

NPRU

Nakhon Pathom
Rajabhat University

การเปรียบเทียบวิธีในการแปลงชุดคำสั่ง
จากภาษาจาวาเป็นคอทลิน
สำหรับการพัฒนาโมบายแอปพลิเคชัน

A Comparative method in mobile application code conversion
between Java and Kotlin

NPRU

Nakhon Pathom
Rajabhat University

NARUAPON SUWANWIJIT

Software Engineeing, Faculty of science and technology

Nakhon Pathom Rajabhat University

naruapon@webmail.npru.ac.th

วัตถุประสงค์

- เปรียบเทียบแนวทางในการปรับปรุงชุดคำสั่งของแอปพลิเคชันจากภาษาจาวาเป็นภาษาคอทลิน และทราบถึงผลลัพธ์และปัญหาในแต่ละทางเลือก

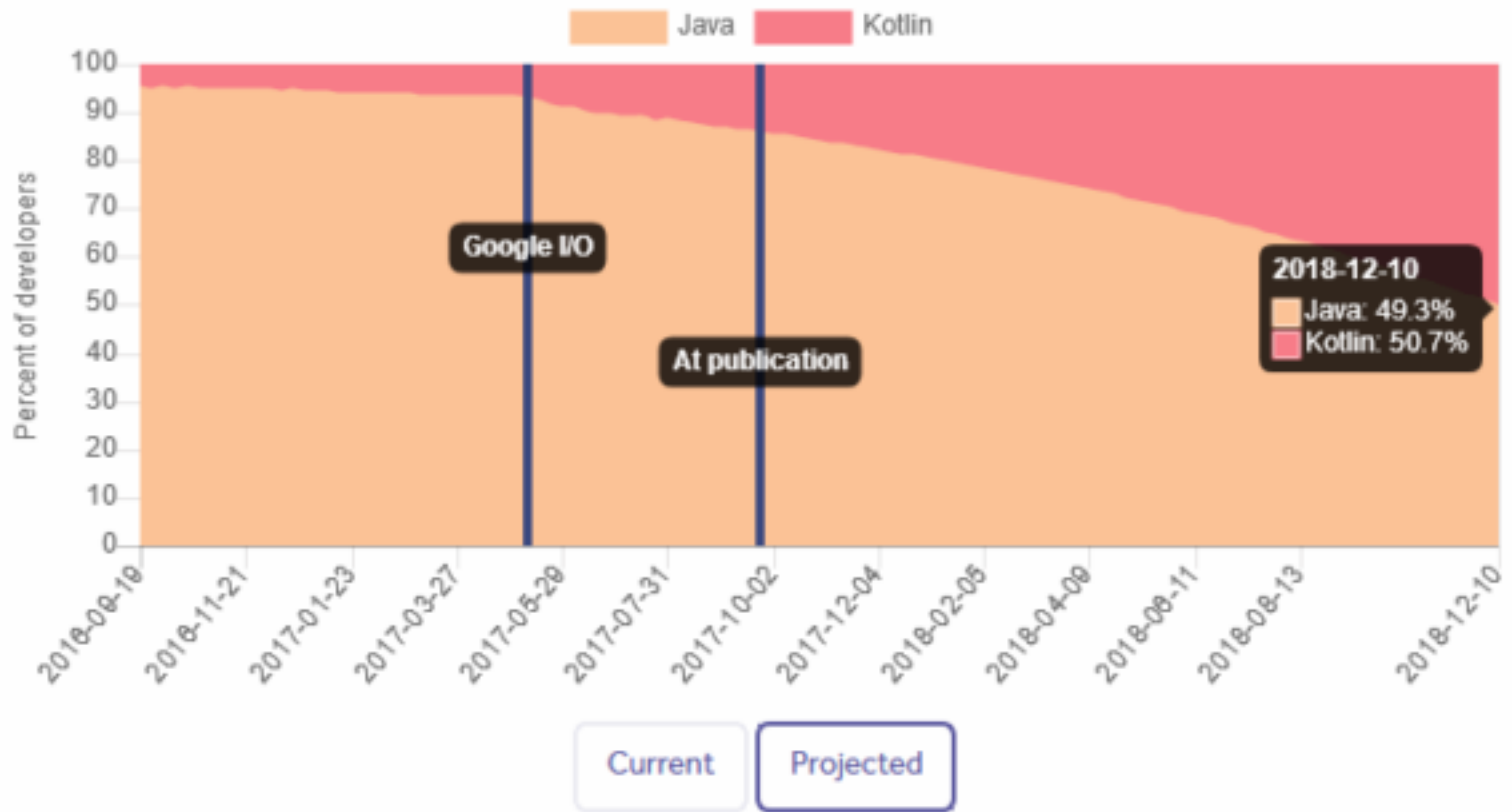
การเติบโตของระบบปฏิบัติการบนสมาร์ทโฟน 4 ปี ตั้งแต่ปี 2019 – 2022

