



Інженерія програмного забезпечення

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>123 Комп'ютерна інженерія</i>
Освітня програма	<i>Комп'ютерні системи та мережі</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>очна(денна)/заочна</i>
Рік підготовки, семестр	<i>2 курс, 3 семестр</i>
Обсяг дисципліни	<i>5 кредитів (ECTS). Загальна кількість годин - 150, включаючи 72 годин роботи в класі (заочна форма - 16) та 78 години самостійного навчання (заочна форма - 134).</i>
Семестровий контроль/ контрольні заходи	<i>3 семестр - іспит</i>
Розклад занять	<i>2 заняття на тиждень за розкладом http://rozklad.kpi.ua/</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: 3 семестр - к.т.н, доцент, Андрій Антонюк, моб.: +380503588019, Email: antoniuk.andrii@ill.kpi.ua, особиста веб-сторінка: http://se-111.blogspot.com Лабораторні: 3 семестр - к.т.н, доцент, Андрій Антонюк, моб.: +380503588019, Email: antoniuk.andrii@ill.kpi.ua, особиста веб-сторінка: http://se-111.blogspot.com</i>
Розміщення курсу	<i>BigBlueButton, Cisco, Google, TeamViewer, Zoom, http://se-111.blogspot.com</i>

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Метою вивчення кредитного модуля "Інженерія програмного забезпечення" є надбання студентом практичних навичок проектування та побудови програм для обчислювальних систем:

- вивчення принципів проектування та побудови програмного забезпечення для комп'ютерних систем
- вивчення шаблонного підходу до проектування програмних систем та мови UML
- вивчення методів та засобів розробки програмного забезпечення
- вивчення об'єктно-орієнтованого підходу до розробки програмного забезпечення.

Предметом кредитного модуля "Інженерія програмного забезпечення" є - методи, способи, шаблони, інструментальні засоби та системи для дослідження, автоматизованого та автоматичного проектування, налагодження, виробництва й експлуатації, проектна документація, стандарти, процедури та засоби підтримки керування життєвим циклом програмних засобів.

Студенти після засвоєння кредитного модуля мають продемонструвати такі **програмні результати навчання (ФКЗ, ФК8, ПРН10)::**

знання:

- етапів, методів та стандартів розробки програмного забезпечення;
- шаблонного методу проектування;

- базових шаблонів проектування програмного забезпечення та принципів їх побудови та застосування,
- принципів компонентного проектування.

вміння:

- визначати джерела вимог і забезпечувати процес їх формування;
- розробляти специфікації вимог користувачів;
- здійснювати аналіз вимог;
- розробляти специфікацію програмних вимог;
- моделювати різні аспекти системи;
- проектувати компоненти архітектурного рішення;
- застосовувати та створювати компоненти багаторазового використання.

досвід:

- шаблонного методу проектування;
- розробки специфікацій вимог користувачів;
- моделювання різних аспектів системи;
- застосування та створення компонент багаторазового використання.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Перелік дисциплін, або знань та умінь, володіння якими необхідні студенту:

- базовий рівень володіння мовою програмування Java, або JavaScript, або C++, або Python;
- базовий рівень володіння засобами інтегрованого середовища розробки Eclipse або IntelliJ IDEA;
- базовий рівень володіння засобами UML моделювання ArgoUML або Umbrello.

На результатах навчання з даної дисципліни базуються такі наступні дисципліни:

- курсова робота з Інженерії програмного забезпечення;
- організація обчислювальних процесів;
- паралельне програмування;

3. Зміст навчальної дисципліни

Перелік розділів та тем всієї дисципліни:

- Розділ 1. Методології розробки ПЗ:
- Тема 1.1. Основні поняття та проблеми розробки ПЗ.
- Тема 1.2. Життєвий цикл ПЗ; міжнародні стандарти життєвого циклу ПЗ.
- Тема 1.3. Моделі та методології розробки ПЗ.
- Тема 1.4. Аналіз, специфікація, верифікація та валідація вимог до ПЗ. Функціональні та нефункціональні вимоги.
- Розділ 2. Засоби та середовища створення ПЗ:
- Тема 2.1. Інтегровані середовища розробки ПЗ.
- Тема 2.2. Системи управління проектами (Redmine, JIRA).
- Тема 2.3. Системи управління версіями документів, архітектурні особливості (CVS, SVN, Git).
- Тема 2.4. Інструменти автоматизації зборки проектів (утиліта make, системи CMake, Ant та Maven).
- Розділ 3. Проектування ПЗ з використанням шаблонів:
- Тема 3.1. Проектування архітектури ПЗ.
- Тема 3.2. Шаблони проектування ПЗ. Класифікація шаблонів проектування. Графічна нотація.
- Тема 3.3. Структурні шаблони проектування Composite, Decorator, Proxy.
- Тема 3.4. Структурні шаблони проектування Flyweight, Adapter, Bridge, Facade.
- Тема 3.5. Шаблони поведінки Iterator, Mediator, Observer.
- Тема 3.6. Шаблони поведінки Strategy, Chain of Responsibility, Visitor.

