



การเขียน

โปรแกรมเชิงวัตถุด้วยภาษา

จาวา

Basic Object-Oriented Programming with JAVA



JAVA

นรภัทร เพิ่มพูล

100.-



การเขียนโปรแกรมเชิงวัตถุ ด้วยภาษาจาวา (Object-Oriented Programming with JAVA)

ประเพณีวิชาอุตสาหกรรม สาขาวิชาเทคโนโลยีคอมพิวเตอร์
มีเนื้อหาตรงตามหลักสูตรประกาศนียบัตรวิชาชีพชั้นสูง พุทธศักราช 2563
สำนักงานคณะกรรมการการอาชีวศึกษา (สอศ.) กระทรวงศึกษาธิการ
และใช้ได้กับหลักสูตรปริญญาตรี หรือผู้ที่สนใจในงานคอมพิวเตอร์ทั่วไป



เรียบเรียงโดย
นรภัทร เพิ่มพูล

การเขียนโปรแกรมเชิงวัตถุด้วยภาษาจาวา (Object-Oriented Programming with JAVA)

คำอธิบายรายวิชา

ศึกษาและปฏิบัติเกี่ยวกับหลักการเบื้องต้นของการเขียนโปรแกรมเชิงวัตถุด้วยภาษาจาวา JAVA จาวาแพลตฟอร์ม Application Processing Interfacing (API) รูปแบบโปรแกรมภาษาจาวา JAVA โอเปอเรเตอร์ เอ็กเพรสชัน การควบคุมการทำงานของโปรแกรม คลาส ออบเจกต์ และเรฟเฟอเรนซ์ (References) อินเฮอริแตนซ์ (Inheritance) Abstract Class and Interfaces การแพ็ก (Packages) Exception Handling การรับส่งข้อมูลผ่านอินพุต เอาต์พุต Abstract Window Toolkit (AWT) แอปเพล็ต (Applets) การทำ Multi-threading และ Networking การเชื่อมต่อแบบฐานข้อมูล (JAVA Database Connectivity-JDBC) การจัดการ Foundation Classes และ Security

พิมพ์ครั้งที่ 1
พ.ศ. 2565

ข้อมูลทางบรรณานุกรมของสำนักหอสมุดแห่งชาติ

นรภัทร เพิ่มพูล.

การเขียนโปรแกรมเชิงวัตถุด้วยภาษาจาวา.--กรุงเทพฯ : วังอักษร, 2565.
196 หน้า.

1. การเขียนโปรแกรม (คอมพิวเตอร์).
2. จาวา (ภาษาคอมพิวเตอร์).
3. การโปรแกรมเชิงวัตถุ. I. ชื่อเรื่อง.

005.133

ISBN 978-616-495-227-0

จัดพิมพ์และจัดจำหน่ายโดย..



บริษัทวังอักษร จำกัด

69/3 ถนนอรุณอมรินทร์ แขวงวัดอรุณ เขตบางกอกใหญ่ กรุงเทพฯ 10600

โทร. 0-2472-3293-5 โทรสาร 0-2891-0742 Mobile 08-8585-1521

e-Mail : wangaksorn9@gmail.com

Facebook : สำนักพิมพ์ วังอักษร

ID Line : wangaksorn

www.wangaksorn.com



สงวนลิขสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2561

โดยบริษัทวังอักษร จำกัด ห้ามนำส่วนใดส่วนหนึ่งของหนังสือเล่มนี้ไปทำซ้ำ
ดัดแปลง หรือเผยแพร่ต่อสาธารณชน ไม่ว่ารูปแบบใด ๆ นอกจากได้รับอนุญาต
เป็นลายลักษณ์อักษรล่วงหน้าจากทางบริษัทฯ เท่านั้น ชื่อและเครื่องหมายการค้าอื่น ๆ
ที่อ้างอิงในหนังสือฉบับนี้เป็นสิทธิโดยชอบด้วยกฎหมายของเจ้าของแต่ละราย
โดยบริษัท วังอักษร จำกัด มิได้อ้างความเป็นเจ้าของแต่อย่างใด



คำนำ

วิชาการเขียนโปรแกรมเชิงวัตถุด้วยภาษาจาวา ประเภทวิชาอุตสาหกรรม สาขาวิชาเทคโนโลยีคอมพิวเตอร์ มีเนื้อหาตรงตามหลักสูตรประกาศนียบัตรวิชาชีพชั้นสูง พุทธศักราช 2563 สำนักงานคณะกรรมการการอาชีวศึกษา (สอศ.) กระทรวงศึกษาธิการ และใช้ได้กับหลักสูตรปริญญาตรี หรือผู้ที่สนใจในงานคอมพิวเตอร์ทั่วไป ผู้เขียนได้บริหารสาระการเรียนรู้แบ่งเป็น 12 บทเรียน ได้จัดแผนการจัดการเรียนรู้/แผนการสอนที่มุ่งเน้นฐานสมรรถนะ (Competency Based) และการบูรณาการ (Integrated) ตรงตามจุดประสงค์รายวิชา สมรรถนะรายวิชา และคำอธิบายรายวิชา ในแต่ละบทเรียน มุ่งให้ความสำคัญส่วนที่เป็นความรู้ ทฤษฎี หลักการ กระบวนการ ตัวอย่าง แบบฝึกปฏิบัติ และคำถาม เพื่อการทบทวนในการฝึกทักษะประสบการณ์ เร่งพัฒนาบทบาทของผู้เรียนให้เป็นผู้จัดการและแสวงหาความรู้ (Explorer) เพื่อสร้างองค์ความรู้ใหม่ และบทบาทของผู้สอนเปลี่ยนจากผู้ให้ความรู้มาเป็นผู้จัดการชี้แนะ (Teacher Roles) จัดสิ่งแวดล้อมที่เอื้ออำนวยต่อความสนใจเรียนรู้ และเป็นผู้ร่วมเรียนรู้ (Co-investigator) จัดห้องเรียนเป็นสถานที่ทำงานร่วมกัน (Learning Context) จัดกลุ่มเรียนรู้ให้รู้จักทำงานร่วมกัน (Grouping) มุ่งสร้างสรรค์คนรุ่นใหม่ ฝึกทักษะสอนความสามารถที่นำไปทำงานได้ (Competency) สอนความรัก ความเมตตา (Compassion) ความเชื่อมั่น ความซื่อสัตย์ (Trust) เป้าหมาย อาชีพอันยังประโยชน์ (Productive Career) และชีวิตที่มีศักดิ์ศรี (Noble Life) เหนือสิ่งอื่นใด เป็นคนดีทั้งกาย วาจา ใจ มีคุณธรรม มีจรรยาบรรณทางธุรกิจและวิชาชีพ

