



Назва: «Мова програмування Java. Частина 1.»

Тривалість навчання: 52 академічні години

Тип: лекції, лабораторні, модульні тести, сертифікаційні іспити

Короткий опис:

У курсі розглядаються всі необхідні теоретичні та практичні аспекти програмування необхідні для освоєння складнішого курсу «Технології Java SE». Кожна тема закріплюється лабораторною роботою.

Цільова аудиторія:

Студенти, які бажають освоїти нову професію, бажають перепрофілюватися на фахівців з розробки додатків мовою Java.

Цілі навчання для студентів:

- Використовувати у додатках змінні примітивних та об'єктних типів, статичні, локальні змінні.
- Використовувати в додатках основні арифметичні та логічні операції.
- Використовувати в додатках оператори розгалуження і циклу.
- Розробляти та використовувати в додатках класи, (зокрема абстрактні та внутрішні), інтерфейси, пакети, перерахування (enums)
- Використовувати у програмах Java API
- Розробляти у додатках методи: конструктори, статичні та нестатичні, з примітивними та об'єктними аргументами, перевантажені



- Розробляти в додатках ієрархію класів на основі композиції та успадкування з використанням модифікаторів доступу
- Використовувати в додатках концепцію поліморфізму, перевизначати методи суперкласів та реалізовувати методи інтерфейсів
- Розробляти програми за допомогою перевизначення методів суперкласів
- Розробляти програми за допомогою механізму обробки винятків (exceptions) та тверджень (assertions)
- Розробляти програми за допомогою технології обробки текстових даних
- Розробляти програми за допомогою введення та виведення даних
- Розробляти додатки за допомогою управління потоками (threads).
- Розробляти програми з використанням узагальнених (generic) класів та узагальнених методів.
- Розробляти програми за допомогою стандартних класів колекцій та карт.

Теми що розглядаються:

1. Базові принципи Java

1.1. Базові принципи Java

1.2. Будова простих Java-додатків та інструментальні засоби їх створення

1.3. Класифікація типів даних

1.4. Примітивні типи та операції з ними

1.5. Керуючі конструкції Java





1.6. Масиви

2. Об'єктноорієнтоване програмування Java

2.1. Класи та об'єкти

2.2. Методи

2.3. Конструктори

2.4. Статичні елементи

2.5. Секції ініціалізації

2.6. Пакети

2.7. Успадкування та поліморфізм

2.8. Абстрактні класи та інтерфейси

2.9. Робота з рядками

2.10. Винятки та діагностичні твердження

2.11. Винятки та діагностичні твердження

2.12. Вкладені класи

2.13. Перерахування

2.14. Узагальнене програмування

2.15. Колекції, відображення та ітератори

2.16. Дозвіл навантаження

2.17. Багатопоточність

2.18. Важливі сервісні класи

2.19. Основні прийоми ОВП





Назва: «Мова програмування Java. Частина 2.»

Тривалість навчання: 44 академічні години

Тип: Лекції, лабораторні, модульні тести.

Короткий опис:

У курсі розглядаються всі необхідні теоретичні та практичні аспекти розробки додатків на Java, кожен із яких закріплюється лабораторною роботою. Курс передбачає виконання навчального проєкту.

Цільова аудиторія:

Студенти, які бажають освоїти нову професію, бажають перепрофілюватися на фахівців з розробки додатків мовою Java

Цілі навчання для студентів:

- Використовувати компонентну модель у програмах JavaBeans.
- Використовувати у програмах концепції слабкої. взаємозв'язку та сильної ув'язки (loose coupling and high cohesion).
- Використовувати механізм збору сміття у програмах (garbage collector).
- Використовувати у програмах механізм Reflection.
- Використовувати класи пакета в програмах java.util:Formatter, Currency, Locale, класи Date/Time API
- Розробляти програми за допомогою архітектурних шаблонів.



- Розробляти програми за допомогою шаблонів проектування (design patterns).
- Використовувати стандартні SQL-запити у програмах баз даних через JDBC.
- Запускати Java-програми.
- Реалізувати моніторинг та налаштування JVM.
- Використовувати у програмах анотації.
- Формувати JavaDoc документацію.
- Використовувати стандартну бібліотеку в програмах протоколювання.
- Компетенції в SQL.
- Знати основи реляційних баз даних.
- Знати основи мови структурованих запитів SQL.
- Компетенції в Hardware Basic.
- Знати способи представлення даних на комп'ютері.
- Знати основи архітектури комп'ютера.
- Знати основи роботи операційних систем.
- Компетенції в XML.
- Створювати XML-код для опису довільних даних.
- Вміти шукати та отримувати необхідну інформацію у XML-файлах.
- Використовувати простір імен (namespaces) під час роботи з XML.
- Використовувати у програмах DOM.
- Використовувати базові типи UML-діаграм.



- Знати основи проєктного менеджменту.

Теми що розглядаються:

1 Технології Java SE.

1.1 Потоки введення-виведення. Робота з файлами.

1.1.1 Поняття потоку введення-виводу.

1.1.2 Байтові та символьні потоки.

1.1.3 Основні класи потоків.

1.1.4 Принципи обробки помилок введення-виведення.

1.1.5 Блок try з автоматичним керуванням ресурсами.