- Тема 3.7. Шаблони поведінки *Memento, State, Command, Interpreter*.
- Тема 3.8. Породжувальні шаблони проектування *Prototype, Singleton, Factory Method*.
- Тема 3.9. Породжувальні шаблони проектування *Abstract Factory, Builder*.
- Розділ 4. Проектування інтерфейсу користувача:
- Тема 4.1. Призначення та типи інтерфейсів користувача.
- Тема 4.2. Принципи побудови графічного інтерфейсу користувача.
- Тема 4.3. Моделі подій. Моделі промальовування графічного інтерфейсу користувача.
- Розділ 5. Моделювання програмного забезпечення:
- Тема 5.1. Методології моделювання *SADT, IDEF, DFD, ELM, OOAD*. Мови моделювання.
- Тема 5.2. Інформаційне моделювання. Діаграми сутність-зв'язок, класів.
- Тема 5.3. Моделювання бізнес-процесів, організацій та цілей.
- Тема 5.4. Поведінкове моделювання. Діаграми станів, діяльності, взаємодії, послідовності, часові.
- Тема 5.5. Структурне моделювання.
- Тема 5.6. Функціональне моделювання.
- Тема 5.7. Моделювання потоків даних.
- Тема 5.8. Засоби автоматизації моделювання (*ERWin, BPWin, Enterprise Architect* та інші).
- Розділ 6. Методи забезпечення та контролю якості ПЗ:
- Тема 6.1. Якість ПЗ; метрики і стандарти якості ПЗ.
- Тема 6.2. Верифікація та валідація ПЗ.
- Тема 6.3. Тестування ПЗ.
- Тема 6.4. Принципи постійної інтеграції ПЗ.
- Тема 6.5. Оптимізація коду та рефакторинг.
- Тема 6.6. Аспекти продуктивності ПЗ.
- Тема 6.7. Інструменти автоматизації процесів тестування (*JUnit, JMeter*).
- Тема 6.8. Сервера постійної інтеграції (*Hudson, CruiseControl*).
- Розділ 7. Менеджмент програмних проектів:
- Тема 7.1. Задачі управління проектами. Трикутник обмежень.
- Тема 7.2. Управління змістом та якістю проекту.
- Тема 7.3. Управління ресурсами. Планування графіку виконання проекту.
- Тема 7.4. Управління ризиками програмного проекту.
- Тема 7.5. Управління конфігураціями та змінами.
- Тема 7.6. Контроль та моніторинг стану проекту. Метрики контролю.
- Тема 7.7. Організація роботи проектної команди. Ролі та зони відповідальності учасників команди.
-

4. Навчальні матеріали та ресурси

Базова література:

1. *Design Patterns: elements of reusable object-oriented software / Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Indianapolis: - Addison-Wesley, 1994. -417 p. ISBN: 0201633612.*
2. *Mark Grand. Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML, Volume 1, 2nd Edition. - Wiley Publishing, 2002. - 480 p. ISBN: 0471258393*
3. *Інженерія програмного забезпечення: методичні вказівки до виконання лабораторних робіт для студентів спеціальності 123 «Комп'ютерна інженерія»: Ч. I. Структурні шаблони. / Уклад.: А. О. Болдак, О. Н. Абу Усбах. - К.: НТУУ «КПІ», 2011. - 40 с.*
4. *Інженерія програмного забезпечення: методичні вказівки до виконання лабораторних робіт для студентів спеціальності 123 «Комп'ютерна інженерія»: Ч. II. Шаблони поведінки та породжувальні шаблони / Уклад.: А. О. Болдак, О. Н. Абу Усбах. - К.: НТУУ «КПІ», 2011. - 44 с.*
5. *Theo Mandel. The Elements of User Interface Design. - John Wiley & Sons, 2005. - 468 p. ISBN: 0471162671*

Додаткова література:

1. Програмна інженерія: [Підручник] / Лавріщева К.М. – К.: Академперіодика, 2008. – 319 с. ISBN 978–966–02–5052–9
2. Об'єктно-орієнтоване програмування: [Підручник] / В.В. Бублик. – К.: ІТ-книга, 2015. – 624 с.
3. Бандура В.В., Храбатин Р.І. Архітектура та проектування програмного забезпечення: конспект лекцій. — Івано-Франківськ: ІФНТУНГ, 2012. — 240 с.
4. Проектування інформаційних систем: Загальні питання теорії проектування ІС (конспект лекцій): навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського; уклад.: О. С. Коваленко, Л. М. Добровська. – Київ : КПІ ім. Ігоря Сікорського, 2020. – 192с.
5. Поморова О.В., Говорущенко Т.О. Проектування інтерфейсів користувача: навч.посібник - Хмельницький: ХНУ, 2011. - 206 с.
6. Проектування та моделювання програмного забезпечення сучасних інформаційних систем / Г. В. Табунщик, Т.І. Каплієнко, О.А. Петрова – Запоріжжя : Дике Поле, 2016. – 250 с.
7. Навчальний посібник з дисципліни «Технології розробки програмного забезпечення» для студентів спеціальності 123 «Комп'ютерна інженерія» / Дегтярьова Л.М., Гроза П.М., Сомов С.В. – Полтава: ПолтНТУ, 2017. – 218 с.
8. Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд./ Пер. с англ.; Под общей редакцией проф. С. Орлова — СПб.: Питер, 2007. — 736 с.
9. Вебер Дж. Технология Java./ Пер. с англ. – СПб.: БХВ-Петербург, 1997. – 1104 с.
10. Кериевски Дж. Рефакторинг с использованием шаблонов / Пер. с англ. — М.: «Диалектика», 2019. — 400 с. ISBN 978-5-907144-79-8
11. Эккель Б. Философия Java. Библиотека программиста./ Пер. с англ. – С-Пб.: Питер, 2003. – 976 с.
12. Ларман К. Применение UML 2.0 и шаблонов проектирования ./ Пер. с англ. – М.: Новое знание, 2013. – 736 с.
13. Ноутон П., Шилт Г. Java 2. / Пер. с англ. – СПб.,: БХВ-Петербург, 2000. – 1072 с.
14. Фаулер М. Архитектура корпоративных программных приложений / Пер. с англ. — М.: «Вильямс», 2007. — 544 с.
15. Хорстманн К., Корнелл Г. Java 2. Библиотека профессионала, том 1. Основы. / Пер. с англ. — 9-е изд. — М.: «Вильямс», 2014. — 864 с.
16. Хорстманн К., Корнелл Г. Java 2. Библиотека профессионала, том 2. Тонкости программирования / Пер. с англ. — 9-е изд. — М.: «Вильямс», 2014. — 1008 с.
17. Шилдт Г., Холмс Дж. Искусство программирования на Java / Пер. с англ. — М.: «Диалектика», 2005. — 336 с.
18. Шмуллер Дж. Освой самостоятельно UML 2 за 24 часа. Практическое руководство. / Пер. с англ. — М.: «Вильямс», 2005. — 416 с.
19. Эмблер С., Садаладж П. Рефакторинг баз данных: эволюционное проектирование. / Пер. с англ. — М.: «Вильямс», 2007. — 368 с.
20. Гамма Э. и др., Приемы объектно-ориентированного проектирования. Паттерны проектирования. / Пер. с англ. – С-Пб.: Питер, 2001. – 366 с.
21. Гранд М. Шаблоны проектирования в Java. / Пер. с англ. – М.: Новое знание, 2004. – 560 с.
22. Мандел Т. Разработка пользовательского интерфейса. / Пер. с англ. – М.: ДМК Пресс, 2001. – 416 с.

