คู่มือโปรแกรมเมอร์

อินเตอร์พริตเตอร์ไพธอนขนาดเล็กสำหรับบราวเซอร์ที่โปรแกรมได้บนโทรศัพท์มือถือ

A Small Python Interpreter for The Programmable Browser on Mobile Phones

เนื่องจากโครงงานนี้เป็นโครงงานที่มีการทำงานบนโทรศัพท์มือถือ ฉะนั้นคู่มือโปรแกรมนี้จะเป็นการแนะนำและอธิบายการพัฒนาสำหรับโทรศัพท์มือถือ sony ericsson P800

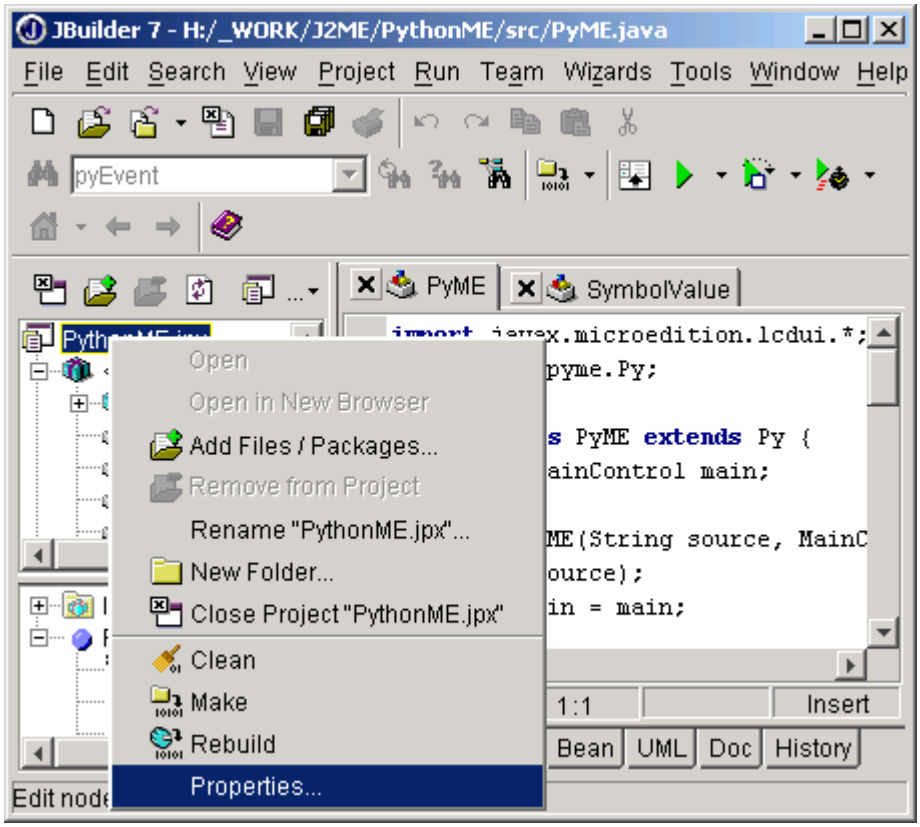
1. ความต้องการของระบบ (System requirement)

- ระบบปฏิบัติการ Windows 98/ME/2000/XP แต่การทำงานได้อย่างสมบูรณ์ที่สุดบน Windows 2000 Professional
- โปรแกรม JBuilder 7 SE หรือสูงกว่า
- โปรแกรม Mobile set 3.0 for JBuilder 7 SE
- โปรแกรม Parser Generator

