



# LẬP TRÌNH DI ĐỘNG

---

## Bài 1: Nhập Môn



# Nội dung

---

1. Giới thiệu môn học
2. Công cụ học tập
3. Giới thiệu về ngôn ngữ java
  - Ngôn ngữ lập trình Java
  - Lập trình hướng đối tượng
  - Công cụ phát triển Java
  - Các loại chương trình Java
  - “Hello, World!” – phiên bản Java
  - Quá trình thực thi ứng dụng Java



Phần 1

# Giới thiệu môn học



# Môn học “lập trình di động”

---

- Tên môn học chính xác phải là “làm quen với lập trình cho thiết bị di động”
  - Ta chỉ làm quen một chút thôi, muốn lập trình thực sự thì cần học nhiều hơn
  - Nên có một thiết bị android nào đó để thử
- Số tín chỉ: 3 (3 lý thuyết)
  - Tại sao không có thực hành? Vì lý thuyết quá nhiều, thậm chí 3 tín chỉ vẫn chưa đủ
- Giảng viên: Trương Xuân Nam, khoa CNTT
- Email: [truongxuannam@gmail.com](mailto:truongxuannam@gmail.com)



# Tài liệu học tập

---

- Bài giảng môn học + demo + tài liệu đọc thêm
- Tài liệu tham khảo:
  - Raymond Gallardo, Scott Hommel, Sowmya Kannan, Joni Gordon, Sharon Biocca Zakhour. *The Java Tutorial: A Short Course on the Basics, 6th edition*.  
Online version: <https://docs.oracle.com/javase/tutorial/>
  - Dave MacLean, Satya Komatineni, Grant Allen. *Pro Android 5*. Apress, 2015
- Online: bài giảng, tài liệu đọc thêm, điểm số, thông báo, ... sẽ được đưa lên website <http://txnam.net>,  
mục **BÀI GIẢNG**



# Kiến thức yêu cầu / nên biết

---

- Kiến thức về lập trình C/C++
- Kiến thức về lập trình hướng đối tượng (cơ bản)
- Kiến thức về SQL (cơ bản)
- Kiến thức về XML (cơ bản)
- Kiến thức về kiến trúc máy tính (đặc biệt là của thiết bị di động)
- Kiến thức về hệ điều hành
- Đã từng sử dụng một thiết bị di động nào đó

# Đánh giá kết quả

- Điểm môn học = ĐQT x **40%** + ĐTCK x **60%**
- Điểm quá trình:
  - Điểm danh
  - Thảo luận
  - Mini project
- Điểm thi cuối kỳ:
  - Vấn đáp hoặc trắc nghiệm
  - Không có giới hạn nội dung thi





# Học môn này có lợi gì?

---

- Có kiến thức về lập trình cho thiết bị di động
- Có hiểu biết sâu sắc hơn về hoạt động của các thiết bị di động và phần mềm trên các thiết bị đó, khai thác tốt hơn các thiết bị đó
- Có khả năng viết chương trình đơn giản cho các thiết bị di động
- Có thêm lựa chọn cho đề tài làm tốt nghiệp
- Có điểm môn học và được ra trường





# Nội dung môn học (1/2)

---

- Cơ bản về ngôn ngữ lập trình Java
- Một số chủ đề chọn lọc trong lập trình Java
- Giới thiệu về lập trình di động và Android
- Activity
- Layout và các điều khiển cơ bản
- Xử lý sự kiện

*Tại sao phải học java? Lập trình android dùng java*



# Nội dung môn học (2/2)

---

- Intent, Notification và Menu
- Lưu trữ, SQLite và content provider
- Dịch vụ và Broadcast Receiver
- Xử lý nền và đa luồng
- Khai thác các dịch vụ di động và mạng
- Các chủ đề nâng cao

*Tại sao chọn android? Dễ, thông dụng, thiết bị rẻ*



Phần 2

# Công cụ học tập



# Công cụ học tập

---

- Công cụ đề xuất: eclipse
  - Miễn phí, mạnh mẽ, tương thích tốt
  - Dùng được cho cả 2 phần nội dung lập trình (java và android) trong môn học
- Một số công cụ khác
  - Netbeans (dùng cho lập trình java)
  - Android Studio (dùng cho lập trình android)
- Tất cả các công cụ trên đều cần bộ phát triển ứng dụng java: JDK (java development kit)
  - Đề xuất sử dụng phiên bản 8



Phần 3

# Giới thiệu về ngôn ngữ java

# Ngôn ngữ lập trình Java

---

- Được bắt đầu từ 1991 bởi **James Gosling**
- Phiên bản Java 1.0 phát hành ngày 23/1/1996
- Được phát triển liên tục:
  - Phiên bản 1.1: 02/1997
  - Phiên bản 2: 12/1998
  - Phiên bản 3: 05/2000
  - Phiên bản 4: 02/2002
  - Phiên bản 5: 09/2004
  - Phiên bản 6: 12/2006
  - Phiên bản 7: 07/2011
  - Phiên bản 8: 03/2014





