

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ФАХОВИЙ КОЛЕДЖ ПРОМИСЛОВОЇ АВТОМАТИКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ОДЕСЬКОГО НАЦІОНАЛЬНОГО ТЕХНОЛОГІЧНОГО УНІВЕРСИТЕТУ»

Циклова комісія комп'ютерних наук та інженерії програмного забезпечення

ЗАТВЕРДЖУЮ
директор ФКПАІТ ОНТУ
підписано Ольга ЄПУР
30.08.2023 року

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ВІЗУАЛЬНОГО ПРОГРАМУВАННЯ

обов'язкова

Освітньо-професійна програма Інженерія програмного забезпечення

Код та найменування спеціальності 121 «Інженерія програмного
забезпечення»

Шифр та найменування галузі знань 12 «Інформаційні технології»

Мова навчання українська

Розроблено та забезпечується: цикловою комісією Комп'ютерних наук та інженерії програмного забезпечення ВСП «Фаховий коледж промислової автоматики та інформаційних технологій Одеського національного технологічного університету»

Розробники:

Наталія ШВЕЦЬ, викладач вищої кваліфікаційної категорії
ФКПАІТ ОНТУ

Розглянуто та схвалено на засіданні циклової комісії Комп'ютерних наук та інженерії програмного забезпечення

Протокол №01 від 28.08.2023 р.

Голова циклової комісії

підписано
(підпис)

Тетяна КОСТИРЕНКО
(Власне ім'я, ПРІЗВИЩЕ)

Гарант освітньо-професійної програми

підписано
(підпис)

Тетяна КОСТИРЕНКО
(Власне ім'я, ПРІЗВИЩЕ)

Розглянуто та схвалено Методичною радою ФКПАІТ ОНТУ

Протокол №1 від 29.08.2023 р.

Голова Методичної ради ФКПАІТ ОНТУ підписано Вікторія ОКСАНІЧЕНКО
(підпис)

1. Пояснювальна записка

Вступ

Програма вивчення навчальної дисципліни «Інструментальні засоби візуального програмування» складена відповідно до Стандарту фахової передвищої освіти із спеціальності 121 «Інженерія програмного забезпечення» та освітньо-професійної програми «Інженерія програмного забезпечення» підготовки фахових молодших бакалаврів.

Предмет вивчення

Предметом вивчення дисципліни «Інструментальні засоби візуального програмування» є теоретичні відомості про візуальний підхід до програмування графічного інтерфейсу користувача засобами бібліотеки JavaFX мови програмування Java.

Міждисциплінарні зв'язки

Попередні – «Інженерна і комп'ютерна графіка», «Об'єктно-орієнтоване програмування», послідовні — «Проектний практикум».

Мета та завдання навчальної дисципліни

Метою викладання дисципліни «Інструментальні засоби візуального програмування» є формування у здобувачів освіти системи теоретичних і практичних знань, вмінь і навичок у проектуванні інтерфейсу користувача та його програмної реалізації мовою Java засобами графічної підсистеми JavaFX.

Основними завданнями вивчення дисципліни «Інструментальні засоби візуального програмування» є:

1. Вивчення основних принципів і понять, покладених в основу розробки програм з графічним інтерфейсом:
 - ознайомлення здобувачів освіти з базовими поняттями, такими як директивні програми та програми, що управляються подіями;
 - розуміння архітектурного шаблону MVC, покладеного в основу проектування і розробки програмного забезпечення з гнучким графічним інтерфейсом користувача;
 - надання знань про програмування багатовіконних застосунків, елементи графічного інтерфейсу і способи обробки подій.
2. Навчання підходам щодо вирішення проблем в роботі програм:
 - отримання практичних навичок роботи з відлагоджувачем середовища розробки програм для тестування програмного забезпечення;
 - ознайомлення з класифікацією помилкових ситуацій;
 - надання знань про способи передбачення і обробки виняткових ситуацій.
3. Надання знань про засоби зберігання інформації, введеної користувачем:
 - ознайомлення з застосуванням файлів даних;

