та побудова планів" (денна)</p> <p>Постановка завдання на розроблення ПЗ. Проект із розроблення ПЗ. Завдання (Tasks). Фази (Summary tasks). Тривалість (Duration) і трудовитрати (Work). Залежності (Dependencies) та зв'язки (Links). Ролі (Roles) і ресурси (Resources). Призначення (Assignments). Основні форми планів робіт. План робіт у формі мережевого графіку. План робіт у формі діаграми Ганта.</p>
<p>Лк3 "Керування та забезпечення якості" (денна)</p> <p>Забезпечення якості ПЗ. Принципи вибору методології розроблення. Ролі (Roles). Уміння (Skills). Групи (Teams). Інструменти (Tools). Техніки (Techniques). Види діяльності (Activities) . Продукти (Work Products). Стандарти (Standards). Якість (Quality). Цінності (Values).</p>
<p>Лб2 "Планування робіт у формі мережевого графіку" (денна)</p> <p>Побудова мережевого графіку із подіями у вузлах та із роботами у вузлах.</p>
<p>Тема 3. Проектування програмного забезпечення</p>
<p>Лк4 "Візуальне моделювання бізнес-процесів" (денна)</p> <p>Контекстна діаграма. декомпозиція. переваги і недоліки нотації. CASE-засоби. ARIS (ARchitecture of Integrated Information Systems). BPMN. структура моделі. дії. події. з'єднання. артефакти.</p>
<p>Лк5 "Візуальне моделювання інформаційних систем" (денна)</p> <p>Візуальні моделі. використання UML у проектах. Unified Modeling Language (UML). принципи моделювання в UML. базові визначення. опис поведінки класифікатора. опис структури класифікатора. групи класифікаторів. метамодель класифікатора. сутності. структурні сутності. поведінкові сутності. структурні діаграми. діаграми поведінки. діаграма варіантів використання (Use case diagram). актори. варіанти використання. діаграма діяльності (Activity diagram). діаграма автомата (State Machine diagram). діаграма послідовності (Sequence diagram).</p>
<p>Лк6 "Уніфікований процес розробки (RUP). Визначення вимог" (денна)</p> <p>Rational unified process (RUP). життєвий цикл ПЗ в RUP. ітеративна розробка ПЗ в RUP. життєвий цикл ПЗ в RUP. співробітник. вид діяльності. артефакт. модель ВВ. робочий процес визначення вимог. процес визначення вимог. ідентифікація акторів у визначенні вимог. ідентифікація ВВ у визначенні вимог. документування ВВ. інтерфейс (UI та UX). юзабіліті. підходи до проектування інтерфейсу. підходи до проектування інтерфейсу. етапи процесу створення UI. аналіз та розроблення ескізів у процесі проектування UI. CASE-засоби розробки ескізів інтерфейсу.</p>