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття (очна форма)

№ лекції	Назва теми лекції та перелік основних питань: перелік дидактичних засобів, посилання на літературу та завдання на самостійну роботу студента (CPC)
1.	Життєвий цикл ПЗ Основні поняття та проблеми розробки ПЗ. Життєвий цикл ПЗ; міжнародні стандарти життєвого циклу ПЗ [1, с. 12-14; 2, с. 41-60] CPC: Стандарти ISO пов'язані з розробкою ПЗ
2.	Моделі та методології розробки Моделі та методології розробки. Аналіз, специфікація, верифікація та валідація вимог до ПЗ. Функціональні та нефункціональні вимоги [1, с. 12-14; 2, с. 41-60] CPC: Гнучкі методології розробки
3.	Інтегровані середовища розробки ПЗ Інтегровані середовища розробки ПЗ. Системи управління проектами (Redmine, JIRA). [1, с. 15-45; 2, с. 19-40; 3, с. 5-7] CPC: Установка і використання систем управління проектами
4.	Управління версіями та автоматизація збірки ПЗ Системи управління версіями документів, архітектурні особливості (CVS, SVN, Git). Інструменти автоматизації зборки проектів (утиліта make, системи CMake, Ant та Maven). [1, с. 15-45; 2, с. 19-40; 3, с. 5-7] CPC: Установка і конфігурування SVN. Застосування Ant для роботи з SVN
5.	Шаблони проектування ПЗ. Проектування архітектури ПЗ. Шаблони проектування ПЗ. Класифікація шаблонів проектування. Графічна нотація [1, с. 140-141, 162-173] CPC: Шаблиони проектування GRASP
6.	Структурні шаблони проектування: Composite та Decorator. Шаблон Composite: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Composite. Шаблон Decorator: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Decorator. [1, с. 173-183, 203-213; 3, с.19-22] CPC: Реалізації Composite та Decorator в класах Java SDK
7.	Структурні шаблони проектування: Proxy та Flyweight. Шаблон Proxy: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Proxy. Шаблон Flyweight: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Flyweight. [1, с. 191-203, 141-152; 3, с.19-22, 29-33] CPC: Реалізації Proxy та Flyweight в класах Java SDK
8.	Структурні шаблони проектування: Adapter, Bridge та Facade. Шаблон Adapter: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Adapter. Шаблон Bridge: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Bridge. Шаблон Facade: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Facade. [1, с. 152-162, 183-191; 3, с.29-33] CPC: Реалізації Adapter, Bridge та Facade в класах Java SDK

№ лекції	Назва теми лекції та перелік основних питань: перелік дидактичних засобів, посилання на літературу та завдання на самостійну роботу студента (СРС)
9.	Шаблони поведінки: Iterator та Mediator. Призначення шаблонів поведінки. Шаблон Iterator: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Iterator. Шаблон Mediator: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Mediator. [1, с. 216-217, 249-263; 4, с. 6-8] СРС: Реалізації Iterator та Mediator в класах Java SDK
10.	Шаблони поведінки: Observer та Strategy. Шаблон Observer: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Observer. Шаблон Strategy: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Strategy. [1, с. 263-272, 280-291; 4, с. 6-8, 14-16] СРС: Реалізації Observer та Strategy в класах Java SDK
11.	Шаблони поведінки: Command та Visitor. Шаблон Command: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Command. Шаблон Visitor: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Visitor. [1, с. 227-236, 300-309; 4, с. 14-16, 22-25] СРС: Реалізації Command та Visitor в класах Java SDK
12.	Шаблони поведінки: State та Memento. Шаблон State: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону State. Шаблон Memento: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Memento. [1, с. 272-280, 291-300, 314-328; 4, с. 22-25] СРС: Реалізації State та Memento в класах Java SDK
13.	Шаблони поведінки: Interpreter та Chain of Responsibility. Шаблон Interpreter: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Interpreter. Шаблон Chain of Responsibility: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Chain of Responsibility. [1, с. 238-249, 217-226; 4, с. 14-16, 22-25] СРС: Реалізації Interpreter та Chain of Responsibility в класах Java SDK
14.	Породжувальні шаблони Prototype, Factory Method та Abstract Factory. Призначення шаблонів, що породжують. Шаблон Prototype: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Prototype. [1, с. 89-93, 121-130; 4, с. 30-32]. Шаблон Factory Method: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Factory Method. Шаблон Abstract Factory: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Abstract Factory. [1, с. 111-121, 93-102; 4, с. 30-32, 37-38] СРС: Реалізації Prototype, Factory Method та Abstract Factory в класах Java SDK
15.	Шаблони Singleton та Builder. Шаблон Singleton: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Singleton. Шаблон Builder: мотивація, структура, учасники, відносини, шляхи застосування та результат