ส่งเสริมสนับสนุนยุทธศาสตร์การพัฒนาระบบคุณวุฒิวิชาชีพ (Vocational Qualification System) ให้สอดคล้องตามมาตรฐานอาชีพ (Occupational Standard) เพื่อสร้างภูมิคุ้มกัน เพิ่มขีดความสามารถในการแข่งขันของประเทศ เพิ่มศักยภาพ กำลังแรงงาน การพัฒนามาตรฐานการปฏิบัติงานระดับชาติ (National Benchmarking) และการวิเคราะห์หน้าที่การงาน (Functional Analysis) เพื่อให้เกิดผลสำเร็จในภาคธุรกิจ อุตสาหกรรม และทุกสาขาอาชีพ เป็นการเตรียมความพร้อมของผู้เรียนเข้าสู่สนามการแข่งขันในประชาคมอาเซียน

นรภัทร เพิ่มพูล



สารบัญ



บทที่ 1 การเขียนโปรแกรมบนระบบมาตรฐานเปิด

1

บทนำ

3

ความเป็นมาและนิยามของระบบมาตรฐานเปิด

4

ระบบมาตรฐานเปิด (Open Standard System)

4

การกำหนดสภาพแวดล้อมในการใช้งานภาษาจาวา

5

การดาวน์โหลดและติดตั้ง Java Development Kit (JDK)

5

การดาวน์โหลดและติดตั้งเอดิเตอร์ อีclipse (Eclipse)

8

หลักการทำงานพื้นฐานของภาษาจาวา

9

การติดตั้งเอดิเตอร์อีclipse

10

สร้างโปรเจกต์ (Project)

12

สร้างโครงสร้างแพ็คเกจ (Package)

12

สร้างคลาส (Class)

13

การเขียนและแก้ไขข้อผิดพลาดเบื้องต้น

14

ทำความเข้าใจเกี่ยวกับ System.out.println()

15

การใช้เอกสารเพื่อการบันทึกและช่วยเหลือ (Javadoc)

17

สรุป

18

แบบทดสอบและกิจกรรมการฝึกทักษะ

19



บทที่ 2 ภาษาจาวาพื้นฐาน

21

รูปแบบการเขียนโปรแกรมในภาษาจาวาพื้นฐานลักษณะโครงสร้างภาษาจาวาทั่วไป

23

การประกาศคลาส

25

กฎการระบุชื่อ (Identifier)

26

การคอมเมนต์ และการสร้างเอกสารในจาวา (javadoc)

28

การสร้างเอกสารในจาวา (javadoc)

29

การคอมไพล์ Javadoc

31



หลักสากลนิยมในการระบุชื่อ	32
ชนิดของข้อมูล ในภาษาจาวา	33
การใช้งานทางคณิตศาสตร์ในภาษาจาวา	39
การแปลงชนิดของข้อมูล	50
การสร้างและใช้งานเมธอดแบบสืบทอดคุณสมบัติ	52
สรุป	57
แบบทดสอบและกิจกรรมการฝึกทักษะ	58



บทที่ 3 อาร์เรย์

59

อาร์เรย์ 1 มิติ	62
อาร์เรย์ 2 มิติ	63
สรุป	67
แบบทดสอบและกิจกรรมการฝึกทักษะ	68



บทที่ 4 การเปรียบเทียบและคำสั่งควบคุมทิศทางของโปรแกรม

69

การควบคุมทิศทางการทำงาน (Control Flow)	71
คำสั่งวนลูป (Looping)	76
คำสั่งข้ามการทำงาน (Jump Statement)	80
สรุป	84
แบบทดสอบและกิจกรรมการฝึกทักษะ	85



บทที่ 5 โปรแกรมภาษาจาวาเชิงวัตถุ

87

ภาษาจาวากับการเขียนโปรแกรมแบบเชิงวัตถุ	89
คุณสมบัติการห่อหุ้ม (Encapsulation)	89
การสร้างคอนสตรัคเตอร์ (Constructor)	95
สรุป	99
แบบทดสอบและกิจกรรมการฝึกทักษะ	100



บทที่ 6 คุณสมบัติเชิงวัตถุ

101

แพ็คเกจ (Package)	103
ตัวระบุ (Modifier) ในภาษาจาวา	104
คีย์คำสั่งที่พบบ่อยในภาษาจาวา	105
คุณสมบัติการถ่ายทอด หรือสืบทอดคุณสมบัติ (Inheritance)	107
คุณสมบัติการเป็นได้หลายรูปแบบหรือการพ้องรูป (Polymorphism)	110
สรุป	113
แบบทดสอบและกิจกรรมการฝึกทักษะ	114



บทที่ 7 อินเตอร์เฟส

115

อินเตอร์เฟส (Interface)	117
การสร้างและกำหนดอินเตอร์เฟส	118
สรุป	123
แบบทดสอบและกิจกรรมการฝึกทักษะ	124



บทที่ 8 การจัดการเหตุการณ์ในภาษาจาวา

125

คอมโพเนนต์ที่สามารถกำหนดเหตุการณ์ได้ (Event Components)	127
คลาสที่เกี่ยวข้องกับเหตุการณ์ (Event Classes)	127
สรุป	135
แบบทดสอบและกิจกรรมการฝึกทักษะ	136



บทที่ 9 การจัดการข้อยกเว้นในภาษาจาวา

137

Exception handling ที่ระบบสร้างมาให้	140
คีย์คำสั่ง Throw ใน Exception	143
สรุป	147
แบบทดสอบและกิจกรรมการฝึกทักษะ	148



