СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ПРОЕКТУВАННЯ ПРОГРАМНИХ СИСТЕМ»

Компонент освітньої програми –
вибірковий (4 *кредити*)

Освітньо-професійна програма	Технології програмування та комп'ютерне моделювання
Спеціальність	113 - Прикладна математика
Галузь знань	11 — Математика і статистика
Рівень вищої освіти	перший бакалаврський
Мова навчання	українська
Профайл викладача (-ів)	Скутар Ігор Дмитрович, канд. фіз.-мат. наук, асистент кафедри ПМІТ *https://amit.chnu.edu.ua/pro-kafedru/spivrobitnyky/skutar-ihor-dmytrovych/*
Контактний тел.	+38(050)9699747
E-mail:	i.skutar@chnu.edu.ua
Сторінка курсу в Moodle	*https://moodle.chnu.edu.ua/course/view.php?id=4423*
Консультації	Вівторок, 14:30 – 15:30

АНОТАЦІЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Дисципліна «Проектування програмних систем» є обов"язковою дисципліною зі спеціальності 113 «Прикладна математика» для освітньої програми «Технології програмування та комп'ютерне моделювання», яка викладається в VI семестрі в обсязі 4 кредити (за Європейською Кредитно-Трансферною Системою ECTS). Дисципліна «Проектування програмних систем» є логічним продовженням вивчення спеціальних дисциплін, здобуті знання знадобляться в подальшій практичній діяльності майбутнім фахівцям галузі ІТ. Курс спрямований на формування у студентів сучасного наукового світогляду та системи спеціальних знань щодо технологій та процесів розробки програмного забезпечення.

Мета навчальної дисципліни: забезпечення студентів необхідними теоретичними знаннями і практичними навиками використання сучасних технологій та підходів у програмуванні для створення програмного забезпечення та роботі в команді розробників.

НАВЧАЛЬНИЙ КОНТЕНТ ОСВІТНЬОЇ КОМПОНЕНТИ

МОДУЛЬ 1	
Тема 1	**МЕТОЛОГІЇ РОЗРОБКИ ПЗ**
Тема 2	**ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**
Тема 3	**UML**
Тема 4	**SOLID-ПРИНЦИПИ ООД**
МОДУЛЬ 2	
Тема 5	**СИСТЕМИ КЕРУВАННЯ ВЕРСІЯМИ**

Тема 6	**РЕФАКТОРИНГ**
Тема 7	**ПАТЕРНИ ПРОЕКТУВАННЯ**
Тема 8	**АНТИПАТЕРНИ**

ФОРМИ, МЕТОДИ ТА ОСВІТНІ ТЕХНОЛОГІЇ НАВЧАННЯ

1. Форми навчання:

- **Лекційні заняття:** Лекції охоплюють теоретичний матеріал з методологій розробки програмного забезпечення, вимог до ПЗ, UML-діаграм, SOLID-принципів, систем керування версіями, патернів та антипатернів проєктування. Лекції супроводжуються презентаціями, які ілюструють складні концепції через приклади та діаграми.
- **Лабораторні роботи:** Лабораторні заняття передбачають виконання проєктів, пов'язаних із розробкою компонентів програмних систем. Студенти працюють над побудовою UML-діаграм, реалізацією та тестуванням коду з дотриманням SOLID-принципів, а також оптимізують існуючі системи шляхом рефакторингу та впровадження патернів.
- **Проєктна робота:** Студенти працюють над командними проєктами, де повинні спроєктувати та реалізувати частину програмної системи з використанням UML, SOLID-принципів, патернів проєктування та систем керування версіями. Цей підхід дозволяє інтегрувати всі вивчені концепції в одному завершеному проєкті.

2. Методи навчання:

- **Проблемно-орієнтоване навчання:** Студенти отримують завдання з аналізу проблем реальних проєктів та пошуку оптимальних рішень. Наприклад, потрібно вибрати відповідний патерн для вирішення конкретної проблеми проєктування або виконати рефакторинг коду, дотримуючись SOLID-принципів.
- **Метод проєктів:** Студенти працюють над командними проєктами, де необхідно застосувати всі вивчені методології та інструменти розробки ПЗ: від створення UML-діаграм до імплементації та тестування коду з використанням систем керування версіями.