№ лекції	Назва теми лекції та перелік основних питань: перелік дидактичних засобів, посилання на літературу та завдання на самостійну роботу студента (СРС)
	використання. Приклад реалізації шаблону Builder. [1, с. 130-138, 102-111; 4, с. 30-32, 37-38] СРС: Реалізації Singleton та Builder в класах Java SDK
16.	Графічний інтерфейс користувача. Призначення та типи інтерфейсів користувача. Принципи побудови графічного інтерфейсу користувача. [5, с. 41-56] СРС: Фреймовий та віконний інтерфейси користувача
17.	Моделі подій та промальовування. Моделі подій. Моделі промальовування графічного інтерфейсу користувача. [5, с. 65-91] СРС: Реалізація легких компонентів на прикладі SWING
18.	Інформаційне моделювання та моделювання бізнес-процесів Методології моделювання SADT, IDEF, DFD, ELM, OOAD. Мови моделювання. Інформаційне моделювання. Діаграми сутність-зв'язок, класів. Моделювання бізнес-процесів, організацій та цілей. [6, с. 506-537] СРС: Rational Unified Process
19.	Поведінкове, структурне та функціональне моделювання Поведінкове моделювання. Діаграми станів, діяльності, взаємодії, послідовності, часові. Структурне моделювання. Функціональне моделювання. [6, с. 506-537] СРС: Моделювання з використанням UML
20.	Моделювання потоків даних Моделювання потоків даних. Засоби автоматизації моделювання (ERWin, BPWin, Enterprise Architect та інші). [6, с. 506-537] СРС: Засоби автоматизації моделювання фірми IBM
21.	Тестування ПЗ Якість ПЗ; метрики і стандарти якості ПЗ. Верифікація та валідація ПЗ. Тестування ПЗ. Інструменти автоматизації процесів тестування (JUnit, JMeter). [6, с. 368-402] СРС: Тестування з використанням JUnit. Відмінності версій
22.	Постійна інтеграція ПЗ Принципи постійної інтеграції ПЗ. Оптимізація коду та рефакторинг. Аспекти продуктивності ПЗ. Сервера постійної інтеграції (Hudson, CruiseControl). [6, с. 368-402] СРС: Установка і використання сервера постійної інтеграції
23.	Управління проектами Задачі управління проектами. Трикутник обмежень. Управління змістом та якістю проекту. [6, с. 562-597] СРС: Засоби автоматизації управління проектами
24.	Управління ресурсами, ризиками та конфігураціями Управління ресурсами. Планування графіку виконання проекту. Управління ризиками програмного проекту. Управління конфігураціями та змінами [6, с. 562-597] СРС: Засоби управління ресурсами, ризиками та конфігураціями
25.	Контроль та моніторинг стану проекту Контроль та моніторинг стану проекту. Метрики контролю. Організація роботи проектною командою. Ролі та зони відповідальності учасників команди. [6, с. 562-597] СРС: Робота команди з гнучкою методологією

Лекційні заняття (заочна форма)

<i>№ лекції</i>	<i>Назва теми лекції та перелік основних питань: перелік дидактичних засобів, посилання на літературу та завдання на самостійну роботу студента (СРС)</i>
1.	Основні поняття та проблеми розробки ПЗ. Життєвий цикл ПЗ; міжнародні стандарти життєвого циклу ПЗ [1, с. 12-14; 2, с. 41-60] Моделі та методології розробки. Аналіз, специфікація, верифікація та валідація вимог до ПЗ. Функціональні та нефункціональні вимоги [1, с. 12-14; 2, с. 41-60] СРС: Стандарти ISO пов'язані з розробкою ПЗ, Гнучкі методології розробки
2.	Шаблони проектування ПЗ. Структурні шаблони проектування: Adapter, Bridge, Composite, Decorator, Facade, Flyweight та Proxy. Призначення, склад, структура, робота [1, с. 141-162, с. 173-213; 3, с.19-22, с.29-33].
3.	Шаблони проектування ПЗ. Шаблони поведінки: Iterator, Mediator, Observer, Strategy, Command, Visitor, State, Memento, Interpreter, Chain of Responsibility. Призначення, склад, структура, робота [1, с. 216-236, 238-300; 4, с. 6-8, 14-16, 22-25]
4.	Шаблони проектування ПЗ. Породжувальні шаблони: Prototype, Factory Method, Abstract Factory, Singleton, Builder. Призначення, склад, структура, робота [1, с. 89-93, 102-111, 121-138; 4, с. 30-32, 37-38]

Лабораторні заняття:

Метою проведення циклу лабораторних робіт є придбання студентами необхідних практичних навичок розробки програмного забезпечення з використанням об'єктно-орієнтованого підходу, застосування шаблонів проектування та моделювання з використанням мови UML.

Лабораторна робота включає:

- 1. розробку та аналіз математичного алгоритму рішення задачі;*
- 2. декомпозицію задачі;*
- 3. застосування шаблонів проектування;*
- 4. роздруківку програми;*
- 5. UML діаграми взаємодії класів;*
- 6. результати виконання програми;*
- 7. документування лабораторної роботи та зробленої програми*
- 8. висновки щодо засвоєння теми лабораторної роботи.*

Очна форма

<i>№ з/п</i>	<i>Назва лабораторної роботи (комп'ютерного практикуму)</i>	<i>Кількість ауд. годин</i>
1.	Проект програмного додатку. Збірка проекту за допомогою Ant. (Розділ 2, Теми 2.1-2.2)	2
2.	Графічна нотація UML. Документування за допомогою JavaDoc. (Розділ 2, Теми 2.3- 2.4)	2
3.	Структурні шаблони проектування ПЗ. Composite, Decorator, Proxy. (Розділ 3, Тема 3.3)	2
4.	Структурні шаблони проектування ПЗ - 2 . Flyweight, Adapter, Bridge, Facade. (Розділ 3, Тема 3.4)	2
5.	Шаблони поведінки. Iterator, Mediator, Observer. (Розділ 3, Тема 3.5)	2
6.	Шаблони поведінки - 2. Strategy, Chain of Responsibility, Visitor. (Розділ 3, Тема 3.6)	2
7.	Шаблони поведінки - 3. Memento, State, Command, Interpreter. (Розділ 3, Тема 3.7)	2

8.	Породжувальні шаблони. Prototype, Singleton, Factory Method. (Розділ 3, Тема 3.8)	2
9.	Породжувальні шаблони - 2. Abstract Factory, Builder. (Розділ 3, Тема 3.9)	2
	Разом:	18