# Ngôn ngữ lập trình Java

---

- Dựa trên ngôn ngữ C/C++, lược bỏ những chi tiết phức tạp: Đơn giản, dễ học và dễ dùng hơn
- Độc lập nền tảng (platform-independent)
- Hướng đối tượng (object-oriented)
- Bảo mật (secure)
- Mạnh mẽ (robust)
- Phân tán (distributed)
- Đa luồng (multithreaded)
- Động (dynamic)



# Lập trình hướng đối tượng

---

- Tiếng Anh: **O**bject-**O**riented **P**rogramming (OOP)
- Là một phương pháp thiết kế và lập trình
- Cách tiếp cận bài toán theo phương pháp chia nhỏ phần mềm thành các module tương tác với nhau
- Mục tiêu: Xây dựng các module (thường gọi là các class), mô phỏng tương tự các đối tượng thực ngoài đời





# Lập trình hướng đối tượng

---

- Đặc điểm của OOP:
  - Data abstraction (sự trừu tượng hóa dữ liệu)
  - Data encapsulation (sự đóng gói dữ liệu)
  - Inheritance (sự kế thừa)
  - Polymorphism (sự đa hình)
- Ưu điểm:
  - Khả năng tái sử dụng cao
  - Giảm chi phí phát triển và bảo trì sản phẩm
  - Gần gũi giữa lập trình và bài toán thực tế



# Lập trình hướng đối tượng

---

- Một số thuật ngữ cần nhớ:
  - Class (lớp)
  - Object (đối tượng)
  - Constructor (hàm tạo)
  - Destructor (hàm hủy)
  - Attribute (thuộc tính)
  - Method (phương thức)
  - Event (sự kiện)
- Các thuật ngữ này sẽ được giải thích kĩ hơn khi gặp tình huống sử dụng chúng



# Công cụ phát triển Java (1/3)

---

- Công cụ gốc: JDK (Java Development Kit)
  - Thường thì mỗi phiên bản lớn của JDK ứng với một phiên bản của ngôn ngữ Java. Ví dụ: JDK 1.7 ứng với phiên bản Java 7
  - JDK là bộ công cụ bắt buộc phải có đối với việc phát triển ứng dụng Java
  - JDK bao gồm:
    - Máy ảo Java (JVM)
    - Các thư viện lập trình
    - Một số công cụ cơ bản cho nhà phát triển
  - JRE (Java Runtime Environment): Để chạy Java Apps



# Công cụ phát triển Java (2/3)

---

- Các công cụ cơ bản, đi kèm trong JDK
  - Trình dịch: `javac`
  - Trình chạy: `java`
  - Dịch ngược: `javap`
  - Tạo tài liệu: `javadoc`
  - Gỡ rối: `jdb`
  - Chạy applet: `appletviewer`



# Công cụ phát triển Java (3/3)

---

- Các công cụ tích hợp: giúp dễ dàng hơn cho sự phát triển, thực thi, gỡ lỗi, bảo trì,... các ứng dụng Java
  - NetBeans
  - Eclipse
  - IntelliJ IDEA
  - JBuilder
  - JDeveloper
  - ...
- Chú ý: Các công cụ trên thường vẫn cần JDK như một thành phần bắt buộc



# Các loại chương trình Java

---

- **Command line application**: ứng dụng đơn giản nhất, chạy trực tiếp từ dòng lệnh
- **GUI application**: ứng dụng dùng giao diện đồ họa tương tác người sử dụng
- **Applet**: ứng dụng java nhúng trong các trang web, tương tác như một thành phần của trang
- **Servlet**: ứng dụng phía server, chạy trên các web server, xử lý các yêu cầu người dùng web
- **Add-on app (plug-in app)**: các thành phần mở rộng chức năng của những phần mềm có hỗ trợ java



# “Hello, World!” – phiên bản Java

---

```
1 public class HelloWorld {  
2     public static void main(String[] args) {  
3         System.out.println("Hello, World!");  
4     }  
5 }
```

- Viết bằng bất kì trình soạn thảo nào
- Lưu tên file là “HelloWorld.java” (bắt buộc)
- Dịch bằng lệnh: `javac HelloWorld.java`
- Chạy bằng lệnh: `java HelloWorld`



# Quá trình thực thi ứng dụng Java

---

- **Bước 1:** Mã Java được dịch thành mã trung gian (mã bytecode). Mã này được xây dựng bởi SUN, độc lập với mã máy. Các đoạn mã được lưu trong các file “.class”
- **Bước 2:** Máy ảo Java nạp các file “.class” và kiểm tra tính toàn vẹn của các file này. Thực hiện một số bước chuẩn bị như khởi tạo bộ nhớ, nạp các thư viện mặc định
- **Bước 3:** Máy ảo Java đọc và thực hiện từng lệnh một trong mã bytecode cho đến khi kết thúc ứng dụng