<p>Лк7 "Уніфікований процес розробки. Аналіз і проектування" (денна)</p> <p>Модель аналізу. мета процесу аналізу вимог. робочий процес аналізу вимог. порядок розробки моделі аналізу. діяльність: аналіз архітектури. клас аналізу. граничний клас (Boundary class). клас сутності (Entity class). керуючий клас (Control class). аналіз реалізації ВВ. діаграма комунікації. діаграма класів аналізу. діаграма послідовності. мета процесу проектування. робочий процес проектування. властивості моделі проектування. порядок розробки моделі проектування. клас проектування. діаграми для проектування ВВ. діаграма огляду взаємодії. діаграма синхронізації. діаграма класів. модель розгортання. діаграма розгортання.</p>
<p>Лк8 "Уніфікований процес розробки. Реалізація та тестування" (денна)</p> <p>Модель реалізації (implementation model). склад моделі реалізації. план складання. модель тестування. склад моделі тестування. модель тестування та відповідальності. робочий процес тестування. тестовий приклад. загальна процедура тестування. дефект. база даних помилок (приклад структури). статуси помилок. важливість помилок. пріоритети помилок. види тестування. тестування в різних моделях ЖЦ.</p>
<p>Лб3 "Моделювання бізнес-процесів у нотаціях IDEF0 та BPMN" (денна)</p> <p>Проведення структурно-функціонального моделювання бізнес-процесів у нотації IDEF0 та їх опис у нотації BPMN.</p>
<p>Лб4 "Розробка специфікації вимог" (денна)</p> <p>Розроблення документу специфікації вимог до програмного продукту. визначення відмінностей між специфікацією вимог та технічним завданням на розробку програмного продукту.</p>
<p>Лб5 "Основи Rational Unified Process. Розроблення моделі варіантів використання програми"</p> <p>Визначення акторів та варіантів використання програмної системи. отримання навичок побудови діаграми варіантів використання.</p>
<p>Тема 4. Формалізовані методології розробки програмних продуктів</p>
<p>Лк9 "Формалізовані методології Microsoft Solutions Framework та RUP" (денна)</p> <p>Формалізовані методології, розроблені ІТ-компаніями. Уніфікований процес розроблення ПЗ - Rational Unified Process (RUP). Ітеративна розробка ПЗ в RUP. Робочий процес визначення вимог. Робочий процес аналіз вимог. Робочий процес проектування. Робочий процес реалізація. Робочий процес тестування. Стандарт СММІ на зрілість процесів розробки ПЗ. Capability Maturity Model Integration (СММІ). Рівні СММІ 2.0. MSF базові принципи. Модель проектної групи (Team Model). Рольові кластери. Дисципліна керування ризиками. Дисципліна керування проектом. Дисципліна керування підготовкою.</p>
<p>Лб6 "Основи Rational Unified Process. Розроблення моделі аналізу програмної системи" (денна)</p> <p>Основи Rational Unified Process. Розроблення моделі аналізу програмної системи</p>
<p>Тема 5. Гнучкі методології розробки програмних продуктів</p>

Лк10 "Гнучкі методології Extreme Programming та Agile" (денна)

Управління проектами і розробка ПЗ. Проектний трикутник/трикутник компромісів. Статут проекту (Project charter). Структура статуту. Розвиток методологій та методів. Гнучка розробка (Agile software development). Методи та фреймворки. Методологія гнучкої розробки (Agile software development). Agile Manifesto. Мета гнучкої розробки. Стадії управління гнучкою розробкою. Організаційна структура компанії-розробника. Функціональна структура. Проектна структура. Матрична структура. Слабка матриця. Збалансована матриця. Сильна матриця. Метод Extreme Programming. Практики у Extreme Programming. Ролі учасників проекту у Extreme Programming. Метод Scrum. Ролі учасників проекту. Артефакти Scrum. Зустрічі Scrum. Безперервна доставка (Continuous delivery). Основні процеси DevOps.

Лб7 "Основи Rational Unified Process. Розроблення моделі проектування" (денна)

Визначення класів проектування, в икористовуючи класи аналізу. Побудова діаграм класів мовою UML. Вивчення складових частин моделі проектування відповідно до вимог RUP. Визначення компонентів програми. Побудова діаграми компонентів мовою UML.

Лб8 "Основи Rational Unified Process. Розроблення моделі реалізації" (денна)

Вивчення складових частин моделі реалізації відповідно до вимог RUP.

Лб9 "Реалізація ПП. Початковий етап" (денна)

Створення та проведення початкового налаштування проекту. Реалізація розмежування прав доступу до функцій програми.

Лб10 "Реалізація ПП. Завантаження даних" (денна)

Реалізація ВВ Завантаження файлу із даними.

Тема 6. Стандартизація ІТ-галузі

Лк11 "Огляд основних вітчизняних та світових стандартів ІТ-галузі" (денна)

Галузеві стандарти. IEEE. SWEBOOK. GDPR. ДСТУ. CMM. Tickit. ISO. Класифікація стандартів розроблення програмних систем (Systems and software engineering). Організації- розробники міжнародних стандартів. ISO/IEC 12207:2017 Systems and software engineering – Software Life Cycle Processes. ISO/IEC 15288:2015 Systems and software engineering - System life cycle processes. ISO/IEC 15289:2017 Systems and software engineering - Content of life-cycle information items (documentation). ISO/IEC 19505:2012 Information technology – Object Management Group Unified Modeling Language (OMG UML). ISO/IEC 20000:2017 Information technology - Service management. Родина стандартів ISO/IEC 27000 - Information technology - Security techniques - Information security management systems (ISMS). Група стандартів ISO/IEC 29110 Systems and software engineering - Life cycle profiles for Very Small Entities. Група стандартів ISO/IEC 29119 Systems and software engineering - Software testing. ISO/IEC 29148:2018 Systems and software engineering - Life cycle processes - Requirements engineering. ISO/IEC 90003:2018 Software engineering -- Guidelines for the application of ISO 9001:2015 to computer software. Державні стандарти України. Стандарт СММІ на зрілість процесів розробки ПЗ. Бібліотека інфраструктури інформаційних технологій (ITIL). Регламент захисту персональних даних (GDPR).