Заочна форма

№ з/п	Назва лабораторної роботи (комп'ютерного практикуму)	Кількість ауд. годин
1.	Проект програмного додатку. Збірка проекту за допомогою Ant. (Розділ 2, Теми 2.1-2.2). Графічна нотація UML. Документування за допомогою JavaDoc. (Розділ 2, Теми 2.3- 2.4)	2
2.	Структурні шаблони проектування ПЗ Composite, Decorator, Proxy. (Розділ 3, Тема 3.3). Структурні шаблони проектування ПЗ Flyweight, Adapter, Bridge, Facade. (Розділ 3, Тема 3.4).	2
3.	Шаблони поведінки Iterator, Mediator, Observer. (Розділ 3, Тема 3.5). Шаблони поведінки Strategy, Chain of Responsibility, Visitor. (Розділ 3, Тема 3.6). Шаблони поведінки Memento, State, Command, Interpreter. (Розділ 3, Тема 3.7)	2
4.	Породжувальні шаблони Prototype, Singleton, Factory Method. (Розділ 3, Тема 3.8). Породжувальні шаблони Abstract Factory, Builder. (Розділ 3, Тема 3.9)	2
	Разом:	8

6. Самостійна робота студента

Очна форма

№ з/п	Назва теми, що виносить на самостійне опрацювання	Кількість годин
1.	Стандарти ISO пов'язані з розробкою ПЗ ISO-12207, ISO-15504, ISO 9000 – ISO 9004 [Додаткова 3, с. 6-16; Додаткова 7, с.43-55]	4
2.	Гнучкі методології розробки [Додаткова 7, с.17-36, 67-83; https://uk.wikipedia.org/wiki/Гнучка_розробка_програмного_забезпечення]	4
3.	Установка і використання систем управління проектами [Додаткова 2, с.384-441; https://uk.wikipedia.org/wiki/Управління_проектами]	4
4.	Установка і конфігурування SVN. Застосування Ant для роботи з SVN [Додаткова 3, с.17-36; https://uk.wikipedia.org/wiki/Subversion ; https://uk.wikipedia.org/wiki/Apache_Ant]	4
5.	Шаблони проектування GRASP [Додаткова 2 с. 105-134; Додаткова 9 с. 299-386; https://uk.wikipedia.org/wiki/GRASP]	2
6.	Реалізації Composite та Decorator в класах Java SDK [1, с. 162-182; https://uk.wikipedia.org/wiki/SDK , http://www.oracle.com/technetwork/java/javase/downloads/index.html]	2
7.	Реалізації Proxy та Flyweight в класах Java SDK [1, с. 191-212]	2
8.	Реалізації Adapter, Bridge та Facade в класах Java SDK [1, с. 141-161, 183-190]	2
9.	Реалізації Iterator та Mediator в класах Java SDK [1, с. 249-271]	2
10.	Реалізації Observer та Strategy в класах Java SDK [1, с. 280-290, 300-308]	2
11.	Реалізації Command та Visitor в класах Java SDK [1, с. 227-235, 314-327]	2
12.	Реалізації State та Memento в класах Java SDK [1, с. 272-279, 291-299]	2

13.	Реалізації Interpreter та Chain of Responsibility в класах Java SDK [1, с. 217-226, 236-248]	2
14.	Реалізації Prototype, Factory Method та Abstract Factory в класах Java SDK [1, с. 93-101, 111-129]	2
15.	Реалізації Singleton та Builder в класах Java SDK [1, с. 102-110, 130-137]	2
16.	Фреймовий та віконний інтерфейси користувача [5, с. 61-69; https://uk.wikibooks.org/wiki/Освоюємо_Java/Графічний_інтерфейс_ко_ристувача]	4
17.	Реалізація легких компонентів на прикладі SWING [5 с. 8-18, 61-69; https://uk.wikibooks.org/wiki/Освоюємо_Java/Менеджери_розташуван_ня]	4
18.	Раціональний уніфікований процес (Rational Unified Process) [Додаткова 7, с. 280-289; https://uk.wikipedia.org/wiki/Rational_Unified_Process]	4
19.	Моделювання з використанням UML [Додаткова 2, с. 599-602; Додаткова 5, с. 34-197; Додаткова 7, 204-246]	4
20.	Засоби автоматизації моделювання фірми IBM [Додаткова 7, с. 443-456; https://uk.wikipedia.org/wiki/Автоматизація_процесу_програмування , http://posibniki.com.ua/post-avtomatizaciya-analizu--ta-proektuvannya--za-dopomogoyu-ibm-rational-rose , http://easy-code.com.ua/2011/05/kerovana-modellyu-rozrobka-nastupnogo-pokolinnya-ibm-rational]	4
21.	Тестування з використанням JUnit. Відмінності версій [Додаткова 1, с.751-765; Додаткова 4, с.91-124; https://uk.wikipedia.org/wiki/Junit , https://habrahabr.ru/post/120101/]	4
22.	Установка і використання сервера постійної інтеграції [6, с. 368-402; https://uk.wikipedia.org/wiki/Неперервна_інтеграція]	4
23.	Засоби автоматизації управління проектами [6, с. 562-597; https://uk.wikipedia.org/wiki/Управління_проектами , https://uk.wikipedia.org/wiki/Розробка_програмного_забезпечення]	4
24.	Засоби управління ресурсами, ризиками та конфігураціями [6, с. 562-597, https://uk.wikipedia.org/wiki/Управління_розробкою_програмного_забезпечення , https://uk.wikipedia.org/wiki/DSDM]	4
25.	Робота команди з гнучкою методологією [6, с. 562-597, https://uk.wikipedia.org/wiki/Гнучка_розробка_програмного_забезпечення]	4
26.	Разом	78

