

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ПРОГРАМУВАННЯ МОВОЮ JAVA»

напрямок підготовки 6.040303 – *системний аналіз*

форма навчання: *денна*

Виписка з навчального плану

Форма навчання	Курс	Семестр	Загальний обсяг (год.)	Всього аудит. (год.)	у тому числі (год.):			Самостійна робота (год.)	Контрольні (модульні) роботи (шт.)	Програмні проекти (шт.)	Курсові проекти (роботи), (шт.)	Залік (сем.)	Екзамен (сем.)
					Лекції	Лабораторні	Практичні						
Денна	3	6	102	51		34	17	51		6		6	

1. АНОТАЦІЯ

Курс присвячено вивченню основ синтаксису мови Java, об'єктно-орієнтованому програмуванню мовою Java. Основні пакети Java-класів. Виняткові ситуації. Потоки введення-виведення. Базові класи для створення GUI-програм (AWT I Swing). Опрацювання подій. Створення власних застосувань та аплетів.

2. ЗМІСТ ПРОГРАМИ

Синтаксис мови Java.

1. Знайомство з мовою Java.

Створення та еволюція мови. Від C/C++ до Java. Девізи Java. Структура мови Java. Середовище розробки програм мовою Java. Комерційні інтегровані середовища розробки програм мовою Java.

Перша проста програма мовою Java. Написання тексту програми. Компіляція програми. Запуск програми. Проблеми. Перший аплет. Перша програма з графічним інтерфейсом.

2. Типи даних, змінні і літерали.

Прості типи. Цілочисельні типи. Дійсні типи. Символи. Логічний тип. Літерали простих типів.

Змінні. Оголошення змінної. Динамічна ініціалізація. Область доступності і час існування змінних.

Перетворення і приведення типів. Автоматичне перетворення типів. Приведення несумісних типів. Автоматичне перетворення типів у виразах.

Масиви. Одновимірні масиви. Багатовимірні масиви. Альтернативний синтаксис оголошення масивів.

3. Операції та оператори.

Операції. Арифметичні операції. Побітові операції. Операції відношення. Логічні операції. Операція присвоєння. Умовна операція «?:». Пріоритети операцій.

Оператори. Оператори вибору. Ітераційні оператори (циклу). Оператори переходу.

Об'єктно-орієнтоване програмування мовою Java.

4. Введення в об'єктно-орієнтоване програмування.

Принципи об'єктно-орієнтованого програмування. Абстракція: класи та об'єкти. Інкапсуляція. Наслідування. Поліморфізм. Спільна дія поліморфізму, інкапсуляції та наслідування.

Відображення у програмах відношень між об'єктами. Програмування відношення "є" між класами. Програмування відношення типу "має". Програмування відношення "використовує". Відношення типу "створює".

5. Перший принцип ООП: інкапсуляція в Java.

Клас як основа інкапсуляції. Визначення класу. Методи класу. Ініціалізація членів класу. Конструктори. Знищення об'єктів. Поняття статичних даних. Ключове слово `final`: незмінні дані; незмінні параметри методів.

Пакети як сховище класів. Імпортування пакетів. Імпортування статичних методів і полів. Створення пакетів. Клас `Package`.

Основи керування доступом. Доступ до класів. Доступ до членів класу за замовчуванням. Загальнодоступні члени класу. Закриті члени класу. Ключове слово `protected` як дружній доступ.

Внутрішні класи. Закриті і захищені внутрішні класи. Внутрішні класи, визначені у методах і блоках. Безіменні внутрішні класи. Статичні внутрішні класи.

6. Другий принцип ООП: наслідування.

Синтаксис наслідування. Ключове слово `super`. Ключове слово `protected` під час наслідування. Використання ключового слова `final` під час наслідування.

Абстрактні класи та інтерфейси. Визначення інтерфейсу та абстрактного класу. Реалізація інтерфейсу. Розширення інтерфейсів за допомогою наслідування. Змінні-поля інтерфейсу. Вибір між абстрактним класом та інтерфейсом.

Клас `Object`: глобальний суперклас. Метод `toString()`. Метод `equals()`. Метод `hashCode()`. Метод `clone()`.

7. Третій принцип ООП: поліморфізм.

Поліморфізм, реалізований шляхом перевантаження методів. Перевантаження методів класу. Перевантаження методів у підкласах. Перевантаження і повернення значень методами. Конструктори і поліморфізм.

Поліморфізм, реалізований шляхом наслідування. Перекривання методів. Заборона перекриття методу. Перекривання методу *equals()*. Висхідне перетворення типів. Нисхідне перетворення типів. Абстрактні класи та інтерфейси для реалізації поліморфізму. Динамічне зв'язування метод-виклик. Виклик поліморфних методів у конструкторах.

ОСНОВНА ЛІТЕРАТУРА

1. *Копитко М.Ф., Іванків К.С.* Основи програмування мовою Java: Тексти лекцій. – Львів: Видавничий центр ЛНУ ім.Івана Франка, 2002. – 83 с.
2. *Бернакевич І.Є., Вагін П.П.* Програмування мовою Java: використання фундаментальних класів: Тексти лекцій. – Львів: Видавничий центр ЛНУ ім.Івана Франка, 2002. – 76 с.
3. *Морган М.* Java 2. Руководство разработчика. – М.: Издательский дом "Вильямс", 2000. – 720 с.
4. *Ноутон П., Шилдт Г.* Java 2. - СПб: БХВ-Петербург, 2000. – 1072 с.
5. *Эккель Б.* Философия Java. Библиотека программиста.- СПб:Питер, 2009. – 640с.
6. *Хорстманн К.С., Корнелл Г.* Java 2. Библиотека профессионала. Т. 1,2. – М., 2007.-896 с.
7. <http://java.sun.com/>

ДОДАТКОВА ЛІТЕРАТУРА

8. *К.Арнольд, Дж.Гослинг* Язык программирования Java. - СПб: Питер, 1997
9. *П.Нотон* Java. Справочное руководство.- М.: Восточная книжная Компания, 1996
10. *Н.И.Смирнов* Java 2: Учебное пособие - М.: "Три Л", 2000
11. *В.О.Сафонов* Введение в Java-технологии.- М.: Наука, 2002
12. *М.Фаулер* Рефакторинг: улучшение существующего кода. - СПб: Символ-Плюс, 2003
13. *Джошуа Блох* Java™ Эффективное программирование.- М.: Издательство "Лори", 2002
14. *Томас М., Пател П., Хадсон А.* Секреты программирования для Internet на Java Перев. с англ. - СПб: Питер, 1997.