**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Львівський національний університет імені Івана Франка**

**Факультет прикладної математики та інформатики**

**Кафедра прикладної математики**

**Затверджено**

На засіданні кафедри прикладної математики факультету прикладної математики та інформатики

Львівського національного університету імені Івана Франка

(протокол № \_\_\_\_ від \_\_\_\_\_\_\_ 20\_\_ р.)

Завідувач кафедри Ящук Ю.О..

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Силабус з навчальної дисципліни**

**«Проєктування об’єктно-зорієнтованих програм»,**

**що викладається в межах**

**ОПП Прикладна математика**

**другого (магістерського) рівня вищої освіти**

**для здобувачів**

**з спеціальності 113 – прикладна математика**

**Львів 2023 р.**

|  |  |
| --- | --- |
| **Назва дисципліни** | Проектування об’єктно-зорієнтованих програм |
| **Адреса викладання дисципліни** | Головний корпус ЛНУ ім. І. Франка м. Львів, вул. Університетська 1 |
| **Факультет та кафедра, за якою закріплена дисципліна** | Факультет прикладної математики та інформатики Кафедра прикладної математики |
| **Галузь знань, шифр та назва спеціальності** | 11 – математика та статистика113 – прикладна математика |
| **Викладачі дисципліни** | Дяконюк Лілія Миколаївна, доцент кафедри прикладної математики |
| **Контактна інформація викладачів** | liliia.m.diakoniuk@lpnu.uaГоловний корпус ЛНУ ім. І. Франка, каб. 278. м. Львів, вул. Університетська 1 |
| **Консультації з питань навчання** **по дисципліні відбуваються** | Консультації в день проведення лекцій/практичних занять (за попередньою домовленістю). |
| **Сторінка курсу** | [https://ami.lnu.edu.ua/course/](https://ami.lnu.edu.ua/course/dv-net-tehnolohiji)  |
| **Інформація про дисципліну** | Курс передбачає вивчення основних приципів проектування програмного забезпечення для великих програмних комплексів з використанням об’єктно-зорієнтованої парадигми |
| **Коротка анотація дисципліни** | Дисципліна «Проєктування об’єктно-зорієнтованих програм» є базовою дисципліною з спеціальності 113 – прикладна математика для освітньої програми магістрів Прикладна математика, яка викладається в 9-му семестрі в обсязі 4 кредити (за Європейською Кредитно-Трансферною Системою ECTS). |
| **Мета та цілі дисципліни** | Метою вивчення вибіркової дисципліни «Проєктування об’єктно-зорієнтованих програм» є освоєння студентами основних принципів розробки програмних проектів на об’єктно-зорієнтованій мові, створення архітектури програмного забезпечення з використанням шаблонів проектування, яка б давала можливість ефективно вносити зміни в структуру, уникати повторів коду, а також ускладнень, економно використовувати ресурси системи та забезпечувати надійність програмного забезпечення.  |
| **Література для вивчення дисципліни** | ***Базова***1. Erich Gamma. Design Patterns: Elements of Reusable Object-Oriented Software" / Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides (Gamma of Gang of Four). -2. E. Freeman. Head First Design Patterns 2E. / E. Freeman, E.Robson. - O’Reilly Media -2020. - 641р.3. Jeffrey Chilberto. Hands-On Design Patterns with C# and .NET Core. / Jeffrey Chilberto, Gaurav Aroraa. - ISBN-139781789133646. – Packt Publishing. – 2019. – 410 p.3.4. A Shvetc. Design patterns. – Refactoring.guru, 2021. - 410 р. |

|  |  |
| --- | --- |
|  |  **Інформаційні ресурси*** http://www.dofactory.com
* http://sourcemaking.com/design\_patterns
* http://www.citforum.ru/SE/project/pattern/
* <https://refactoring.guru/uk/design-patterns>
* <https://echo.lviv.ua/dev/5394>
* <https://www.journaldev.com/1827/java-design-patterns-example-tutorial>
* <https://www.dofactory.com/net/design-patterns>
* <https://dotnettutorials.net/course/dot-net-design-patterns/>
* <https://www.freecodecamp.org/news/solid-principles-explained-in-plain-english/>
* <https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898>
 |
| **Обсяг курсу** | Загальний обсяг: 96 годин. Аудиторних занять: 64 год., з них 16 год. практичних та 32 години лабораторних робіт. Самостійної роботи: 96 год. |
| **Очікувані результати навчання** | Після завершення цього курсу студент буде **Знати**:* Основні парадигми ОЗП
* Переваги і недоліки принципів Solid
* Мову UML.
* Шаблони проєктування
* Пастки програмування.