บทที่ 10 คลาสภายใน

149

1. สมาชิกคลาสแบบสถิต (Static Member Classes)	151
2. สมาชิกของคลาส (Member classes)	152
3. คลาสเฉพาะที่ (Local classes)	153
4. คลาสนิรนามภายใน (Anonymous Inner Classes)	154
สรุป	155
แบบทดสอบและกิจกรรมการฝึกทักษะ	156



บทที่ 11 เธรด

157

เธรดคืออะไร	159
ประเภทของเธรด	160
ข้อดีของเธรด (Thread)	161
วงจรชีวิตของเธรดในภาษาจาวา	161
การใช้ Exceptions ในเธรด	162
การสร้างเธรด (Thread Creation)	163
ตารางการทำงานของเธรดในภาษาจาวา (Java Thread Scheduling)	165
เธรดที่สำคัญในภาษาจาวา	166
วิธีการสร้างเธรด (Thread Spawning)	167
การทำงานให้สอดคล้องกันของเธรด (Thread Synchronization)	170
สรุป	171
แบบทดสอบและกิจกรรมการฝึกทักษะ	172



บทที่ 12 สตรีมไอโอ

173

สตรีมไอโอ (Stream I/O)	175
สรุป	184
แบบทดสอบและกิจกรรมการฝึกทักษะ	185



บรรณานุกรม

186



การเขียนโปรแกรมบนระบบ มาตรฐานเปิด

บทที่ 1

แนวคิด

ระบบมาตรฐานเปิด คือ กฎและข้อกำหนดซึ่งรวบรวมคำอธิบายคุณลักษณะของการออกแบบหรือการดำเนินงานของโปรแกรมหรืออุปกรณ์ที่นำไปแสดงผลหรือเพื่อการเผยแพร่ ทำให้ผู้นำไปใช้มีอิสระในการนำไปใช้ระบบนี้มีผู้เป็นเจ้าของลิขสิทธิ์ชัดเจนแต่ได้อุทิศให้ใช้งานโดยไม่มีค่าใช้จ่าย ซึ่งจะส่งผลให้มีกลุ่มที่ร่วมพัฒนาโปรแกรมหลากหลายและสามารถให้คำปรึกษาหรือแก้ปัญหาได้อย่างดี

สาระการเรียนรู้

1. ความเป็นมาและนิยามของระบบมาตรฐานเปิด
2. ระบบมาตรฐานเปิด (Open Standard System)
3. การกำหนดสภาพแวดล้อมในการใช้งานภาษาจาวา
4. การดาวน์โหลดและติดตั้ง Java Development Kit (JDK)
5. การดาวน์โหลดและติดตั้งเอดิเตอร์ อีคลิปส์ (Eclipse)
6. หลักการทำงานพื้นฐานของภาษาจาวา
7. การติดตั้งเอดิเตอร์อีคลิปส์
8. สร้างโปรเจกต์ (Project)
9. สร้างโครงสร้างแพ็คเกจ (Package)
10. สร้างคลาส (Class)
11. การเขียนและแก้ไขข้อผิดพลาดเบื้องต้น
12. ทำความเข้าใจเกี่ยวกับ `System.out.println()`
13. การใช้เอกสารเพื่อการบันทึกและช่วยเหลือ (API Doc)



สมรรถนะ

1. ปฏิบัติการติดตั้งเอดิเตอร์อ็คลิปส์ สำหรับการเขียนโปรแกรมได้สำเร็จ
2. สร้างโปรเจกต์และโครงสร้างแพ็คเกจได้ถูกต้อง
3. เขียนและแก้ไขข้อผิดพลาดเบื้องต้นได้ถูกต้อง

จุดประสงค์เชิงพฤติกรรม

1. อธิบายความเป็นมาของระบบมาตรฐานเปิด
2. อธิบายระบบมาตรฐานเปิด
3. อธิบายหลักการทำงานพื้นฐานของภาษาจาวา
4. ปฏิบัติการติดตั้งเอดิเตอร์อ็คลิปส์ สำหรับการเขียนโปรแกรมได้
5. สามารถสร้างโปรเจกต์และโครงสร้างแพ็คเกจได้
6. สามารถเขียนและแก้ไขข้อผิดพลาดเบื้องต้นได้



การเขียนโปรแกรมบนระบบมาตรฐานเปิด



บทนำ

การเขียนโปรแกรมบนระบบมาตรฐานเปิด (Open Standard System) มีเนื้อหาเกี่ยวกับความรู้พื้นฐานเกี่ยวกับการเขียนโปรแกรมจากระดับพื้นฐาน ซึ่งสามารถนำมาเขียนโปรแกรมบนระบบมาตรฐานเปิด เริ่มตั้งแต่ความหมายของระบบมาตรฐานเปิด การติดตั้งและเตรียมใช้งานภาษาจาวา การคอมไพล์และรันจาวาแอปพลิเคชัน ทำความเข้าใจเกี่ยวกับบทบาทของจาวาเวอร์ชวลแมชีน (Java Virtual Machine) หรือ JVM การใช้งานเอกสารเพื่อการบันทึกและช่วยเหลือ (API Docs) รูปแบบการเขียนภาษาจาวา (Syntax) การใช้คำสั่งควบคุมด้วยภาษาแบบต่าง ๆ การใช้ภาษาจาวาในกรอบของการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) โดยทำความเข้าใจเรื่องของการสร้าง แพ็กเกจ (Package) คลาส (Class) แบบต่าง ๆ เช่น Abstract Class และ Interface Class การประกาศคุณสมบัติ (Properties) และเมธอด (Method) ตลอดจนคุณสมบัติเด่นในการเขียนโปรแกรมเชิงวัตถุ การควบคุมการเกิดข้อผิดพลาด (Exception and handle run time error) การใช้งานอินพุต-เอาต์พุตคลาส (I/O Class) จากคลาสที่มีใน JDK ที่สร้างไว้ให้ใช้ จากทั้งหมดนี้จะนำเอดิเตอร์ที่ได้รับความนิยมมาใช้เพื่อการเขียนโปรแกรมคือ **อีคลิป์เอดิเตอร์ (Eclipse Editor)** ซึ่งสามารถนำมาใช้โดยไม่ต้องเสียค่าใช้จ่ายใด ๆ วิธีการและเครื่องมือนี้เป็นที่ได้รับความนิยมในการใช้งานจนกลายเป็นมาตรฐานที่มีผู้นำไปใช้งานอย่างกว้างขวาง เพื่อการเขียนโปรแกรมหรือพัฒนาโปรแกรมบนระบบมาตรฐานเปิดนั่นเอง