Лб11 "Реалізація ПП. CRUD операції (Частина 1)" (денна) Реалізація ВВ редагування інформації.
Лб12 "Реалізація ПП. CRUD операції (Частина 2)" (денна) Реалізація ВВ додавання інформації та видалення інформації.
Лб13 "Реалізація ПП. Пошук даних" (денна) Реалізація ВВ Відбір інформації за критерієм X. Реалізація ВВ Відбір інформації за критерієм XY.
Лб14 "Реалізація ПП. Запис даних" (денна) Реалізація ВВ Запис відібраної інформації. Проведення уточнення інтерфейсу із використанням розмітки WPF.
Тема 7. Атестація та верифікація
Лк12 "Роль атестації та верифікації в забезпеченні надійності програмного забезпечення протягом усього життєвого циклу" (денна) Надійність (Reliability). U-видна крива (bathtub curve) розподілу помилок у жц обладнання. Верифікація (Verification). Атестація/валідація (Validation). Надійність ПЗ протягом ЖЦ. Характеристики надійності. Методики забезпечення надійності. Забезпечення надійності в життєвому циклі ПЗ. Прогнозування збоїв. Матриця джерело/класифікація збоїв. Попередження збоїв. Усунення помилок та забезпечення стійкості. Оцінювання в ході Verification & Validation (V&V). Планування V&V. V&V у життєвому циклі програми. Методи V&V. Переваги інспектування. Види тестування. Test-Driven Development (TDD). Планування верифікації та атестації.
Лб15 "Засоби рефакторингу коду та автоматизації тестування" (денна) Ознайомлення із поняттям рефакторингу та вивчення засобів його виконання в MS Visual Studio. Формування навичок використання наявних в MS Visual Studio інструментів тестування програм.
Лб16 "Формування інсталяційного пакету програми засобами MS Visual Studio" (денна) Створення інсталяційного пакету для розробленої програми.

9. Стратегія викладання та навчання

9.1 Методи викладання та навчання

Дисципліна передбачає навчання через:

МН1	Лекційне навчання
МН2	Практикоорієнтоване навчання
МН3	Самостійне навчання

Лекційне навчання використовується для пояснення теоретичних засад та методологій створення програмних продуктів (РН 1, РН 2). Лабораторні заняття допомагають студентам практично застосувати отримані знання. Це може включати роботу зі спеціалізованими

програмними інструментами для управління проектами, створення прототипів, моделювання архітектури, вивчати етапи життєвого циклу програмного продукту, від ідеї до реалізації та супроводу (РН 1, РН 2, РН 3). У деяких лабораторних заняттях можлива робота у групах для вирішення складних завдань (РН 1, РН 2). Це допомагає студентам навчитися співпрацювати, обмінюватися ідеями та ділитися відповідальністю. Проектне навчання дозволяє студентам застосовувати всі отримані знання та навички для реального створення програмного продукту від концепції до реалізації (РН 3). Самостійна робота студентів полягає в опрацюванні рекомендованих викладачем літературних джерел на задану тематику, а також самостійного пошуку матеріалу.

Опанування дисципліни сприяє розвитку у студентів аналітичного мислення, комунікаційних навичок, командної співпраці, критичного мислення та проблемного вирішення, технічної експертизи.