2. การติดตั้งและการใช้โปรแกรม

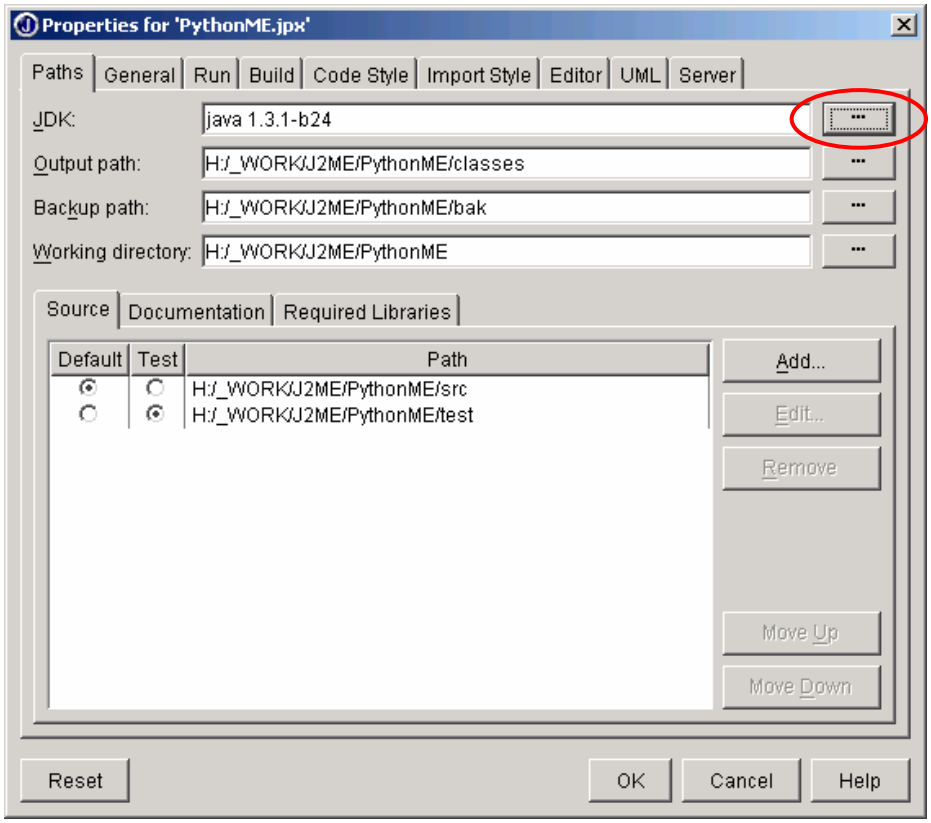
2.1 การทำให้โปรแกรม JBuilder สามารถเขียนโปรแกรมสำหรับมือถือ

1. ลงโปรแกรม JBuilder ตามปกติ และลงโปรแกรม Mobile set 3.0 for JBuilder 7 SE
2. เปิดโปรแกรม JBuiler ขึ้นมาแล้วทำการสร้าง Project ขึ้นมาตามปกติ
3. คลิกขวาที่ชื่อ Project แล้วเลือกที่ Properties…



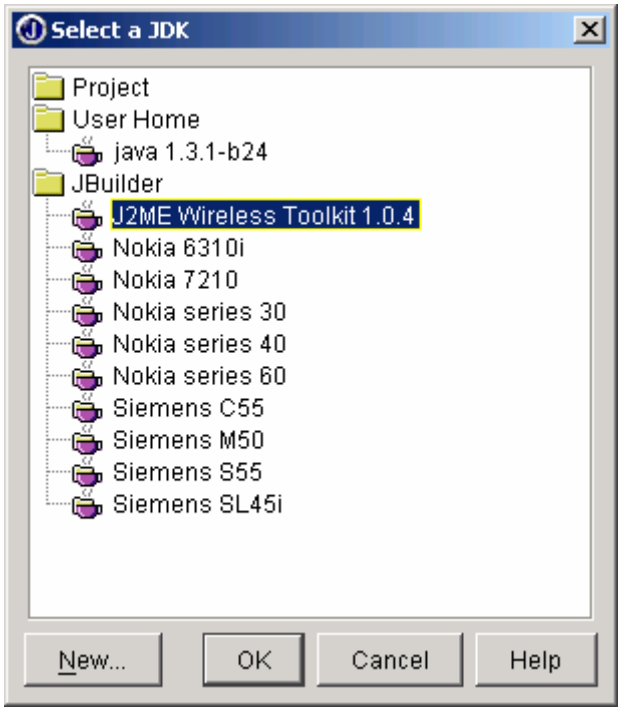
รูปที่ 2-1 แสดงตำแหน่ง Properties ของ Project ในโปรแกรม JBuilder

4. ที่ Tab Paths ทำการคลิกที่ ปุ่ม … ของช่อง JDK: ตามรูปที่ 2-2



รูปที่ 2-2 แสดงปุ่ม … บนหน้าต่าง Properties

5. ทำการเลือกที่ J2ME Wireless Toolkit 1.0.4 (ตรงนี้หากโปรแกรม JBuilder 7 ยังไม่ได้ทำการลง Mobile set จะไม่มี เท่านี้ก็จะเป็นอันว่าเสร็จการทำให้ JBuilder สามารถเขียนโปรแกรมบนมือถือได้