ความเป็นมาและนิยามของระบบมาตรฐานเปิด

อุตสาหกรรมซอฟต์แวร์นั้นมีการพัฒนาไปอย่างกว้างขวางและหลากหลายรูปแบบ ตั้งแต่อดีตจนถึงปัจจุบัน จึงเป็นเหตุให้มีความหลากหลายทั้งมาตรฐาน และความนิยมในการนำมาใช้งาน ทั้งหมดนี้มีการวิวัฒนาการไปตามแนวทางและมาตรฐานของตนเอง ทั้งนี้หากเงื่อนไขการนำไปใช้งานที่อยู่ภายใต้กฎหมายลิขสิทธิ์ เรียกว่า **“ซอฟต์แวร์ที่มีลิขสิทธิ์ (License Software)”** ส่วนอีกกลุ่มหนึ่ง คือ ผู้นิยมซอฟต์แวร์แบบเปิดเผยรหัส (Open Source Software) ซึ่งเป็นซอฟต์แวร์ที่มีเงื่อนไขในการนำไปใช้อีกแนวทางหนึ่ง ในส่วนของซอฟต์แวร์ที่มีลิขสิทธิ์นั้น ถ้าต้องการนำมาใช้จะต้องซื้อสิทธิ์ตามเงื่อนไขของผู้เป็นเจ้าของลิขสิทธิ์ ส่วนระบบซอฟต์แวร์แบบเปิดเผยรหัสนั้นจะมีเงื่อนไขในการนำมาใช้โดยไม่ต้องเสียค่าใช้จ่ายใด ๆ ภายใต้ข้อตกลงของบทบัญญัติของระบบซอฟต์แวร์แบบเปิดเผยรหัส (Directors, 2012) ภายหลังเกิดปัญหาว่าได้มีการนำระบบซอฟต์แวร์แบบเปิดเผยรหัสออกไปขายหรือทำผิดเงื่อนไขเสียเอง ทำให้ผู้พัฒนาหรือผู้ใช้ซอฟต์แวร์ประเภทดังกล่าวเกิดความกังวลถึงอนาคตของการใช้ซอฟต์แวร์ประเภทดังกล่าว จึงเกิดระบบมาตรฐานเปิด (Open Standard System) ขึ้นโดยซอฟต์แวร์นั้นมีเจ้าของลิขสิทธิ์ที่ชัดเจน แต่ได้อุทิศซอฟต์แวร์นั้นให้นำไปใช้อย่างไม่มีค่าใช้จ่ายใดๆ ทั้งนี้มีคณะกรรมการพิจารณามาตรฐานในสิ่งที่เกี่ยวข้อง รวมทั้งอุปกรณ์ที่เข้าร่วมกันกับซอฟต์แวร์ดังกล่าวด้วย จึงทำให้ได้รับความนิยมอย่างมากในปัจจุบัน ดังคำนิยามของระบบมาตรฐานเปิดดังต่อไปนี้



ระบบมาตรฐานเปิด (Open Standard System)

นิยามของคำว่า **“ระบบมาตรฐานเปิด (Open Standard System)”** คือ กฎและข้อกำหนดซึ่งรวบรวมคำอธิบายคุณลักษณะของการออกแบบหรือการดำเนินงานของโปรแกรมหรืออุปกรณ์ที่นำไปแสดงผล หรือเพื่อการเผยแพร่ ทำให้ผู้นำไปใช้หรือกลุ่มผู้นำไปใช้มีความอิสระในการนำไปใช้

มาตรฐานเปิดมีการเจริญเติบโตอย่างรวดเร็ว โดยมีความร่วมมือกันทำงาน และส่งเสริมให้อุปกรณ์ของผู้ผลิตค่ายหนึ่ง สามารถทำงานร่วมกับอุปกรณ์จากผู้ผลิตที่แตกต่างกันได้ ระบบมาตรฐานเปิดนั้นเป็นมาตรฐานที่เป็นกรรมสิทธิ์ของบริษัทที่นำเสนอออกสู่ตลาด และผลักดันให้มาตรฐานของตนเองได้รับความนิยมด้วยความเต็มใจของบริษัทที่ได้อุทิศให้บุคคลทั่วไปใช้ โดยไม่มีค่าใช้จ่ายใด ๆ (Haas, 2012) นอกจากนี้ เราสามารถศึกษารายละเอียดได้จากวิกิพีเดีย ว่าด้วยระบบมาตรฐานแบบเปิด (Wikipedia) คือ ภาษาจาวา (Java Programming Language) ซึ่งในที่นี้จะกล่าวถึงเพียงส่วนที่จะนำมาใช้เพื่อการพัฒนาโปรแกรมเพื่อการศึกษาในหนังสือเล่มนี้เท่านั้น