- ระบบปฏิบัติการแอนดรอยด์เฉลี่ยอยู่ที่ 86.7%
- ระบบปฏิบัติการ iOS 13.2%

การพัฒนาแอปตรอย์แอปพลิเคชัน

- Eclipse และ ภาษาจาวา
- ปี 2014 Android Studio 1.0
- ปี Android Studio 3.0 ภาษา คอทลิน

สัดส่วนของนักพัฒนาที่ใช้ภาษาคอทลินในการพัฒนา แอปพลิเคชันในปี 2016 - 2018



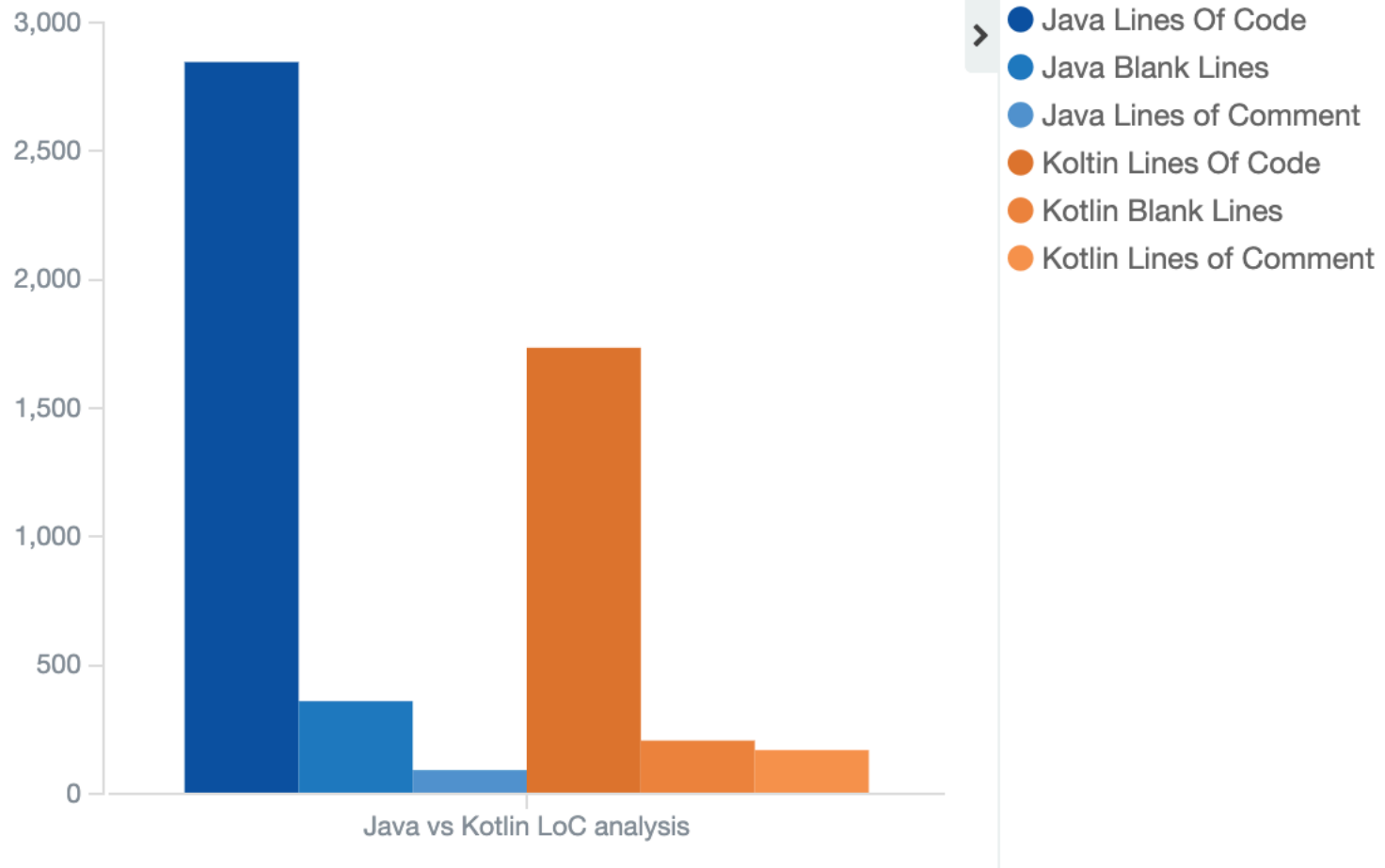
ที่มา : <https://www.bleepingcomputer.com/>

