ความซับซ้อน ของ โปรแกรมทางพานิช /\*\* สไล12 \*\*/

1.ความซับซ้อนของตัวโปรแกรมเอง

2.non func ทำให้ซับซ้อน เช่น ต้องการความเร็ว หรือ เพอฟอเม้น

3. ต้องรองรับโปรแกรมสมัยก่อน

4. โปรแกรมเมอ เขียนโค๊ตโดยไม่อิงหลักใดหลักหนึ่ง เช่นไลเบอรี่เปลี่ยนเรื่อยๆ หรือ เฟรมเวิคเปลี่ยนเรื่อยๆ

จะรู้ได้อย่างไรว่าโปรแกรมนี้ซับซ้อน

* 1 Hierarchic Structure แบ่งออกเป็นลำดับชั้นๆ เช่น ระบบหายใจ มี ระบบปอด ระบบหมุนเวียนอากาศ บลาๆ
* 2Relative Primitives เชื่อมความสัมพันของโครงสร้าง มองมุมมองต่างๆกัน
* 3Separation of Concerns แบ่งการทำงานออกเป็นส่วนๆ
* 4Common Patterns วางระบบจากรูปแบบ แบบง่ายๆ
* 5Stable Intermediate Form ค่อยเพิ่มจากระบบรากฐาน

//\* สไล 16 \*//

* “Is a” hierarchy – Class structure
* “Part of” hierarchy – Object structure

//\* สไล 18 \*//

Algorithmic Decomposition = top down design ค่อยๆแยกจากปัญหาใหญ่ๆ แล้วย่อยไปเรื่อยๆ

ปัญหาคือถ้าหาไม่เจอว่าอยู่ตรงไหนต้องทำใหม่

Object Oriented Decomposition = แบ่งเป็นส่วนๆโดยไม่ต้องคำถึงรายระเอียดทั้งหมด แค่รู้ก็พอว่ามันควรจะทำไรได้บ้างและมีอะไรบ้าง

ส่วนประกอบของ การ Design

* Notation: Language for expressing each model ภาษาที่อทิบายโมเดล
* Process: Activities leading to the construction of models
* Tools: Artifacts that eliminate tedium and enforce the rules on the model to prevent inconsistency

Slide 2 page 4

* Major elements (a model is not an OO model without these elements) ต้องมี
  + Abstraction คิดว่าสิ่งๆนั้นทำอะไรได้บ้างหรือประกอบด้วยอะไร (มองจากภายนอก) แมวมีขน ตา หมวก
  + Encapsulation การปกปิดข้อมูลการอิมพริเม้น ออกจากผู้ใช้งาน
  + Modularity การแบ่งออกเป็นส่วนๆและมีการเชื่อต่อกันอย่างหลวมๆ
  + Hierarchy
    - Is a คือ?
    - Part of เป็นส่วนประกอบของ
* Minor elements (useful, but not essential) ควรมี
  + Typing
    - Strong & Weak typing
      * Strong ไม่ยอมรับไท ที่ไม่ถูกต้อง
      * Weak ยอมรับแล้วปรับเปลี่ยนให้
    - Dynamic & Static typing
      * Dynamic
      * Static
  + Concurrency ความสามารถในการทำพร้อมกัน
  + Persistence ความสามารถในการเก็บข้อมูลในการคงอยู่ ถึงแม้คราสที่สร้างมันจะหายไปแต่ถ้าสร้างไปแล้วก็ยังคงอยู่ วิทีคือเก็บลงในเดต้าเบส

Polymorphism

การทำให้ type 1 type ถูกแทนได้มากกว่า 1 รูปแบบ โดยให้มันอยู่ในรูปแบบของ subclass

เช่น 1 ชื่อ ทำได้ 2 แบบ