1.1.6 Клас File. Операції із файлами.

1.1.7 Клас RandomAccessFile.

1.2 Серіалізація

1.2.1 Поняття серіалізації.

1.2.2 Серіалізація складних об'єктів.

1.2.3 Серіалізація та успадкування.

1.2.4 Модифікатор transient.

1.2.5 Нестандартна серіалізація.

1.2.6 Інтерфейс Externizable.

1.3 Бази даних та JDBC.

1.3.1 Поняття таблиці, ключа, зв'язку таблиць.

1.3.2 Поняття SQL.



1.3.3 Команди `create table`, `select`, `insert`, `update`, `delete`.

1.3.4 Організація JDBC; драйверний принцип.

1.3.5 Встановлення з'єднання. Класи `DriverManager` та `Connection`.

1.3.6. Обробка результатів запиту. Клас `ResultSet`.

1.3.7. Підготовлені запити.

1.3.8 Аналіз метаданих.

1.3.9 Базові можливості управління транзакціями.

1.4 Основи Swing

1.4.1 Введення у Swing.

1.4.2. Основні інтерфейсні елементи.

1.4.3 Опрацювання подій.

1.4.4 Менеджери розміщення.

1.5 Завантаження класів. Рефлексія

1.5.1 Процес завантаження класів

1.5.2 Основні завантажувачі.

1.5.3 Ієрархія завантажувачів класів

1.5.4 Основні можливості `Reflection`.

1.5.5 Клас `Class`.

1.5.6 Отримання інформації про клас.

1.5.7 динамічне завантаження класів та створення екземплярів.

1.5.8 Використання рефлексії для отримання вказівників на методи.

1.5.9. Динамічний виклик методів.



1.5.10 Використання рефлексії доступу до приватних членів.

1.6 Анотації.

1.6.1 Поняття інструкції

1.6.2 Анотація @Override.

1.6.3 Приклад написання власних анотацій.

1.7 Java 8 нові features.

1.7.1 Lambda Expressions.

1.7.2 Functional Interfaces.

1.7.3 Method References : Operator.

1.7.4 Streams

1.7.5 New Date/Time API in Java 8.

1.7.6. Static methods and default methods in interfaces.

1.8 Java – Networking.

1.8.1 TCP/IP Networking Basics.

1.8.2 Java Socket's and ServerSocket's.

1.8.3 UDP Networking Basics.

1.8.4 UDP DatagramSocket.

1.8.5 Using URLConnection and InetAddress.

1.9 Проксі-класи.

1.9.1 Клас Proxy.

1.9.2 Типові використання проксі-об'єктів.

Розробка програмного забезпечення



1.10.1 Життєвий цикл розробки ПЗ.

1.10.2 Інструменти збирання java. Основи Maven.

1.10.3. Прийоми тестування. Основи JUnit 4.

1.10.4 Основні інструкції, що підтримуються в JUnit 4.

1.10.5 Затвердження assert, що підтримується JUnit 4.

1.10.6 Створення тестових завдань у Junit.

1.11 Git Basics.

1.11.1 Creating GitHub repo.

1.11.2 Committing your changes.

1.11.3 Reverting.

1.11.4 Pushing to a remote repository.

1.11.5 Pulling from a remote repository.

1.11.6 Resolving Conflicts.

1.11.7 Branches.

1.12 Складання сміття.

1.12.1 Фундаментальні принципи збирання сміття.

1.12.2 Поняття доступного для збирання об'єкта.

1.12.3 Фіналізатори.

1.12.4 Поняття про воскресіння об'єктів.

1.12.5 М'які посилання (Weak Reference).

1.13 Основи безпеки.

1.13.1 Рівні захисту у Java.



1.13.2 Security Manager та файли повноважень.

1.13.3 Модель пісочниці.

Попередня підготовка:

Особисті якості:

Уважність, вміння зосереджуватись, логічне мислення, критичний підхід.

Досвід:

Обов'язковий інтенсивний досвід роботи з комп'ютером не менше одного року.

Робота з браузером

- Запуск ПК
- Вхід у систему з паролем.
- Запуск програм, перемикання між запущеними програмами
- Розуміння принципів роботи програм
- Завершення програм
- Встановлення та видалення програм
- Вміння самостійно розумітися на нових програмах
- Налаштування операційної системи та програм під персональні потреби

Робота з офісними програмами

- MS Word
- MS Excel
- MS PowerPoint



Робота з браузером

- Internet Explorer або Opera
- Відкриття інтернет-адреси
- Робота з посиланнями
- Робота із закладками та вікнами

Обов'язково:

- Розуміння принципів побудови та роботи алгоритмів
- Досвід програмування будь-якою мовою
- Проходження фінального іспиту з курсу

Сертифікація:

Обов'язковим для отримання сертифіката є проходження всіх модульних тестів, а також сертифікаційний іспит MainAcademy certified associate in Java Developing. Крім того, необхідна розробка та успішна здача навчального проекту.

IT-English:

Усі студенти академії мають можливість пройти додаткові заняття з англійської мови з глибоким вивченням IT-термінології. Ті, хто успішно закінчив курси Main Academy та впевнено володіють англійською мовою, мають найбільші шанси на працевлаштування в IT-сфері.