การกำหนดสภาพแวดล้อมในการใช้งานภาษาจาวา

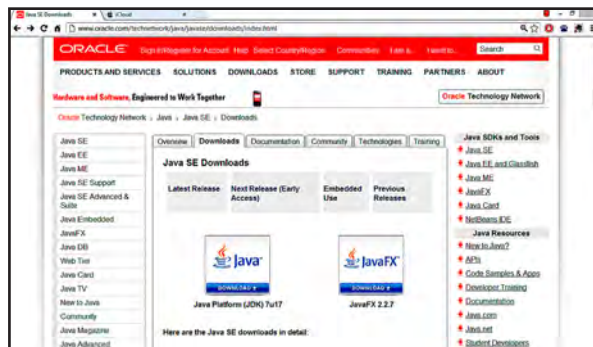
สิ่งที่จะต้องคำนึงถึงในการสร้างระบบงานใด ๆ ก็ตาม คือ สภาพแวดล้อมที่จำเป็นต้องใช้ในการพัฒนาระบบ ซึ่งในความเป็นจริงแล้วการกำหนดค่าต่าง ๆ ในปัจจุบันนี้สามารถทำได้ง่ายกว่ายุคก่อน ๆ มาก ในที่นี้จะกล่าวถึงการเลือกใช้โปรแกรมหรือเครื่องมือในการเขียนโปรแกรมด้วยภาษาจาวา ซึ่งเราสามารถใช้เครื่องมือตั้งแต่โปรแกรมโน้ตแพด (Notepad) ไปจนถึงเอดิเตอร์อื่น ๆ ที่สร้างขึ้นมารองรับอย่างหลากหลาย และการที่จะเลือกใช้นั้นจะต้องคำนึงถึงความง่ายในการกำหนดค่าการใช้งานตลอดจนรองรับอนาคตได้เป็นอย่างดี ในที่นี้จะขอเลือกเครื่องมือที่ได้รับความนิยมอย่างสูง สามารถนำมาใช้ได้โดยไม่ต้องเสียค่าใช้จ่ายใด ๆ และถือได้ว่าเป็นระบบมาตรฐานเปิดตัวหนึ่ง เอดิเตอร์ตัวนี้จะทำงานในลักษณะเฟรมเวิร์ค (Framework) มากกว่าที่จะเป็นเพียงเอดิเตอร์เพียงอย่างเดียว ทำให้สามารถติดตั้งส่วนเสริม (Add-on หรือ Plug-in) ได้หลากหลายและทำงานเบ็ดเสร็จในตัว เช่น เอดิเตอร์อีclipse เป็นเครื่องมือในการพัฒนาภาษาจาวาตามปกติ ต่อมาเราต้องการใช้ GWT (Google Web Tools Kit) หรือ Web service อื่น ๆ เราก็นำมากำหนดเป็นส่วนเสริมเพิ่มเติมเข้าไปในภายหลังได้อย่างสะดวกรวดเร็ว

ในการใช้งานจะแบ่งออกเป็นส่วนที่เกี่ยวข้องสองส่วนใหญ่ ๆ คือ การดาวน์โหลดและติดตั้ง JDK และการดาวน์โหลดและติดตั้งเอดิเตอร์อีclipse (Eclipse) ทั้งหมดนี้จะสามารถติดตั้งบนระบบปฏิบัติการที่เลือกใช้ ในที่นี้จะขอกล่าวถึงเฉพาะการติดตั้งบนระบบปฏิบัติการวินโดวส์เท่านั้น



การดาวน์โหลดและติดตั้ง Java Development Kit (JDK)

เราสามารถดาวน์โหลด JDK ได้จากเว็บไซต์ที่ปรากฏ โดยจะพบว่าลิงก์ที่กำหนดให้ดาวน์โหลด <http://www.oracle.com/tech-network/java/javase/downloads/index.html> รูปที่ 1.1



รูปที่ 1.1 แสดงหน้าเว็บไซต์

เพื่อดาวน์โหลด JDK เวอร์ชันล่าสุด

ในการดาวน์โหลดและติดตั้ง JAVA Development Kit (JDK) นั้น เมื่อเข้าไปในเว็บไซต์ผู้เรียนจะพบว่า มีไฟล์ให้ดาวน์โหลดหลากหลายไฟล์โดยแบ่งตามระบบปฏิบัติการของเครื่องคอมพิวเตอร์ที่ใช้

Java SE Development Kit 8u74		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux x86	154.74 MB	jdk-8u74-linux-i586.rpm
Linux x86	174.92 MB	jdk-8u74-linux-i586.tar.gz
Linux x64	152.74 MB	jdk-8u74-linux-x64.rpm
Linux x64	172.9 MB	jdk-8u74-linux-x64.tar.gz
Mac OS X x64	227.27 MB	jdk-8u74-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.72 MB	jdk-8u74-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.09 MB	jdk-8u74-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.02 MB	jdk-8u74-solaris-x64.tar.Z
Solaris x64	96.19 MB	jdk-8u74-solaris-x64.tar.gz
Windows x86	182.01 MB	jdk-8u74-windows-i586.exe
Windows x64	187.31 MB	jdk-8u74-windows-x64.exe

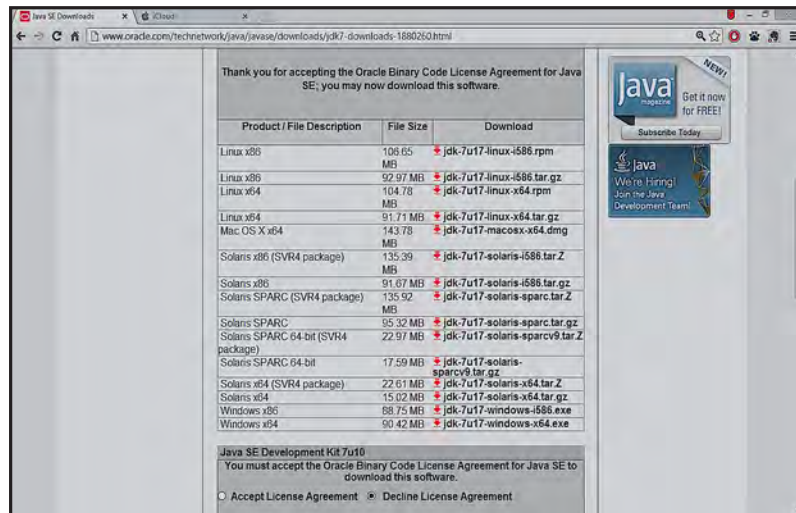
จากรูปจะเห็นว่า มีระบบปฏิบัติการให้เลือกตั้งแต่ Linux, Mac OS X, Solaris และ Windows ผู้เรียนจึงต้องเลือกดาวน์โหลด และติดตั้งตามระบบปฏิบัติการที่ตนใช้งาน ทั้งนี้ในแต่ละระบบปฏิบัติการจะมีการแบ่งเป็นระบบ 32 bit และ 64 bit ยกตัวอย่างเช่น Windows x86 เป็นระบบปฏิบัติการที่ทำงานแบบ 32 bit และ Windows x64 เป็นระบบปฏิบัติการที่ทำงานแบบ 64 bit ดังนั้น ก่อนที่จะติดตั้ง ผู้เรียนควรตรวจสอบปฏิบัติการที่ใช้ก่อนว่าเป็น “สภาพแวดล้อม” ไฉฉะจึงจะใช้งานได้อย่างเหมาะสม และถูกวิธี