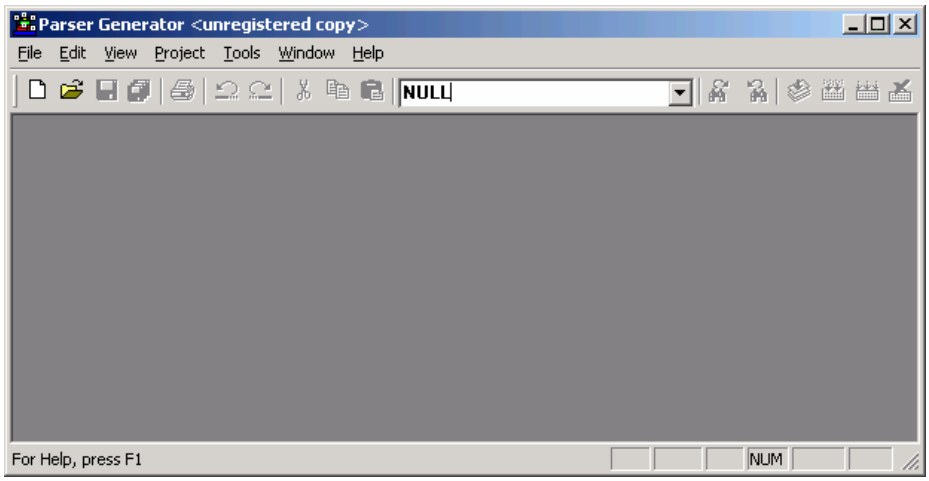


รูปที่ 2-3 แสดงรูปการเลือก JDK สำหรับใช้กับมือถือ

2.2 โปรแกรม Parser Generator

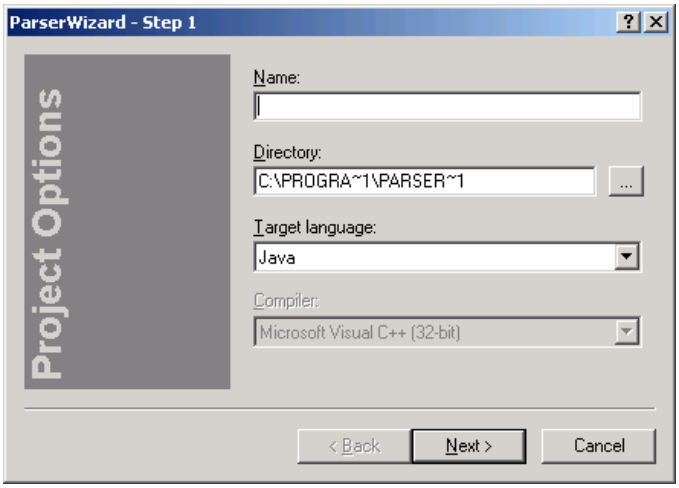
โปรแกรม Parser Generator เป็นโปรแกรมที่ใช้สำหรับสร้าง Regular Expression และ BNF แล้วโปรแกรมจะทำการสร้าง code ของภาษาต่างๆ เช่น c, c++ หรือ java ตามที่ต้องการจาก Regular Expression และ BNF ที่เราได้ทำการกำหนดไว้ ซึ่งวิธีการใช้งานเป็นดังต่อไปนี้

1. ติดตั้งโปรแกรม Parser Generator ตามขั้นตอนปกติ
2. เลือกที่ start → Programs → Parser Generator 2 → Parser Generator จะปรากฏหน้าต่างดังนี้



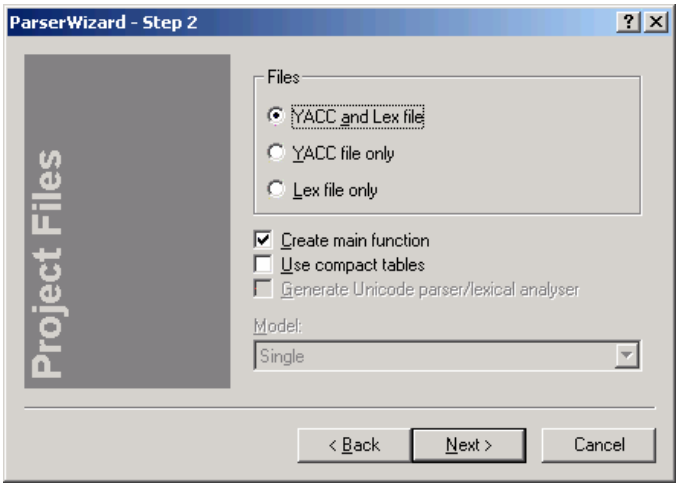
รูปที่ 2-4 แสดงหน้าต่างโปรแกรม Parser Generator