**Вміти:*** Будувати різні види UMLдіаграм.
* Обгрунтовувати вибрану схему архітектури програмного забезпечення
* Використовувати шаблони проектування
* Уникати стандартних пасток в проектуванні, які приводять до проблемних ситуацій.
* Реалізовувати шаблони на обраній мові, з врахуванням специфіки реалізації
 |
| **Ключові слова** | Шаблони про’єктування, UML-діаграми, архітектура програмного забезпечення. |
| **Формат курсу** | Очний  |
|  | Проведення лекцій, лабораторних робіт та консультацій. |
| **Теми** | 1. Огляд курсу. Історія створення. Особливості розробки великих програмних комплексів.
2. Огляд парадигм програмування з визначенням місця ОЗП.
3. Мова UML. Типи діаграм та особливості їх використання.
4. Класифікація шаблонів проєктування. Основне призначення.
5. Породжуючі шаблони.
6. Структурні шаблони.
7. Особливості використання поведінкових шаблонів.
8. Стандартні пастки проєктування.
9. Проблеми верифікації програмного забезпечення.
 |
| **Підсумковий контроль, форма** | Комбінований залік вкінці семестру |
| **Пререквізити** | Для вивчення курсу студенти потребують базових знань з курсів:* Об’єктно-орієнтоване програмування
* Програмне забепечення.
* Алгоритми обчисювальних процесів
* Архітектури комп’ютера
* Бази даних
 |

|  |  |
| --- | --- |
| **Навчальні методи та техніки, які будуть** **використовуватися під час** **викладання курсу** | Презентації, лекції Індивідуальні завдання Груповий та індивідуальний проект |
| **Необхідне обладнання** | Комп’ютер із програмним забезпеченням, інтернет зв’язок |
| **Критерії оцінювання** **(окремо для кожного виду** **навчальної діяльності)** | Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням: * індивідуальні завдання : 40% семестрової оцінки; максимальна кількість балів -40.
* розробка проєктів 60% семестрової оцінки; максимальна кількість балів – 60;
* Підсумкова максимальна кількість балів 100.

**Письмові роботи:** Очікується, що студенти оформлять діаграми та створять програми на одній з запропованих мов(с#, java, c++) та захистять виконану роботу Груповий проєкт передбачає публічний захист з презентацією та обгрунтуванням обраної архітектури, демонстрацією роботи програми.**Академічна доброчесність**: Очікується, що програмні роботи студентів будуть створені ними самостійно. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою для її незарахуванння викладачем, незалежно від масштабів плагіату чи обману. **Відвідання занять** є важливою складовою навчання. Очікується, що всі студенти відвідають лекційні та лабораторні заняття курсу. Для студентів передбачені терміни здачі індивідальних завдань. Невчасний захист індивідуальних завдань є підставою до зменшення отриманих балів. Для отримання залікового підсумкового контролю мусять бути виконаний весь пакет завдань, незалежно від кількості отриманих балів за попередні завдання.**Література.** Студентам буде наданий рекомендований список літератури та інформаційних джерел та проведено їх аналіз. Водночас, заохочуватиметься використання також й іншої літератури та джерел, яких немає серед рекомендованих.П**олітика виставлення балів.** Враховуються бали набрані за захист індивідуальних завдань, виконання домашніх робіт, а також за активність роботи на заняттях. Жодні форми порушення академічної доброчесності не толеруються. |
| **Питання до заліку чи екзамену.** | Формуються з деталізацією відповідних зазначених тем в окремому додатку. |
| **Опитування** | Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу. |

Схема курсу

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тиж.  | Тема, план, короткі тези  | **Форма діяльності (заняття)**  | Література. Ресурси в інтернеті  | Завдання, год.  | Термін виконання  |
| 1  | **Тема 1.** Огляд курсу. Особливості розробки великих програмних комплексів.   | лекція, лабораторна роботасамостійна робота  | [1]-[4]  | 1 | 2тижні  |
| 2  | **Тема 2. Огляд парадигм програмування з визначенням місця ОЗП.** | лекція, лабораторна робота, самостійна робота  | [1]-[4]  | 2  | 2тижні  |
| **Тема 3. Мова UML.** Типи діаграм та особливості їх використання.Побудова діаграм взаємозв’язків між класами, діаграм прецедентів.  | Лекція, лабораторна робота, самостійна робота  | [1]-[4]  | 3 | 2тижні  |
| 3  | **Тема 4.**   **Класифікація шаблонів проєктування.** Принципи SOLID. Основне призначення шаблонів проєктування. | Лекція, лабораторна робота, самостійна робота   | [1]-[4]  | 4 | 2тижні  |
| 4  | **Тема 5.** **Породжуючі шаблони.**  | Лекція, лабораторна робота, самостійна робота  | [1]-[4]  | 5 | 2тижні   |
| **Тема 6. Структурні паттерни.** | Лекція, лабораторна робота, самостійна робота  | [1]-[4]  | 6 | 2тижні  |
| 5  | **Тема 7. Поведінкові шаблони.** | Лекція, лабораторна робота, самостійна робота  | [1]-[4]  | 7 | 2тижні  |
| 6  | **Тема 8. Стандартні пастки проєктування** | Лекція, лабораторна робота, самостійна робота   | [1],[2],[4]  | 8 | 2тижні  |
| **Тема 9. Захисти та обговорення групових проєктів.** | Лекція,лаборатор-на робота, самостійна робота  | [2],[3]  | 9 | 2тижні  |
|  |  |  |  |  |  |