3. Освітні технології:

- **Використання інтерактивних середовищ розробки:** IDE середовища активно використовуються під час навчання для написання коду, його тестування та відлагодження.
- **Системи управління навчанням (LMS):** використовується система Moodle для організації дистанційного навчання, завантаження матеріалів та комунікації зі студентами.
- **Системи керування версіями (GitHub, GitLab):** для виконання проєктних завдань студенти активно використовують системи керування версіями. Це дозволяє не лише відслідковувати зміни в коді, але й організовувати командну роботу над спільним проєктом.

Таке поєднання методів, форм і технологій дозволяє забезпечити комплексний підхід до вивчення дисципліни "Технології програмування мовою Python" і сприяє розвитку як теоретичних, так і практичних навичок студентів.

ФОРМИ Й МЕТОДИ КОНТРОЛЮ ТА ОЦІНЮВАННЯ

***Поточний контроль*:** контрольне тестування з кожного модуля, вибіркове опитування на лекційних заняттях.

Підсумковий контроль – *залік.*

КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Оцінювання програмних результатів навчання здобувачів освіти здійснюється за шкалою європейської кредитно-трансферної системи (ECTS).

Критерієм успішного оцінювання є досягнення здобувачем вищої освіти мінімальних порогових рівнів (балів) за кожним запланованим результатом навчання.

ПОЛІТИКА АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні освітнього компонента регламентовано чинними документами ЧНУ:

- «Кодекс академічної доброчесності та етики Чернівецького національного університету імені Юрія Федьковича» https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/kodeks-akademichnoi-dobrochesnosti-ta-etyky-chernivetskoho-natsionalnoho-universytetu-imeni-yuriia-fedkovycha/
- «Положення про виявлення та запобігання академічного плагіату у Чернівецькому національному університеті імені Юрія Федьковича» https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polozhennia-pro-vyiavlennia-ta-zapobihannia-akademichnomu-plahiatu-u-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha/
- «Політика використання штучного інтелекту в Чернівецькому національному університету імені Юрія Федьковича» https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polityka-vykorystannia-shtuchnoho-intelektu-v-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha/

ВИКОРИСТАННЯ ЗДОБУВАЧАМИ ОСВІТИ ІНСТРУМЕНТІВ ШТУЧНОГО ІНТЕЛЕКТУ (ШІ) ПІД ЧАС ВИКОНАННЯ НАВЧАЛЬНИХ ЗАВДАНЬ:

- *допускається за умови* дотримання принципів академічної доброчесності (здобувач освіти несе відповідальність за зміст, достовірність і результати поданої роботи, а у разі використання ШІ зобов'язаний зазначити факт і спосіб його застосування);
- *не допускається* використання ШІ як заміни самостійного виконання завдань або з метою порушення вимог академічної доброчесності;
- *додаткові обмеження чи вимоги* щодо використання ШІ можуть визначатися викладачем в освітньому процесі для конкретних видів навчальної діяльності відповідно до особливостей дисципліни.

ІНФОРМАЦІЙНІ РЕСУРСИ

1. Refactoring.Guru. Design Patterns and Refactoring [Електронний ресурс]. –

Режим доступу: https://refactoring.guru/.
2. Дизайн-патерни — просто, як двері [Електронний ресурс]. – Режим доступу: http://designpatterns.andriybuday.com/.
3. Chacon S., Straub B. Pro Git. 2nd ed. [Електронний ресурс]. – Режим доступу: https://git-scm.com/book/uk/v2.
4. Fowler M. Refactoring [Електронний ресурс]. – Режим доступу: https://refactoring.com/.
5. Agile Manifesto [Електронний ресурс]. – Режим доступу: https://agilemanifesto.org/.
6. Object Management Group. Unified Modeling Language (UML) [Електронний ресурс]. – Режим доступу: https://www.omg.org/spec/UML/.
7. Atlassian. Git Tutorials [Електронний ресурс]. – Режим доступу: https://www.atlassian.com/git/tutorials.

Детальна інформація щодо вивчення курсу «Назва навчальної дисципліни» висвітлена у робочій програмі навчальної дисципліни:

https://media.chnu.edu.ua/media/e31epwx4/113_pr_software_systems_design.pdf