Заочна форма

№ з/п	Назва теми, що виносить на самостійне опрацювання	Кількість годин
1.	Стандарти ISO пов'язані з розробкою ПЗ ISO-12207, ISO-15504, ISO 9000 – ISO 9004 [Додаткова 3, с. 6-16; Додаткова 7, с.43-55]	4
2.	Гнучкі методології розробки [Додаткова 7, с.17-36, 67-83; https://uk.wikipedia.org/wiki/Гнучка_розробка_програмного_забезпечення]	4
3.	Інтегровані середовища розробки ПЗ. Системи управління проектами (Redmine, JIRA). [1, с. 15-45; 2, с. 19-40; 3, с. 5-7]	4

4.	Управління версіями та автоматизація збірки ПЗ. Системи управління версіями документів, архітектурні особливості (CVS, SVN, Git). Інструменти автоматизації зборки проектів (утиліта make, системи CMake, Ant та Maven). [1, с. 15-45; 2, с. 19-40; 3, с. 5-7]	4
5.	Проектування архітектури ПЗ. Шаблони проектування ПЗ. Класифікація шаблонів проектування. Графічна нотація [1, с. 140-141, 162-173]	4
6.	Графічний інтерфейс користувача. Призначення та типи інтерфейсів користувача. Принципи побудови графічного інтерфейсу користувача. [5, с. 41-56]	4
7.	Моделі подій та промальовування. Моделі подій. Моделі промальовування графічного інтерфейсу користувача. [5, с. 65-91]	4
8.	Інформаційне моделювання та моделювання бізнес-процесів. Методології моделювання SADT, IDEF, DFD, ELM, OOAD. Мови моделювання. Інформаційне моделювання. Діаграми сутність-зв'язок, класів. Моделювання бізнес-процесів, організацій та цілей. [6, с. 506-537]	4
9.	Поведінкове, структурне та функціональне моделювання. Поведінкове моделювання. Діаграми станів, діяльності, взаємодії, послідовності, часові. Структурне моделювання. Функціональне моделювання. [6, с. 506-537]	4
10.	Моделювання потоків даних. Моделювання потоків даних. Засоби автоматизації моделювання (ERWin, BPWin, Enterprise Architect та інші). [6, с. 506-537]	4
11.	Тестування ПЗ. Якість ПЗ; метрики і стандарти якості ПЗ. Верифікація та валідація ПЗ. Тестування ПЗ. Інструменти автоматизації процесів тестування (JUnit, JMeter). [6, с. 368-402]	4
12.	Постійна інтеграція ПЗ. Принципи постійної інтеграції ПЗ. Оптимізація коду та рефакторинг. Аспекти продуктивності ПЗ. Сервера постійної інтеграції (Hudson, CruiseControl). [6, с. 368-402]	4
13.	Управління проектами. Задачі управління проектами. Трикутник обмежень. Управління змістом та якістю проекту. [6, с. 562-597]	4
14.	Управління ресурсами, ризиками та конфігураціями. Управління ресурсами. Планування графіку виконання проекту. Управління ризиками програмного проекту. Управління конфігураціями та змінами [6, с. 562-597]	4
15.	Контроль та моніторинг стану проекту. Контроль та моніторинг стану проекту. Метрики контролю. Організація роботи проектної команди. Ролі та зони відповідальності учасників команди. [6, с. 562-597]	4
16.	Установка і використання систем управління проектами [Додаткова 2, с.384-441; https://uk.wikipedia.org/wiki/Управління_проектами]	4
17.	Установка і конфігурування SVN. Застосування Ant для роботи з SVN [Додаткова 3, с.17-36; https://uk.wikipedia.org/wiki/Subversion ; https://uk.wikipedia.org/wiki/Apache_Ant]	4
18.	Шаблони проектування GRASP [Додаткова 2 с. 105-134; Додаткова 9 с. 299-386; https://uk.wikipedia.org/wiki/GRASP]	4
19.	Реалізації Composite та Decorator в класах Java SDK [1, с. 162-182; https://uk.wikipedia.org/wiki/SDK , http://www.oracle.com/technetwork/java/javase/downloads/index.html]	4
20.	Реалізації Proxy та Flyweight в класах Java SDK [1, с. 191-212]	2
21.	Реалізації Adapter, Bridge та Facade в класах Java SDK [1, с. 141-161, 183-190]	2

22.	Реалізації Iterator та Mediator в класах Java SDK [1, с. 249-271]	2
23.	Реалізації Observer та Strategy в класах Java SDK [1, с. 280-290, 300-308]	2
24.	Реалізації Command та Visitor в класах Java SDK [1, с. 227-235, 314-327]	2
25.	Реалізації State та Memento в класах Java SDK [1, с. 272-279, 291-299]	2
26.	Реалізації Interpreter та Chain of Responsibility в класах Java SDK [1, с. 217-226, 236-248]	2
27.	Реалізації Prototype, Factory Method та Abstract Factory в класах Java SDK [1, с. 93-101, 111-129]	2
28.	Реалізації Singleton та Builder в класах Java SDK [1, с. 102-110, 130-137]	2
29.	Фреймовий та віконний інтерфейси користувача [5, с. 61-69; https://uk.wikibooks.org/wiki/Освоюємо_Java/Графічний_інтерфейс_користувача]	4
30.	Реалізація легких компонентів на прикладі SWING [5 с. 8-18, 61-69; https://uk.wikibooks.org/wiki/Освоюємо_Java/Менеджери_розташування]	4
31.	Раціональний уніфікований процес (Rational Unified Process) [Додаткова 7, с. 280-289; https://uk.wikipedia.org/wiki/Rational_Unified_Process]	4
32.	Моделювання з використанням UML [Додаткова 2, с. 599-602; Додаткова 5, с. 34-197; Додаткова 7, 204-246]	4
33.	Засоби автоматизації моделювання фірми IBM [Додаткова 7, с. 443-456; https://uk.wikipedia.org/wiki/Автоматизація_процесу_програмування , http://posibniki.com.ua/post-avtomatizaciya-analizu--ta-proektuvannya--za-dopomogoyu-ibm-rational-rose , http://easy-code.com.ua/2011/05/kerovana-modellyu-rozrobka-nastupnogo-pokolinnya-ibm-rational/]	4
34.	Тестування з використанням JUnit. Відмінності версій [Додаткова 1, с.751-765; Додаткова 4, с.91-124; https://uk.wikipedia.org/wiki/Junit , https://habrahabr.ru/post/120101/]	4
35.	Установка і використання сервера постійної інтеграції [6, с. 368-402; https://uk.wikipedia.org/wiki/Неперервна_інтеграція]	4
36.	Засоби автоматизації управління проектами [6, с. 562-597; https://uk.wikipedia.org/wiki/Управління_проектами , https://uk.wikipedia.org/wiki/Розробка_програмного_забезпечення]	4
37.	Засоби управління ресурсами, ризиками та конфігураціями [6, с. 562-597, https://uk.wikipedia.org/wiki/Управління_розробкою_програмного_забезпечення , https://uk.wikipedia.org/wiki/DSDM]	4
38.	Робота команди з гнучкою методологією [6, с. 562-597, https://uk.wikipedia.org/wiki/Гнучка_розробка_програмного_забезпечення]	4
39.	Разом	134