ขนาดของซอฟต์แวร์

การวัดขนาดของซอฟต์แวร์ สามารถทำได้ด้วยวิธีการต่างๆ

- การวัดจากจำนวนบรรทัด
- การวัดจากฟังก์ชันพอยต์
- จำนวนบรรทัดของชุดคำสั่ง ซึ่งบ่งบอกถึงขนาดของซอฟต์แวร์ได้ แต่ไม่นับชุดคำสั่งที่ระบบสร้างขึ้นมาให้ และพวกคอมเมนต์ต่างๆ

เปรียบเทียบจำนวน LOC ของภาษาจาวาเมื่อเทียบกับ ภาษา Kotlin



ที่มา : <https://eng.uber.com/>

Java

```
public class Product{
    String productID;
    String productName;
    double price;
    public Product(String productID, String productName, double price){
        this.productID = productID;
        this.productName = productName;
        this.price = price;
    }
    public String toString(){
        return productID + " " + productName + " , Price = " + price;
    }
}
```

Kotlin

```
public class Product(val productID:String, val productName:String, val price:
Double){
    override fun toString() = “$productID $productName , Price = $price”
}
```

งานวิจัยที่เกี่ยวข้อง

- Subham Bose, Aditi Kundu, Madhuleena Mukherjee, Madhurima Banerjee.(2018). **A comparative study : Java vs Kotlin programming in android application development.** International Journal of Advanced Research in Computer Science,” vol. 9, no. 3, 41-45.
- Dony George, Priyanka Girase, Mahesh Gupta, Prachi Gupta, Aakanksha Sharman.(2010). **Programming Language Inter-conversion.** International Journal of Computer Applications, vol. 1, no. 20, (0975-8887).

งานวิจัยที่เกี่ยวข้อง (ต่อ)

- A.A. Terekhov, C. Verhoef. (2000). **The Realities of Language Conversion**. IEEE Software, Volume: 17 , Issue: 6, 111 – 124.

A comparative study : Java vs Kotlin programming in android application development

แปลง Java เป็น Kotlin ในประเด็นดังนี้

- Extension Function
- Checked Exception
- Constructors
- Null Safety
- Lazy-Loading

Programming Language Inter-conversion.

แปลงจากภาษา C ++ ไปเป็นภาษา Java ในประเด็นดังนี้

- Pointers
- Pre-processor
- Destructors
- Friend Function
- Operator Overloading

The Realities of Language Conversion.