3. คลิกที่เมนู Parser → ParserWizard… เพื่อทำการสร้างโปรเจ็คขึ้นมาใหม่ ซึ่งจะได้หน้าต่างดังนี้



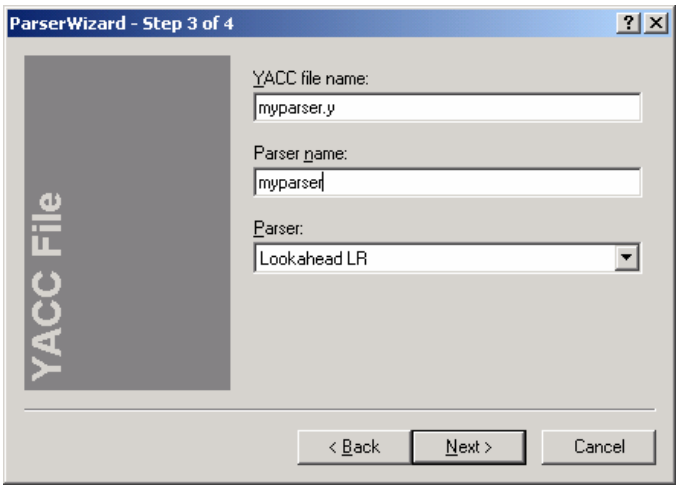
รูปที่ 2-5 แสดงหน้าต่างสำหรับการสร้างโปรเจ็คใหม่

4. ทำการตั้งชื่อโปรเจ็ค Directory สำหรับโปรเจ็ค และ ภาษาที่ต้องการจะให้โปรแกรมทำการสร้างให้ จากนั้นคลิกที่ปุ่ม Next > จะเข้าสู่ step 2 ดังรูป



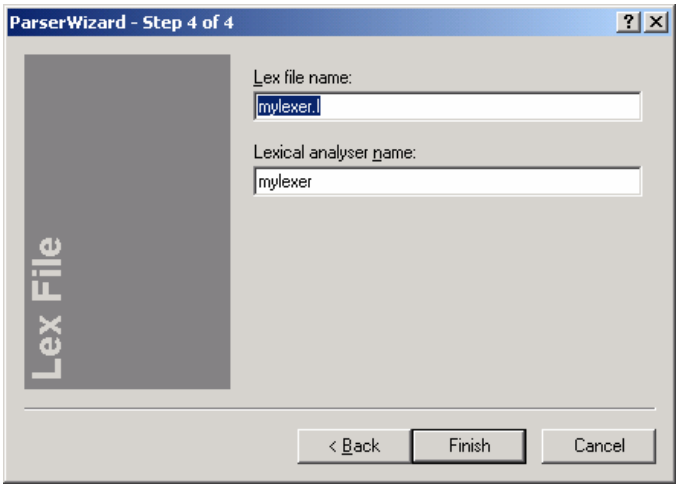
รูปที่ 2-6 แสดงหน้าต่าง step 2 ของการสร้างโปรเจ็คใหม่

5. ใน step 2 นี้จะไม่เลือกอะไรเนื่องจากได้ทำการเลือกไว้ที่ YACC and Lex file ซึ่งหมายถึงการสร้างทั้ง BNF และ Regular Expression อยู่แล้ว จากนั้นคลิกที่ปุ่ม Next> จะเข้าสู่ step 3 ดังรูป



รูปที่ 2-7 แสดงหน้าต่าง step 3 ของการสร้างโปรเจ็คใหม่

6. ใน step 3 นี้จะเป็นการตั้งชื่อไฟล์ และตั้งชื่อ class สำหรับ BNF ส่วนในช่อง Parser: นั้นเป็น algorithm ในการทำ parsing ซึ่งจะใช้แบบ Look ahead LR จากนั้นคลิกที่ปุ่ม Next > จะเข้าสู่ step 4 ดังรูป