9.2 Види навчальної діяльності

НД1	Інтерактивні лекції
НД2	Виконання лабораторних робіт за темами 1-7
НД3	Робота з підручниками та релевантними інформаційними джерелами
НД4	Електронне навчання у системі mix.sumdu.edu.ua

10. Методи та критерії оцінювання

10.1. Критерії оцінювання

Визначення	Чотирибальна національна шкала оцінювання	Рейтингова бальна шкала оцінювання
Відмінне виконання лише з незначною кількістю помилок	5 (відмінно)	$90 \leq RD \leq 100$
Вище середнього рівня з кількома помилками	4 (добре)	$82 \leq RD < 89$
Загалом правильна робота з певною кількістю помилок	4 (добре)	$74 \leq RD < 81$
Непогано, але зі значною кількістю недоліків	3 (задовільно)	$64 \leq RD < 73$
Виконання задовольняє мінімальним критеріям	3 (задовільно)	$60 \leq RD < 63$
Можливе повторне складання	2 (незадовільно)	$35 \leq RD < 59$
Необхідний повторний курс з навчальної дисципліни	2 (незадовільно)	$0 \leq RD < 34$

10.2 Методи поточного формативного оцінювання

	Характеристика	Дедлайн, тижні	Зворотний зв'язок

МФО1 Настанови викладача в процесі виконання лабораторних завдань	Перевірка виконаних лабораторних завдань на аудиторному занятті.	1	Коментарі під час заняття від викладача
МФО2 Обговорення та самокорекція виконаної роботи студентами	Обговорення із студентами опрацьованих самостійно розділів рекомендованої додаткової літератури для поглибленого вивчення матеріалу дисципліни	2 тижні	Групові та індивідуальні консультації
МФО3 Перевірка та оцінювання письмових завдань	Вказівка на помилки, допущені студентом при виконанні лабораторних робіт, виявлені при перевірці звіту, завантаженого на платформу електронного навчання.	1 тиждень після здачі роботи	платформа електронного навчання
МФО4 Тести (автоматизовані тести) для самоконтролю	Тести для самоконтролю до кожної теми	Протягом семестру	Платформа електронного навчання
МФО5 Обговорення прикладів протягом лекційних занять	Обговорення із студентами прикладів та практичних кейсів, які розглядаються протягом вивчення теми на лекційному занятті	Протягом лекційного заняття згідно графіку навчального процесу	Освітня платформа для онлайн занять

10.3 Методи підсумкового сумативного оцінювання

	Характеристика	Дедлайн, тижні	Зворотний зв'язок
МСО1 Підсумковий контроль: екзамен	Тестування на платформі онлайн-навчання за теоретичним матеріалом, який вивчався протягом семестру	екзаменаційний тиждень	платформа онлайн-навчання
МСО2 Звіт за результатами виконання лабораторних робіт	Виконання завдань лабораторної роботи та оформлення звіту	1 тиждень	платформа онлайн-навчання, коментарі викладача під час проведення занять

МСО3 Проміжний модульний контроль у формі тестування	Тестування на платформі онлайн-навчання за теоретичним матеріалом, який вивчався протягом модуля атестаційний тиждень 1,2 модуля	атестаційний тиждень	платформа онлайн-навчання
МСО4 Оцінювання відповідей на завдання для дискусій та обговорень	Оцінювання правильності відповідей на поставлені проблемні питання з урахуванням активності студента в обговоренні відповідей інших студентів, що приймають участь у дискусії на сторінці форуму платформи онлайн-навчання.	1 тиждень	Платформа онлайн-навчання

Контрольні заходи:

	Максимальна кількість балів	Мінімальна кількість балів	Можливість перескладання з метою підвищення оцінки
4 семестр	100 балів		
МСО1. Підсумковий контроль: екзамен	40		
	40	24	Ні
МСО2. Звіт за результатами виконання лабораторних робіт	36		
Лабораторні роботи № 1-8 (8x2)	16	10	Ні
Лабораторні роботи № 9-14 (6x3)	18	11	Ні
Лабораторні роботи № 15-16 (2x1)	2	1	Ні
МСО3. Проміжний модульний контроль у формі тестування	20		
2x10	20	12	Ні
МСО4. Оцінювання відповідей на завдання для дискусій та обговорень	4		
4x1	4	2	Ні