Java Development Kit (JDK) คือ ชุดเครื่องมือที่ใช้ในการพัฒนาภาษาจาวา หากพบว่ายังไม่มีการติดตั้ง เราจะต้องทำการดาวน์โหลด JDK มาเสียก่อน ประกอบไปด้วย Java compiler, Java debugger, Java doc และ Java interpreter หรือ Java VM นอกจากนี้ JDK ประกอบด้วย 3 รุ่นย่อยที่จะนำมาพัฒนาด้วยภาษาจาวาดังนี้ (Wikipedia, 2555)

- » Java SE (Standard Edition) สำหรับพัฒนาโปรแกรมบนคอมพิวเตอร์เดสก์ทอปทั่วไป
- » Java ME (Micro Edition) สำหรับพัฒนาโปรแกรมบนอุปกรณ์พกพา เช่น โทรศัพท์มือถือ หรือพีดีเอ ส่วนมากใช้เขียนโปรแกรมเกม
- » Java EE (Enterprise Edition) สำหรับพัฒนาโปรแกรมในองค์กรใหญ่ ๆ หรือมีขอบเขตของโครงการขนาดใหญ่

การใช้งานจริงนั้นเรามักนิยมติดตั้งโดยการเลือก J2SE เพื่อสนองความต้องการในการพัฒนาขั้นพื้นฐาน แต่ถ้าจะเลือก J2EE นั้นจะเหมาะกับงานในระดับใหญ่หรือมีทีมงานมาก ส่วน J2ME นั้นเหมาะกับการพัฒนาบนมือถือ ซึ่งระยะต่อมาได้รับความนิยมน้อย เพราะมีรูปแบบการพัฒนาอื่น ๆ ที่ได้รับความนิยมมากกว่า เช่น การเขียนโปรแกรมบนมือถือสมาร์ทโฟน เช่น ไอโฟนที่ใช้การพัฒนาด้วยภาษา Objective-C หรือแม้กระทั่งมีการใช้รูปแบบการพัฒนาด้วยภาษาจาวา แต่ก็เลือกที่จะใช้ไลบรารีหรือคลาสอื่น ๆ ที่พัฒนาขึ้นมาเฉพาะเจาะจงกับสมาร์ทโฟนชนิดนั้น ๆ มากกว่าการใช้ J2ME เช่น การพัฒนาบนระบบปฏิบัติการแอนดรอยด์ (Android) เป็นต้น

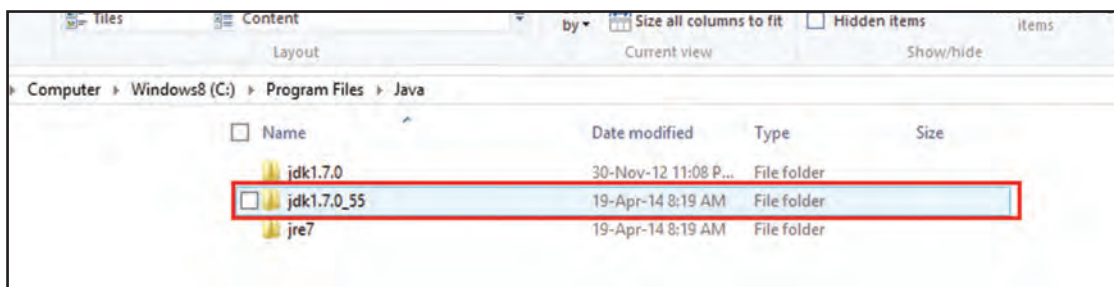
ส่วนของ JDK ณ เวลานั้นผู้เป็นเจ้าของ คือ บริษัทออราเคิล (Oracle) ซึ่งอดีตมีผู้ริเริ่มแนวคิดนี้มานานแล้วและเริ่มเป็นที่รู้จักมากขึ้นภายใต้บริษัทซันไมโครซิสเต็ม (Sun Microsystems) มีการพัฒนาหลายเวอร์ชัน และนี่ก็คือตัวอย่างหนึ่งของระบบเปิด ที่มีตัวตนของเจ้าของซอฟต์แวร์จริง แต่สามารถนำไปใช้ได้โดยไม่เสียค่าใช้จ่ายใด ๆ แม้ว่าต่อมาจะเปลี่ยนเจ้าของซอฟต์แวร์ในภายหลังดังกล่าว การดาวน์โหลดจะต้องเลือก Accept ตามข้อตกลงของ Java Platform, Enterprise Edition 6 SDK Update 4 (with JDK 7) License Agreement (อาจเปลี่ยนแปลง Edition ในอนาคตได้) ให้เลือกเวอร์ชันที่ตรงกับระบบปฏิบัติการที่ใช้แล้วติดตั้งโดยเพียงแต่คลิกเลือกลิงก์นั้นก็จะทำการดาวน์โหลดทันที



รูปที่ 1.2 ลิงก์ดาวน์โหลด Java SE

หลังจากดาวน์โหลดเสร็จ เราต้องติดตั้งลงบนเครื่องคอมพิวเตอร์และจะพบว่าหลังติดตั้งเสร็จจะพบโฟลเดอร์ดังรูปที่ 1.3