รูปที่ 2-8 แสดงหน้าต่าง step 4 ของการสร้างโปรเจ็คใหม่

7. ใน step 4 นี้จะเป็นการตั้งชื่อไฟล์ และตั้งชื่อ class สำหรับ Regular Expression จากนั้นคลิกที่ปุ่ม Finish ก็จะเป็นการเสร็จสิ้นการสร้างโปรเจ็คใหม่

3. แนวทางการพัฒนา

การพัฒนาอินเตอร์พรีตเตอร์ไพธอนขนาดเล็กนี้จะมีส่วนหลักๆ อยู่ด้วยกัน 3 ส่วนซึ่งแต่ละส่วนก็จะมี class ต่างๆ ที่ทำหน้าที่ในแต่ละส่วนนั้นซึ่งจะอธิบายเฉพาะส่วนที่มีความสำคัญ สำหรับการนำไปพัฒนาต่อ ซึ่งจะได้แนะนำเป็นส่วนๆ ได้ดังนี้

3.1 ส่วนของการ Fetch คำสั่ง

- Class com.pyme.Py เป็น abstract class ซึ่งจะทำหน้าที่เป็นเหมือน class หลักสำหรับรับ source code ที่จะนำมาทำการอินเตอร์พรีต ซึ่งใน class นี้จะทำการ Fetch คำสั่งและเรียกส่วนของการ Analyze มาทำงานอีกทีหนึ่ง

  Method สำคัญ

  - **public void Interprete();** เป็น method สำหรับทำการอินเตอร์พรีต source code ที่ได้รับเข้ามาเมื่อทำการ new class นี้
  - **public abstract void pyEvent(int eventType, String eventMsg);** เป็น abstract method สำหรับเมื่อการอินเตอร์พรีตนั้นเกิดเหตุการณ์ต่างๆ ซึ่งจะต้องนำเหตุการณ์นั้นไปกระทำต่อซึ่งก็จะมีด้วยกัน 3 เหตุการณ์คือ
    1. STDOUT คือ เมื่อมีการแสดงผลออกทาง output
    2. OPENURL คือ เมื่อมีการทำคำสั่งเปิดเว็ปไซด์
    3. SENDMAIL คือ เมื่อมีการทำคำสั่งส่งเมลล์

3.2 ส่วนของการ Analyze

- Class com.pyme.lexyacc.* เป็น package ของ class ที่จะเกิดจากการสร้างขึ้นอัตโนมัติ ด้วยโปรแกรม Parser Generator ซึ่งตรงนี้เราจะต้องทำการสร้าง Regular Expression และ BNF ขึ้นมาก่อนจากนั้นก็ทำการ generate ก็จะได้มาเป็นไฟล์ภายใต้ package นี้นั่นเอง

3.3 ส่วนของการ Execute

- Class com.pyme.execute.ParseTree เป็น class ของโครงสร้างของคำสั่งแบบต่างๆ ซึ่งถ้าหากนำไปพัฒนาต่อ และมี type ของคำสั่งใหม่ๆ เพิ่มเข้ามาก็จะต้องเพิ่มชนิดของคำสั่งเข้าไปใน class นี้ด้วย
- Class com.pyme.execute.SimPyExec เป็น class สำหรับทำการ execute คำสั่งต่างๆ ที่ได้ทำการออกแบบไว้แล้ว ซึ่งจะทำการ execute ไปตามโครงสร้างของคำสั่ง

  Method สำคัญ

  - **public SymbolValue execution (SymbolValue symValue);** เป็น method ที่จะนำโครงสร้างของคำสั่งเข้ามา execute ฉะนั้นหากมีการพัฒนาต่อให้มีคำสั่งเพิ่มขึ้นก็จะต้องเพิ่มการ execute เข้าไปที่ method นี้
- Class com.pyme.execute.SymbolTable เป็น class ที่ใช้จัดการกับตัวแปรต่างๆ ที่ใช้ในแต่ละคำสั่ง ในส่วนนี้หากมีการจัดการเรื่องของ local และ global variable ก็จะต้องทำการจัดการใน class นี้
- Class com.pyme.execute.SymbolValue เป็น class ที่เป็นโครงรวมของชุดคำสั่งทั้งหมด ซึ่งบงบอกถึงชนิดของข้อมูลต่างๆ คำสั่งต่างๆ รวมถึงทรีของคำสั่งด้วย