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

- Студент має бути присутнім на кожному занятті.
- Студент може бути відсутнім на занятті, якщо здав відповідну лабораторну роботу або з поважної причини: хвороба, тощо.

- Під час заняття студенти можуть розмовляти між собою тільки з дозволу викладача.
- В разі потреби скористатися телефоном студент може тихенько вийти з класу без дозволу викладача і тихенько повернутися по завершенні розмови по телефону.
- В разі потреби студент може скористатися засобами зв'язку для пошуку інформації в інтернеті, тощо.
- На одному занятті захищається тільки одна лабораторна робота. В разі бажання студента здати ще одну роботу, студент має стати в кінець черги. За одне заняття можна здати лише 2 роботи.
- В разі потреби можна захищати роботу в режимі онлайн, для цього викладачем буде виділений окремий час.
- Кожна лабораторна робота має свій термін здачі. Своєчасна здача роботи заохочується додатковими балами.
- Перескладання лабораторної роботи можливе на наступному занятті.
- Здача чужої роботи карається штрафом 8 балів.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

На першому занятті студенти знайомляться з рейтинговою системою оцінювання, яка ґрунтується на Положенні про систему оцінювання результатів навчання https://document.kpi.ua/files/2020_1-273.pdf.

Рейтинг студента складається з балів, які він/вона отримує за лабораторні роботи (R1), за модульні контрольні роботи (R2) та за іспит (R3):

$$R_s = R_1 + R_2 + R_3 = 100 \text{ балів}$$

Поточний контроль: оцінка виконання лабораторних робіт. Кожна лабораторна робота складається з 4 частин: протокол (1 бал), роздруківка програми або файл з текстом сирцевого коду (1 бал), документування програми за допомогою інтегрованого середовища розробки (1 бал), захист лабораторної роботи (1 бал). Своєчасна здача окремих частин роботи заохочується додатковим балом. В результаті максимальна оцінка лабораторної роботи може бути 8 балів.

Календарний контроль: модульні контрольні роботи (МКР) провадяться двічі на семестр як моніторинг поточного стану виконання вимог силабусу. За виконання МКР студент може отримати до 4 балів за кожну контрольну роботу.

Семестровий контроль: іспит. За іспит студент може отримати до 20 балів.

В результаті максимальний рейтинг складається з трьох вказаних частин і дорівнює:

$$9 \text{ лабораторних робіт} \times 8 \text{ балів} = 72 \text{ бали}$$

$$2 \text{ модульні контрольні роботи} \times 4 \text{ бали} = 8 \text{ балів}$$

$$\text{Іспит} = 20 \text{ балів}$$

$$\text{Рейтинг} = 72 + 8 + 20 = 100 \text{ балів}$$

Умови допуску до семестрового контролю: мінімальна позитивна оцінка за індивідуальне завдання /зарахування усіх лабораторних робіт/ семестровий рейтинг 60 балів.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

- Зарахування сертифікатів проходження дистанційних чи онлайн курсів за відповідною тематикою не передбачається.
- Перелік питань, які виносяться на семестровий контроль:
3 семестр

1. Програмна інженерія. Визначення. Історія.
2. Які умови мають бути виконані для ефективного застосування *Flyweight*.
3. Реалізувати шаблон *Builder*.
4. XML. Призначення. Структура. Приклад.
5. Шаблон *Adapter*.
6. Шаблон *Abstract Factory* на базі *Prototype*.
7. Правила яким має відповідати *well-formed XML*-документ.
8. Двосторонній *Adapter*.
9. Реалізувати шаблон *Abstract Factory* на базі *Factory Method*.
10. Простори імен XML. Призначення. Приклад використання.
11. Шаблон *Bridge*.
12. Шаблиони прозорий *Composite*.
13. ANT. Призначення. Структура файлу сценарію. Приклад сценарію.
14. Шаблон *Flyweight*.
15. Шаблон *Command*.
16. Відмінність ANT от аналогів. Алгоритм створення нових задач для ANT.
17. Шаблон *Flyweight*.
18. Шаблиони безпечний *Composite* та внутрішній *Iterator* для обходу "в глибину" ієрархічних структур на його основі.
19. Життєвий цикл ПЗ. Стандарти ЖЦПЗ.
20. Шаблон *Flyweight*. Призначення, мотивація, структура, учасники. *Flyweight*-об'єкти що поділяються на неподільні.
21. Шаблиони безпечний *Composite* та *Iterator*-курсор для обходу "в ширину" ієрархічних структур на його основі.
22. Модель і методологія ЖЦПЗ. Коротка характеристика основних моделей ЖЦ.
23. Шаблон *Facade*.
24. Шаблиони безпечний *Composite* та *Visitor* для представлення та обчислення арифметичних виразів.
25. Основні процеси ЖЦПЗ.
26. Шаблон *Proxy*.
27. Шаблон *Strategy* для алгоритмів сортування.
28. Каскадна модель ЖЦПЗ.
29. Шаблон *Composite*.
30. Шаблон *Chain of Responsibility*.
31. Ітеративна / інкрементна модель ЖЦПЗ.
32. Шаблон *Decorator*.
33. Шаблон *Observer* з менеджером оновлень.
34. Спіральна модель Боэма.
35. Шаблон *Observer*.
36. Шаблон *State* зі зміною станів у контексті.
37. Гнучкі методології розробки ПЗ. Загальні особливості.
38. Шаблон *Mediator*.
39. Шаблон змінний *Adapter*.
40. Вимоги до програмного забезпечення. Визначення.
41. Шаблон *Iterator*.
42. Шаблон *Flyweight* з подільними та неподільними об'єктами.
43. Функціональні та нефункціональні вимоги. Визначення. Приклад.
44. Шаблон *Visitor*. Призначення, структура, учасники. В яких випадках використання *Visitor* недоречно. Одинарна та подвійна диспетчеризація.
45. Шаблон *Bridge*.
46. Користувацькі вимоги.