- вивчення інфраструктури колекцій як інструменту для зберігання і управління набором однотипних даних.
- 4. Робота з програмним забезпеченням:
 - проєктування та розробка програмного забезпечення із застосуванням об'єктно-орієнтованої парадигми програмування;
 - застосування інструментальних засобів та інтегрованих середовищ для створення програмного забезпечення;
 - способи перенесення застосунків на різні обчислювальні платформи та підходи щодо запуску програм.
- 5. Використання засобів розв'язання задач:
 - здатність застосовувати знання у практичних ситуаціях.
 - самостійне створення програмних рішень для конкретних сценаріїв.
- 6. Акцент на критичному мисленні та оптимізації:
 - розвиток навичок готувати і приймати рішення, виробляти у оцінку нових ідей, управляти процесами, проєктами;
 - здатність вибрати оптимальні підходи щодо стилю візуального програмування для конкретних задач.
- 7. Підготовка до роботи в команді:
 - використання веб-сервісів для спільної розробки програмного забезпечення;
 - розвиток навичок комунікації.

Ці завдання спрямовані на те, щоб здобувачі освіти отримали досвід у застосуванні візуального програмування для розробки програм з графічним інтерфейсом користувача, зрозуміли його принципи і можливості для створення програмних систем та їх супроводження.

Компетентності та результати навчання

У результаті вивчення навчальної дисципліни «Інструментальні засоби візуального програмування» здобувач освіти отримує наступні програмні компетентності та програмні результати навчання, які визначені в Стандарті фахової передвищої освіти із спеціальності 121 «Інженерія програмного забезпечення»

(<https://mon.gov.ua/storage/app/media/Fakhova%20peredvyshcha%20osvita/Zatverdzeni.standarty/2021/09/21/121-inzh.prohr.zabezp.21.09.docx>) та освітньо-професійній програмі «Інженерія програмного забезпечення» (<https://dev-kpa.fakel.com.ua/storage/uploads/4t0YvRV8MBZ1IXWV9i190ZBGF5H7rglXYysWLzuH.pdf>) підготовки фахових молодших бакалаврів.

Загальні компетентності:

- ЗК05.** Знання та розуміння предметної області та розуміння професійної діяльності.
- ЗК06.** Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- ЗК07.** Здатність застосовувати знання у практичних ситуаціях.

Спеціальні (фахові, предметні) компетентності:

СК02. Здатність накопичувати знання в галузі інформаційних технологій та усвідомлювати важливість навчання протягом усього життя.

СК03. Здатність застосовувати теоретичні та емпіричні знання для розроблення, тестування, впровадження та супроводу програмного забезпечення.

СК07. Здатність розробляти модулі і компоненти програмного забезпечення за допомогою типових алгоритмів та інструментів.

СК08. Здатність забезпечувати інформаційну та функціональну безпеку програмного забезпечення.

СК09. Здатність вибирати та використовувати ефективні інструментальні засоби розробки програмного продукту.

СК10. Здатність реалізовувати всі етапи життєвого циклу програмного забезпечення.

СК12. Здатність проєктувати, розробляти та обслуговувати веб-застосунки з динамічним контентом, використовуючи веб-технології, технології комп'ютерної графіки та анімації.

Програмні результати навчання:

РН02. Систематизувати та узагальнювати інформацію про підходи, методи та засоби розробки і супроводу програмного забезпечення.

РН03. Застосовувати спеціалізовані емпіричні та теоретичні знання у сфері інженерії програмного забезпечення.

РН05. Розробляти та супроводжувати програмне забезпечення.

РН07. Застосовувати стандарти, специфікації в процесах життєвого циклу програмного забезпечення.

РН11. Обирати інструментальні засоби, ефективні методи та здійснювати тестування програмних систем.

РН14. Розуміти предметну область, застосовувати знання у професійній діяльності.

РН15. Аналізувати та узагальнювати необхідну інформацію з різних джерел та ресурсів для розв'язання професійних задач з урахуванням сучасних досягнень інформаційних технологій.