Оцінювання протягом семестру проводиться у формі виконання індивідуальних та групових лабораторних робіт. Всі роботи повинні бути виконані самостійно та здані на перевірку вчасно у встановлений викладачем термін. Проміжний контроль знань проводиться наприкінці кожного модуля у формі тестування, при підготовці до якого використовуються матеріали лекцій та рекомендована література. Студент може бути допущений до іспиту, якщо він набрав мінімум 20 балів протягом семестру, і при цьому виконав 60% лабораторних завдань з дисципліни, а також написав модульні контролі з дисципліни. Частина кредитів дисципліни може бути перезарахована за умови успішного проходження масового відкритого онлайн курсу з підтвердженням персоніфікованим сертифікатом освітньої платформи. Перелік масових відкритих онлайн курсів та обсяг кредитів для перезарахування

визначається лектором курсу після детального аналізу структури та наповнення масового відкритого онлайн курсу. Наприклад, для курсу "Introduction to Software Project Management" на платформі Alison при виконанні розділів курсу можуть бути перезараховані лабораторні роботи 1 (Тема 1), 2 (Тема 2), 3-4 (Тема 3) загальним обсягом 8 балів.

11. Ресурсне забезпечення навчальної дисципліни

11.1 Засоби навчання

ЗН1	Мультимедіа, відео- і звуковідтворювальна, проєкційна апаратура (відеокамери, проєктори, екрани, смартдошки тощо)
ЗН2	Прикладне програмне забезпечення (Microsoft Visual Studio)
ЗН3	Комп'ютери, комп'ютерні системи та мережи

11.2 Інформаційне та навчально-методичне забезпечення

Основна література	
1	Бородкіна І. Л. Інженерія програмного забезпечення: Посібник для студентів вищих навчальних закладів / І. Л. Бородкіна, Г. О. Бородкін. - К.: Центр учбової літератури, 2021. - 204 с.
Допоміжна література	
1	Березький О. М. Програмування : навч. посіб. / О. М. Березький, Ю. М. Батько, М. М. Касянчук та ін. — Тернопіль : ТНЕУ, 2018. — 256 с.
2	Коноваленко, І. В. Програмування мовою C# 7.0 : навч. посіб. / І. В. Коноваленко, П. О. Марущак, В. Б. Савків. — Тернопіль : Тернопільський нац. техн. ун-т ім. І. Пулюя, 2017. — 300 с.
Інформаційні ресурси в Інтернеті	
1	Цибульник, С. О. Технології розроблення програмного забезпечення. Частина 1. Життєвий цикл програмного забезпечення [Електронний ресурс] : підручник для здобувачів ступеня бакалавра за спеціальністю 151 «Автоматизація та комп'ютерно-інтегровані технології» / С. О. Цибульник, К. С. Барандич. – Київ : КПІ ім. Ігоря Сікорського, 2022. – 270 с. - Режим доступу: https://ela.kpi.ua/handle/123456789/50623
2	Масовий онлайн-курс "Synchronizing WPF UI Elements Using Event Bindings, TextBox Placeholder WPF, and Async Yield Return" [Електронний ресурс]. - Режим доступу: https://www.classcentral.com/classroom/youtube-synchronizing-wpf-ui-elements-using-event-bindings-textbox-placeholder-wpf-and-async-yield-return-127244
3	Масовий онлайн-курс "Introduction to Software Project Management" [Електронний ресурс]. - Режим доступу: https://alison.com/course/introduction-to-software-project-management

4	Грицюк Ю. І. Особливості визначення вимог до програмного забезпечення та проблеми їх аналізу / Ю. І. Грицюк, І. Ф. Лешкевич // Науковий вісник НЛТУ України. - 2017. - Вип. 27(4). - С. 148-158. - Режим доступу: http://www.irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe?I21DBN=LINK&P21DBN=UJRN&Z21ID=&S21REF=10&S21CNR=20&S21STN=1&S21FMT=ASP_meta&C21COM=S&2_S21P03=FILA=&2_S21STR=nvnltu_2017_27.4_35
5	Кучеров, Д. П. Інженерія програмного забезпечення [Електронний ресурс] : навч. посіб. / Д. П. Кучеров, Є. Б. Артамонов. – Київ : НАУ, 2017. – 386 с. - Режим доступу: https://er.nau.edu.ua/handle/NAU/25927
6	Нагорний В. В. Технологія створення програмних продуктів [Електронний ресурс] : онлайн курс / В. В. Нагорний. - Суми: СумДУ, 2023. - Режим доступу: https://mix.sumdu.edu.ua/info/nmk/f688dcae-65cd-4c81-9cec-6f770cfce4e2