47. Шаблон *Strategy*. Призначення, структура, учасники. Результати застосування та альтернативи.
48. Шаблон *Builder*. Забезпечити існування лише одного екземпляра кожного конкретного білдера.
49. Системні вимоги.
50. Шаблон *Command*. Призначення, структура, учасники. Порівняти *Command* та *Strategy*. Спільні та відмінні риси.
51. Шаблон *Abstract Factory* на базі *Prototype*.
52. Документування вимог.
53. Шаблон *Chain of Responsibility*. Призначення, структура, учасники. Результати використання.
54. Шаблон *Abstract Factory* на базі *Factory Method*. Як забезпечити існування лише одного екземпляра кожної конкретної фабрики.
55. Розробка вимог. Модель, учасники, управління та контроль.
56. Шаблон *Observer*. Призначення, структура, учасники. Неочікувані оновлення. Причини та способи нейтралізації.
57. Шаблиони прозорий *Composite* та *Interpreter* для представлення та обчислення арифметичних виразів.
58. Формування та аналіз вимог.
59. Шаблон *Strategy*. Призначення, структура, учасники. Наявність якого механізму в мові програмування знімає необхідність у *Strategy*.
60. Шаблон *Command*. Як забезпечити можливість протоколювання та "відкату" команд.
61. Атестація вимог.
62. Шаблон *Visitor*. Призначення, структура, учасники. Порівняти *Visitor* та внутрішній *Iterator*.
63. Шаблиони безпечний *Composite* та внутрішній *Iterator* для обходу "в глибину" ієрархічних структур на його основі.
64. Управління вимогами.
65. Шаблон *Iterator*. Призначення, структура, учасники. Яким чином клієнт не знаючи конкретного агрегату створює ітератор, що здатний працювати із агрегатом.
66. Шаблиони безпечний *Composite* та *Iterator*-курсор для обходу "в ширину" ієрархічних структур на його основі.
67. UML. Призначення. Історія створення.
68. Шаблон *Memento*. Призначення, структура, учасники. Порівняти з альтернативними рішеннями.
69. Шаблиони безпечний *Composite* та *Visitor* для представлення та обчислення арифметичних виразів.
70. Види UML-діаграм з короткою характеристикою кожної. Приклади.
71. Шаблон *State*. Призначення, структура, учасники.
72. Шаблон *Strategy* для алгоритмів сортування. Як забезпечити незалежність реалізації від класів агрегатів та елементів.
73. Діаграма класів. Призначення, Нотація. Приклад.
74. Шаблон *Template Method*. Призначення, структура, учасники. Результати використання. З чого складається шаблонний метод..
75. Шаблон *Chain of Responsibility*. Як забезпечити можливість видалення обробника та зміни пріоритету обробника шляхом його переміщення в ланцюжку.
76. Відношення асоціації, агрегації та композиції на діаграмі класів. Приклад.
77. Шаблон *Interpreter*. Призначення, структура, учасники. Результати використання і альтернативи. Порівняти з *Composite*.
78. Шаблон *Observer* з менеджером оновлень.
79. Відношення реалізації, генералізації та залежності на діаграмі класів. Приклад.

80. Шаблон Singleton. Призначення, структура, учасники. Результати використання. Порівняти Singleton та клас з статичними членами.
81. Шаблон State зі зміною станів у контексті. Як забезпечити існування лише одного екземпляра кожного конкретного стейта.
82. Ролі. Мультиплікатори. Стереотипи. Приклад.
83. Шаблиони Prototype та Factory Method. Призначення, структура, учасники та результати. Порівняти їх застосування для створення об'єкту без інформації про його клас.
84. Шаблон змінний Adapter.
85. Діаграма активностей. Призначення. Нотація. Приклад.
86. Шаблон Abstract Factory. Призначення, структура, учасники та результати. Порівняти різні реалізації AF..
87. Шаблон Flyweight з подільними та неподільними об'єктами. Забезпечити існування лише одного екземпляра фабрики.
88. Діаграма послідовності. Призначення. Нотація. Приклад.
89. Шаблон Builder. Призначення, структура, учасники та результати. Ролі учасників шаблону. Чи можливо виключення учасника Builder в разі наявності лише одного ConcreteBuilder? Обґрунтувати.
90. Шаблон Bridge.
91. Діаграма кооперації. Призначення. Нотація. Приклад.
92. Шаблиони GRASP. Коротка характеристика кожного шаблону.
93. Шаблон Builder. Як забезпечити існування лише одного екземпляра кожного конкретного білдера.
94. Діаграма прецедентів. Призначення. Нотація. Приклад.
95. Шаблон MVC. Призначення, структура, учасники та результати. Модифікації.
96. Реалізувати шаблон Abstract Factory на базі Prototype.

Робочу програму навчальної дисципліни (силабус):

Складено доцентом кафедри ОТ, к.т.н., с.н.с., Антоноюком Андрієм Івановичем

Ухвалено кафедрою обчислювальної техніки (протокол № 10 від 25 травня 2022 р.)

Погоджено Методичною комісією факультету (протокол № 10 від 09.06.2022)