- การแปลงชุดคำสั่งแบบอัตโนมัติ
- COBOL to Visual Basic
- COBOL to C
- COBOL to Java
- OS/VS COBOL to VS COBOL II

Mobile Application

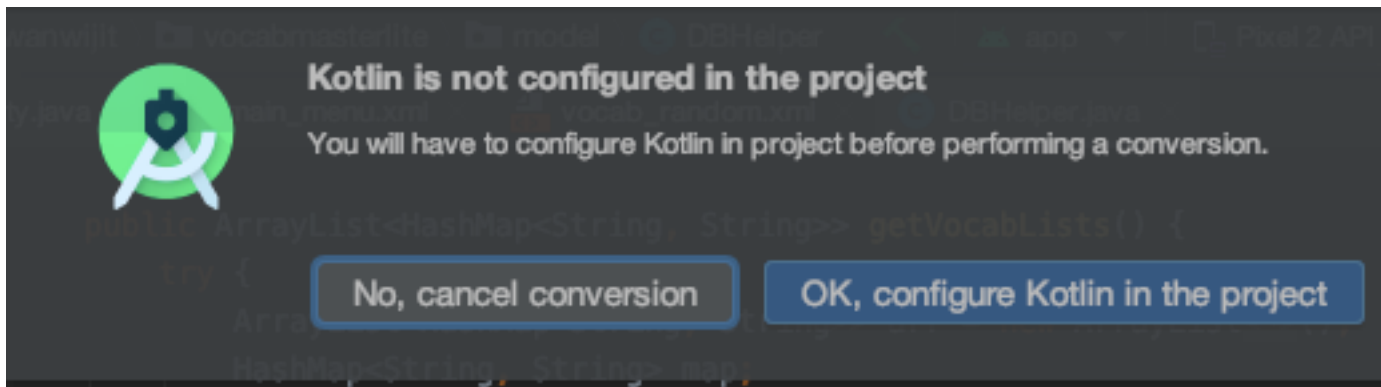
- Native application
- Hybrid application
- Cross-Platform application

การแปลงชุดคำสั่ง 3 แนวทาง

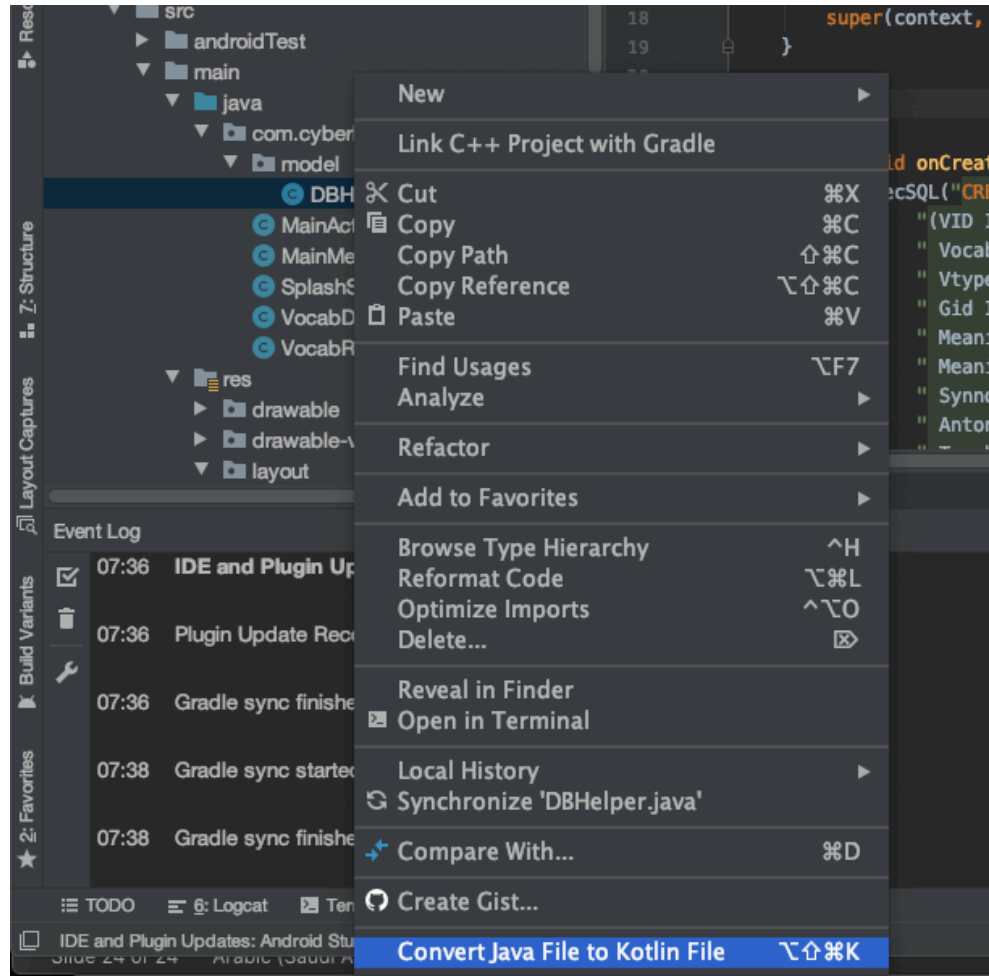
- การแปลงชุดคำสั่งโดยใช้เครื่องมือ – A1
- การแปลงชุดคำสั่งโดยการสร้างชุดคำสั่งขึ้นมาใหม่แต่ใช้โปรเจ็คเดิม – A2
- การแปลงชุดคำสั่งโดยการสร้างชุดคำสั่งขึ้นมาใหม่โดยการสร้างโปรเจ็คใหม่ – A3