РН17. Розробляти застосунки, використовуючи сучасні веб-технології.

2. Інформаційний обсяг навчальної дисципліни

2.1 Тематичний план

Таблиця 2.1 – Тематичний план навчальної дисципліни

№ з/п	Назва змістових модулів і тем
Змістовий модуль 1: Графічна підсистема мови Java. Інструментальні засоби розробки JavaFX-програм	
Тема 1	Розвиток графічної підсистеми мови Java
Тема 2	Інструментальні засоби розробки JavaFX-програм
Тема 3	Налаштування інтегрованого середовища розробки для створення JavaFX-програм
Тема 4	Розробка JavaFX-програм за допомогою прикладного програмного інтерфейсу графічної підсистеми
Тема 5	Розробка JavaFX-програм за допомогою мови FXML
Тема 6	Структура програми, створеної за допомогою мови FXML. Ідентифікатори компонентів. Клас контролера
Змістовий модуль 2: Управління вікном програми. Завершення роботи JavaFX-програм. Діалогові вікна. Архітектура MVC. Обробка подій	
Тема 7	Здійснення доступу до головного вікна програми
Тема 8	Ініціалізація роботи програми з графічним інтерфейсом
Тема 9	Отримання програмою вхідних даних за допомогою аргументів командного рядка. Завершення роботи JavaFX-програм. Виконання тривалих операцій
Тема 10	Створення, відображення, приховування вікна. Стили вікон
Тема 11	Зміна кольору фону, використання зображення в якості фону вікна. Зміна та отримання розмірів вікна
Тема 12	Зміна піктограми в заголовку вікна. Закриття вікна із програми. Діалогові вікна. Клас Alert
Тема 13	Архітектура MVC як інструмент побудови гнучкого дизайну програми. Обробка подій
Разом: 120 годин	

2.2 Зміст дисципліни

Змістовий модуль 1: Графічні підсистеми мови Java. Інструментальні засоби для розробки JavaFX-програм

Тема 1. Розвиток графічної підсистеми мови Java.

Бібліотеки AWT, Swing, JavaFX. Бібліотека класів та каркас програми. JavaFX: структура програми, методи життєвого циклу програми. Потік виконання застосунку.

Тема 2. Інструментальні засоби розробки JavaFX-програм.

Встановлення JDK. Встановлення бібліотеки JavaFX. Створення і запуск JavaFX-застосунку з командного рядка

Тема 3. Налаштування інтегрованого середовища розробки для створення JavaFX-програм.

Особливості налаштування популярних середовища розробки для роботи з JavaFX.

Тема 4. Розробка JavaFX-програм за допомогою прикладного програмного інтерфейсу графічної підсистеми.

Створення програм, які використовують прикладний програмний інтерфейс каркасу JavaFX. Застосування модуля `module-info.java`, використання опції `--add-modules`.

Тема 5. Розробка JavaFX-програм за допомогою мови FXML.

Послідовність створення JavaFX-програм, які використовують мову FXML. Робота з програмою Scene Builder.

Тема 6. Структура програми, створеної за допомогою мови FXML. Ідентифікатори компонентів. Клас контролера.

Клас `Main`. Призначення файлу `.css`? Структура файлу `.fxml`. Унікальні ідентифікатори компонентів.

Змістовий модуль 2: Управління вікном програми. Завершення роботи JavaFX-програм. Діалогові вікна. Архітектура MVC. Обробка подій

Тема 7. Здійснення доступу до головного вікна програми.

Доступ до головного вікна програми всередині класу контролера. Доступ до головного вікна програми через обробник події. Доступ до головного вікна програми через посилання на головне вікно в статичному полі класу `Main`.

Тема 8. Ініціалізація роботи програми з графічним інтерфейсом.

Метод `initialize()` та інтерфейс `Initializable`. Оператор `::`. Методи `init()` і `stop()`.

Тема 9. Отримання програмою вхідних даних за допомогою аргументів командного рядка. Завершення роботи JavaFX-програм. Виконання тривалих операцій.