เราจะพบว่าหลังจากติดตั้งแล้วจะมีโฟลเดอร์ทั้ง jdk ในเวอร์ชันนั้น และโฟลเดอร์ jre เพื่อใช้ในการรันโปรแกรมต่าง ๆ ที่จำเป็นต้องใช้ JVM (Java Virtual Machine) ฉะนั้นเอดิเตอร์ Eclipse จำเป็นต้องใช้สภาพแวดล้อมนี้เพื่อใช้ในการรัน แต่ถ้าหากต้องการพัฒนา ก็ต้องใช้ JDK เพื่อนำมาใช้งานในการพัฒนาโปรแกรมต่อไป

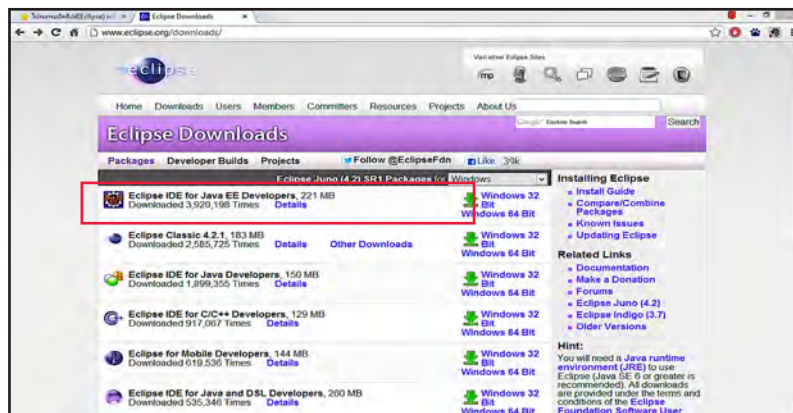


รูปที่ 1.3 แสดงโฟลเดอร์หลังจากที่มีการติดตั้ง JDK แล้ว



การดาวน์โหลดและติดตั้งเอดิเตอร์ อีคลิปส์ (Eclipse)

การใช้งานเอดิเตอร์ อีคลิปส์ (Eclipse) เป็นเครื่องมือเขียนโปรแกรม ของบริษัทไอบีเอ็ม (IBM) สามารถรองรับชุดพัฒนาโปรแกรมภาษาจาวา (Java Development Kit) หรือ นิยมเรียกว่า JDK โดยจะมีอยู่หลายเวอร์ชัน ซึ่งทุกเวอร์ชันสามารถทำงานร่วมกับเอดิเตอร์ตัวนี้ได้เป็นอย่างดี เราสามารถนำมาใช้โดยไม่ต้องเสียค่าใช้จ่ายใด ๆ หากเราต้องการใช้งานจะต้องทำการดาวน์โหลดอีคลิปส์ (Eclipse) มาจากเว็บไซต์ <http://www.eclipse.org/downloads/> โดยจะมีให้เลือกดาวน์โหลดอย่างมากมาย แต่เราจะต้องเลือกใช้ให้เหมาะสมกับการใช้งานดังรูปที่ 1.4 ด้านล่าง (ศึกษาความแตกต่างของ JDK แต่ละชนิดได้จากหน้าที่ 5)



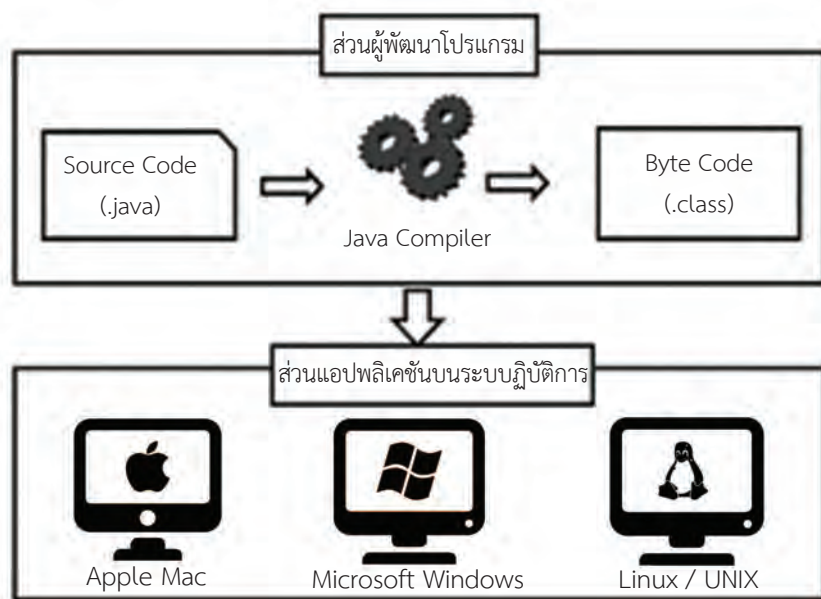
รูปที่ 1.4 หน้าเว็บไซต์สำหรับการดาวน์โหลด Eclipse

จากรูปที่ 1.4 เราควรเลือก Eclipse IDE for Java EE Developer เนื่องจากมีไลบรารีมาตรฐานและคลาสส่วนใหญ่จะมีอยู่อย่างครบถ้วน เหมาะกับการเขียนโปรแกรมด้วยภาษาจาวา แต่ก่อนที่จะทำการดาวน์โหลดควรที่จะต้องศึกษารายละเอียดก่อนเสมอ และจะต้องเลือกอีคลิปส์ที่เข้ากันได้กับระบบปฏิบัติการที่เราเลือกใช้ ซึ่งโดยปกติแล้วเว็บไซต์จะทำการตรวจสอบว่า ระบบปฏิบัติการของผู้ใช้นั้นเป็นชนิดใดก็จะแสดงรายการที่สามารถดาวน์โหลดได้ปรากฏขึ้นมา เว็บไซต์นี้เป็นเว็บไซต์อย่างเป็นทางการของ IBM ที่อาจมีการปรับปรุงให้เป็นเวอร์ชันที่ทันสมัยอยู่ตลอดเวลา นั้นหมายความว่า เราจะได้เอดิเตอร์ Eclipse ที่ทันสมัยที่สุด ณ เวลาที่เราเลือกดาวน์โหลด ณ เวลานั้น ๆ ก็ได้