การแปลงชุดคำสั่งโดยใช้เครื่องมือ – A1

- Android Studio มีเครื่องมือที่ช่วยในการแปลงจากภาษาจาวาเป็นภาษาคอทลิน เพราะว่าภาษาคอทลินได้รับการสนับสนุนจากกูเกิ้ลอย่างเป็นทางการ แต่ก่อนจะดำเนินการจำเป็นจะต้องติดตั้งปลั๊กอิน



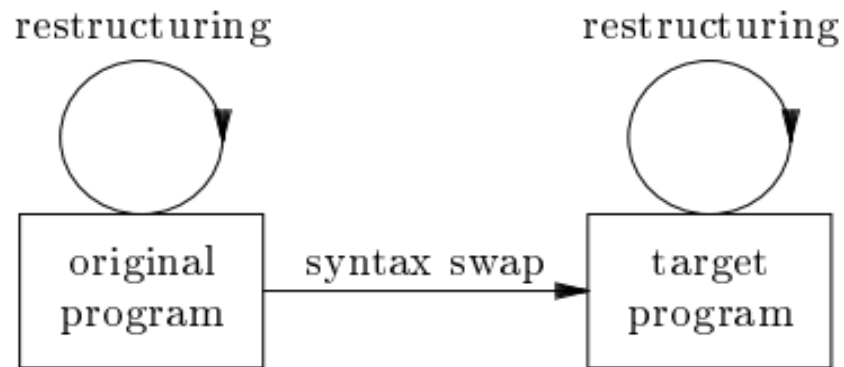
Android Studio



การแปลงชุดคำสั่งโดยการสร้างชุดคำสั่งขึ้นมาใหม่แต่ ใช้โปรเจ็คเดิม – A2

- การเลือกแปลงชุดคำสั่งขึ้นมาที่ละไฟล์ ทำการลบไฟล์ที่จะทำการแปลง และสร้างไฟล์ใหม่ขึ้นมา หลังจากทีสร้างไฟล์และเขียนชุดคำสั่งขึ้นมาใหม่