Передача вхідних даних програмам з графічним інтерфейсом через масив `String args[]`. Способи і особливості завершення JavaFX-програм. Проблеми з виконанням тривалих операцій. Запуск тривалої операції в окремому потоці.

Тема 10. Створення, відображення, приховування вікна. Стилі вікон.

Робота з вікном: створення, відображення, приховування, завдання стилю.

Тема 11. Зміна кольору фону, використання зображення в якості фону вікна. Зміна та отримання розмірів вікна

Управління фоном контейнера і розмірами вікна.

Тема 12. Зміна піктограми в заголовку вікна. Закриття вікна із програми. Діалогові вікна. Клас Alert.

Зміна піктограми вікна. Способи закриття вікна програмним шляхом, особливості. Діалогові вікна: модальні, немодальні. Вікна класу Alert.

Тема 13. Архітектура MVC. Обробка подій.

Архітектура MVC як інструмент побудови гнучкого дизайну програми. Обробка подій. Реалізація обробки подій: визначення методу handle() інтерфейсу EventHandler, використання лямбда-виразів, оператор ::.

2.3 Перелік практичних робіт з дисципліни

Таблиця 2.2 – Тематичний план практичних робіт

№ з/п	Тема практичної роботи
1.	Створення і запуск JavaFX-програм за допомогою командного рядка. Робота з jar-архівом
2.	Створення і запуск JavaFX-програм, що використовують API JavaFX
3.	Створення і запуск JavaFX-програм, що використовують FXML та CSS
4.	Доступ до головного вікна програми через посилання на головне вікно в статичному полі класу Main
5.	Управління вікном програми
6.	Клас Scene. Розміри контейнерів та компонентів. Внутрішні відступи. Аналіз прикладів проєктів
7.	Місце розташування вузлів усередині контейнера. Управління видимістю вузла. Управління доступністю вузла. Аналіз прикладів проєктів
8.	Абсолютне позиціонування. Горизонтальне вирівнювання. Клас HBox. Аналіз прикладів проєктів
9.	Вертикальне вирівнювання. Клас VBox. Клас TitledPane: панель із заголовком. Аналіз прикладів проєктів.
10.	Клас GridPane: розміщення вузлів усередині комірок таблиці
	Всього: 40 годин

2.4 Кількість годин разом за програмою дисципліни

Розподіл дисципліни у годинах									
Курс	I		II		III		IV		Всього
Семестр	1	2	3	4	5	6	7	8	
Повний обсяг часу							120		120
Аудиторні заняття, годин							66		66
із них (кількість годин):									
лекції							26		26
практичні							40		40
Самостійна робота							54		54
Курсова робота							90		90

3. Інструменти, обладнання та програмне забезпечення, використання яких передбачає дисципліна

- 1 Пакет розробника програм мовою Java. //Oracle: [сайт]
<https://www.oracle.com/cis/java/technologies/downloads/>
- 2 Інтегроване середовище розробки IntelliJ Idea. // Jet Brains: [сайт]/
<https://www.jetbrains.com/idea/download/>
- 3 Інтегроване середовище розробки NetBeans. //NetBeans: [сайт]
<https://netbeans.apache.org/front/main/download/>
- 4 Інтегроване середовище розробки Eclipse: //Eclipse Foundation [сайт]
<https://www.eclipse.org/downloads/>
- 5 Бібліотека JavaFX SDK. //Gluon: [сайт]
<https://gluonhq.com/products/javafx/>

4. Інформаційні ресурси

Базові (основні):

- 1 Васильєв О. М. Програмування мовою Java / О. М. Васильєв - М. Тернопіль: Навчальна книга — Богдан, 2020. - 696 с.
- 2 Кунгурцев О. Б. Основи програмування мовою Java. Середовище Net Beans: навчальний посібник для студентів вищих навчальних закладів / О. Б. Кунгурцев; за ред. Т.В. Ковалюк. - Одеса, 2016.-183 с.
- 3 Kishori Sharan, Peter Späth. Learn JavaFX 17: Building User Experience and Interfaces with Java. APress, 2022. —924 с.