หลักการงานพื้นฐานของภาษาจาวา

ภาษาจาวา เป็นภาษาที่มีหลักการงานที่สามารถนำไปใช้ได้กับทุกระบบปฏิบัติการ หรือ Write-one Run-Anywhere หากระบบปฏิบัติการนั้น ๆ มีส่วนของ JVM (Java Virtual Machine) ก็จะสามารถทำการรันโปรแกรมได้โดยไม่ต้องนำไปดัดแปลงโปรแกรมหรือทำการคอมไพล์ใหม่ ดังรูปที่ 1.5 ส่วนของผู้พัฒนาโปรแกรมนั้นจะเกี่ยวข้องกับผู้ที่เขียนโปรแกรมมากที่สุดที่จะพัฒนาโปรแกรมขึ้นมาได้ในส่วนของแอปพลิเคชันบนระบบปฏิบัติการต่าง ๆ จะมี JVM ทำหน้าที่ตีความหมายเป็น Native-Code ที่ทำให้ CPU บนระบบปฏิบัติการต่าง ๆ สามารถทำการประมวลผลได้



รูปที่ 1.5 แสดงรูปแบบการทำงานพื้นฐานของภาษาจาวาบนระบบปฏิบัติการต่าง ๆ

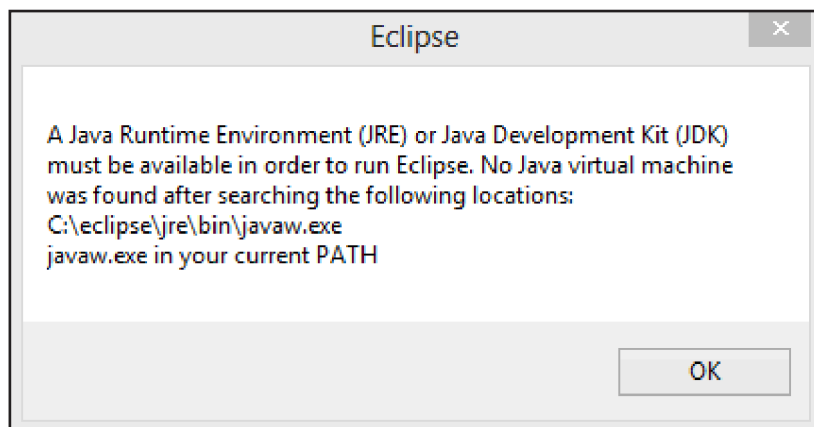
จากรูปที่ 1.5 คือ รูปแบบการทำงานพื้นฐานของภาษาจาวาบนระบบปฏิบัติการต่าง ๆ ซึ่งเราจะต้องทำการกำหนดค่าเริ่มต้น แต่ในปัจจุบันนี้มีเอดิเตอร์ที่ช่วยลดภาระการกำหนดค่าต่าง ๆ ให้สามารถทำงานได้สะดวกขึ้น ดังเอดิเตอร์อิดีลส์ที่เมื่อลงโปรแกรมเสร็จแล้วก็สามารถเขียนโปรแกรมบนเอดิเตอร์ได้เลยโดยไม่ต้องกำหนดเส้นทาง (Path) เพื่อให้ทำการรันโปรแกรมได้ แต่เอดิเตอร์ชนิดนี้จะทำการเชื่อมโยงไปยัง JVM เมื่อมีการสร้างโปรเจกต์ทุกครั้ง



การติดตั้งเอดิเตอร์อีคลิป์ส

อีคลิป์ส (Eclipse) นั้น เป็นตัวเอดิเตอร์แบบเฟรมเวิร์ค เป็นเครื่องมือสำหรับการเขียนโปรแกรม การคอมไพล์โปรแกรมและสั่งให้โปรแกรมทำงาน (run) ได้ และยังสามารถติดตั้งเครื่องมือเสริมได้หลายชนิด ตามจุดประสงค์ที่แตกต่างกันของผู้ใช้งาน

หลังจากการดาวน์โหลดเสร็จเราจะทำการแตกไฟล์ (.zip) เพื่อนำไปใช้งาน โดยไม่ต้องติดตั้งโปรแกรมใด ๆ เพียงแต่นำไฟล์เดอร์ Eclipse ไปวางไว้ที่ไดรฟ์ที่ต้องการเก็บแล้วก็สามารถเรียกใช้ eclipse.exe ภายใต้ไฟล์เดอร์ Eclipse ได้ทันที หากไม่มีการติดตั้ง JDK ก็จะขึ้นข้อความเตือนและไม่สามารถใช้งานได้ จะสังเกตเห็นว่าเราจะต้องมี Java Runtime Environment (JRE) ซึ่งเป็นสภาพแวดล้อมที่ต้องมีในการรันเอดิเตอร์นี้ ซึ่งมีส่วนของ Java Virtual Machine (JVM) เพื่อให้สามารถเอกซิคิวส์โปรแกรมที่พัฒนาจากภาษาจาวาได้ แต่ถ้าเป็น JDK จะประกอบไปด้วยชุดของซอฟต์แวร์ซึ่งเราสามารถพัฒนาด้วยภาษาจาวาได้ โดยจะมีส่วนประกอบอื่น ๆ ที่จำเป็นต้องใช้เพื่อการพัฒนาโปรแกรมด้วยภาษานี้ ดังนั้น เราจึงควรที่จะเลือกติดตั้ง JDK ก่อนเสมอ



รูปที่ 1.6 เมื่อเอดิเตอร์ Eclipse ไม่สามารถรันได้เนื่องจากไม่มี JDK

การทำงานของเอดิเตอร์ Eclipse หลังจากติดตั้งจนเสร็จสมบูรณ์แล้ว ในการเปิดใช้งานครั้งแรก เราจะต้องกำหนดค่าสภาพแวดล้อมต่าง ๆ ของอีคลิป์สก่อน เช่น การกำหนดเส้นทาง (Path) ที่จะเรียกใช้งาน JDK การกำหนดพื้นที่ทำงาน (Workspace) ซึ่งเป็นส่วนที่เก็บรวบรวมโปรเจกต์ แพ็กเกจ ฯลฯ และส่วนอื่น ๆ ที่ควรกำหนดไว้ตั้งแต่ต้น ดังจะได้กล่าวต่อไป