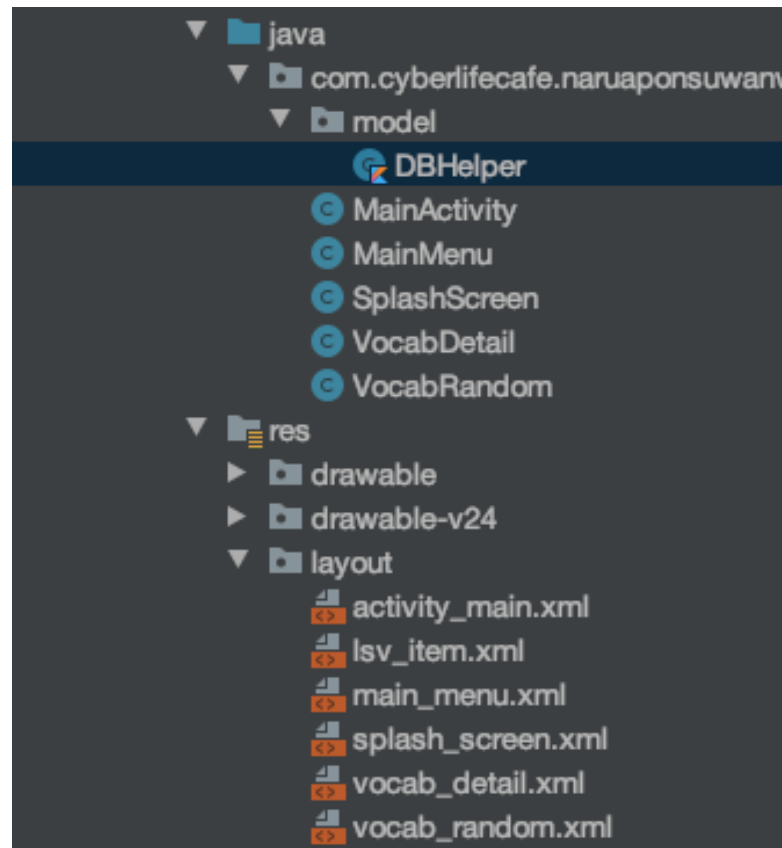
กระบวนการแปลงจากภาษาหนึ่งไปยังอีกภาษาหนึ่ง



การแปลงชุดคำสั่งโดยการสร้างชุดคำสั่งขึ้นมาใหม่โดย

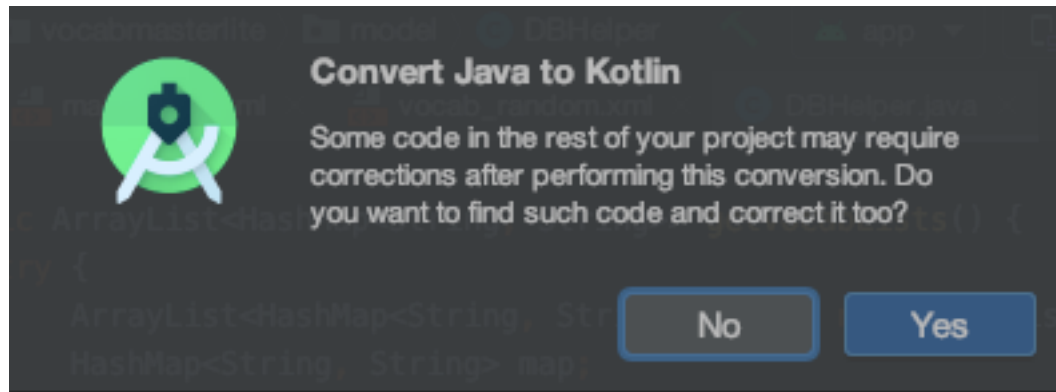
การสร้างโปรเจ็คใหม่ – A3

- สร้างไฟล์ต่างๆ ขึ้นมาใหม่จำนวนมาก ไม่เพียงแค่ไฟล์ภาษาที่นามสกุล เช่น ไฟล์ที่ใช้ในการออกแบบหน้าจอ, ไฟล์ทรัพยากรต่างๆ ของระบบ



ปัญหาที่พบ

- การแก้ไขนักพัฒนาต้องมีความรู้ทางด้านภาษาคอทลินบ้าง เพราะต้องรู้วิธีในการแก้ปัญหาที่เกิดขึ้น



ปัจจัยในการเลือกวิธีที่ใช้ในทางในการแปลงชุดคำสั่ง

แนวทาง	ระยะเวลาที่ใช้	ความรู้ทางด้านภาษาโปรแกรม
A1	น้อย หากไม่พบปัญหา	น้อย
A2	ปานกลาง	ปานกลาง
A3	มาก	มาก

สรุป

- ในแนวทางที่ A1 กับ A2 หากพบปัญหาเมื่อแปลงแล้ว ไม่จำเป็นต้องแปลงทั้งหมด
- หากต้องการประสิทธิภาพของแอปพลิเคชัน ควรใช้แนวทาง A3 งานวิจัยที่ควรศึกษาต่อไป
- การแปลงชุดคำสั่งของภาษาคอทลินเป็นภาษา Swift
 - การเลือกแปลงในส่วนที่เป็น Business logic