Додаткові:

- 1 Java The Complete Reference, 9th Edition, Herbert Schildt.: Україна книга Language: English, 2017. — 1376 с.
- 2 Core Java Volume I--Fundamentals, 11th Edition/ К/ Horstman/Україна книга. Language: English, 2019. — 864 с.

5. Форма підсумкового контролю

Екзамен (VI семестр), поточний контроль.

6. Засоби діагностики результатів навчання

Перевірка та оцінювання знань здобувачів освіти може проводитись кількома методами:

1. Оцінювання знань здобувача освіти під час практичних занять.
2. Виконання індивідуальних навчально-дослідних завдань.
3. захист практичних робіт.
4. Тестування.
5. Проведення поточно-модульного контролю.
6. Проведення екзамену.
7. захист курсової роботи.

6.1 Питання для самоконтролю

1. Графічна підсистема мови Java:
 - a. Які етапи в своєму розвитку пройшла графічна підсистема мови Java?
 - b. В чому полягає різниця між бібліотекою класів і каркасом програми?
 - c. Поясніть особливості бібліотек AWT, Swing та JavaFX?
2. Інструментальні засоби розробки JavaFX-програм:
 - a. Яке додаткове програмне забезпечення необхідно встановити для розробки Java-програм з графічним інтерфейсом, створеним за допомогою каркаса JavaFX?
 - b. Як налаштувати інтегроване середовище розробки для роботи з JavaFX SDK?
 - c. Для чого призначена програма Scene Builder і як її підключити до середовища розробки програм?
3. Розробка JavaFX-програм за допомогою прикладного програмного інтерфейсу графічної підсистеми:
 - a. В чому полягає підхід до розробки програм з графічним інтерфейсом засобами JavaFX API?
 - b. Для чого призначений модуль module-info.java в налаштуваннях IDE?
 - c. Для чого призначена опція --add-modules в налаштуваннях IDE?
4. Розробка JavaFX-програм за допомогою мови FXML:
 - a. Яку структуру мають JavaFX-програми, створені з використанням мови FXML?
 - b. Що містить файл *.fxml?
 - c. Для чого призначений клас контролера?
5. Доступ до головного вікна програми:

- a. З якою метою може бути необхідно звернутися до головного вікна програми?
 - b. Як отримати доступ до головного вікна із класу контролера?
 - c. Як можна здійснити доступ до головного вікна через обробник події?
6. Ініціалізація роботи програми з графічним інтерфейсом:
 - a. В чому полягає різниця в підході до налаштування компонентів при використанні методу `initialize()` і інтерфейсу `Initializable`?
 - b. Для чого призначений і як працює оператор `::`?
 - c. Для чого призначені методи `init()` і `stop()`?
7. Отримання програмою вхідних даних за допомогою аргументів командного рядка. Завершення роботи JavaFX-програми. Виконання тривалих операцій:
 - a. Як отримати вхідну інформацію через `String args[]`?
 - b. В чому полягає особливість завершення роботи програми засобами `System.exit()` та `Platform.exit()`?
 - c. Чому тривалі операції слід виконувати в окремому потоці програми?
8. Створення, відображення, приховування вікна. Стили вікон. Зміна кольору фону, використання зображення в якості фону вікна. Зміна та отримання розмірів вікна:
 - a. Як створити, відобразити і приховати вікно?
 - b. Якими можуть бути стилі вікна?
 - c. Як управляти фоном вікна, розмірами?
9. Закриття вікна із програми. Діалогові вікна. Клас `Alert`:
 - a. Які є способи закриття вікна програмним шляхом? В чому полягає їх особливість?
 - b. Існують модальні і немодальні діалогові вікна. Чим відрізняється їх поведінка?
 - c. Назвіть типи модальних вікон класу `Alert`.
10. Архітектура MVC. Обробка подій
 - a. В чому полягає сутність і мета архітектурного шаблону MVC?
 - b. Як працюють програми, які мають графічний інтерфейс?
 - c. Які є підходи до програмування обробників подій?