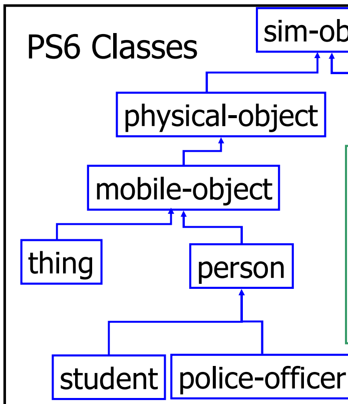
Lecture 24: Gödel's Proof

CS150: Computer Science
University of Virginia
Computer Science

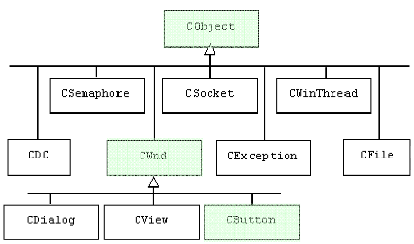
David Evans
http://www.cs.virginia.edu/evans

PS6 Classes

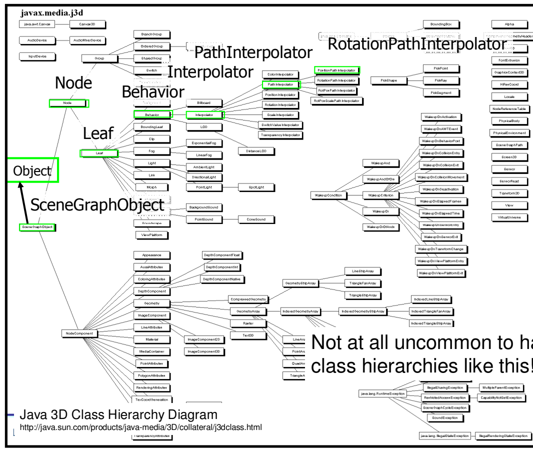


Are there class hierarchies like this in the "real world" or just in fictional worlds like Charlottansville?

Microsoft Foundation Classes



CButton inherits from CWnd inherits from CObject
"A button is a kind of window is a kind of object"



Java 3D Class Hierarchy Diagram
http://java.sun.com/products/java-media/3D/collateral/j3dclass.html

Quiz?

http://www.cs.virginia.edu/forums/viewtopic.php?t=1651

Story So Far

- Much of the course so far:
  - Getting comfortable with recursive definitions
  - Learning to write a program to do (almost) anything (PS1-4)
  - Learning more elegant ways of programming (PS5-6)
- This Week:
  - Getting ***un***-comfortable with recursive definitions
  - Understanding why there are some things no program can do!

Computer Science/Mathematics

Wednesday

- Computer Science (Imperative Knowledge)
  - Are there (well-defined) problems that cannot be solved by *any* procedure?

Today

- Mathematics (Declarative Knowledge)
  - Are there true conjectures that cannot be the shown using *any* proof?

Mechanical Reasoning

Aristotle (~350BC): *Organon*

Codify logical deduction with rules of inference (syllogisms)

Every A is a P
X is an A — Premises
X is a P — Conclusion

Every *human* is *mortal.*
Gödel is *human.*
Gödel is *mortal.*

More Mechanical Reasoning

- Euclid (~300BC): *Elements*
  - We can reduce geometry to a few axioms and derive the rest by following rules
- Newton (1687): *Philosophiæ Naturalis Principia Mathematica*
  - We can reduce the motion of objects (including planets) to following axioms (laws) mechanically

Mechanical Reasoning

- Late 1800s – many mathematicians working on codifying "laws of reasoning"
  - George Boole, *Laws of Thought*
  - Augustus De Morgan
- Whitehead and Russell, 1911-1913
  - *Principia Mathematica*
  - Attempted to formalize all mathematical knowledge about numbers and sets

All **true** statements about numbers

Perfect Axiomatic System

Derives **all** true statements, and **no** false statements starting from a finite number of axioms and following mechanical inference rules.

Incomplete Axiomatic System

Derives **some, but not all** true statements, and **no** false statements starting from a finite number of axioms and following mechanical inference rules.

incomplete

Inconsistent Axiomatic System

Derives **all** true statements, and **some** false statements starting from a finite number of axioms and following mechanical inference rules.

some false statements

Principia Mathematica

- Whitehead and Russell (1910– 1913)
  - Three Volumes, 2000 pages
- Attempted to axiomatize mathematical reasoning
  - Define mathematical entities (like numbers) using logic
  - Derive mathematical "truths" by following mechanical rules of inference
  - Claimed to be *complete* and *consistent*
    - All true theorems could be derived
    - No falsehoods could be derived

Russell's Paradox

- Some sets are not members of themselves
  - set of all Jeffersonians
- Some sets are members of themselves
  - set of all things that are non-Jeffersonian
- *S* = the set of all sets that are not members of themselves
- Is *S* a member of itself?

Russell's Paradox

- *S* = set of all sets that are not members of themselves
- Is *S* a member of itself?
  - If *S* **is** an element of *S*, then *S* **is** a member of itself and should **not** be in *S*.
  - If *S* **is not** an element of *S*, then *S* **is not** a member of itself, and **should** be in *S*.

Ban Self-Reference?

- *Principia Mathematica* attempted to resolve this paragraph by banning self-reference
- Every set has a type
  - The lowest type of set can contain only "objects", not "sets"
  - The next type of set can contain objects and sets of objects, but not sets of sets

Russell's Resolution?

Set ::= Set_n

Set_0 ::= { x | x is an *Object* }

Set_n ::= { x | x is an *Object* or a Set_{n-1} }

S: Set_n

Is *S* a member of itself?

No, it is a Set_n so, it can't be a member of a Set_n

Epimenides Paradox

Epidenides (a Cretan):

"All Cretans are liars."

Equivalently:

"This statement is false."

Russell's types can help with the set paradox, but not with these.

Gödel's Solution

All consistent axiomatic formulations of number theory include *undecidable* propositions.

(GEB, p. 17)

undecidable – cannot be proven either true or false inside the system.

Kurt Gödel

- Born 1906 in Brno (now Czech Republic, then Austria-Hungary)
- 1931: publishes *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme* (*On Formally Undecidable Propositions of Principia Mathematica and Related Systems*)

- 1939: flees Vienna
- Institute for Advanced Study, Princeton
- Died in 1978 – convinced everything was poisoned and refused to eat

Gödel's Theorem

In the Principia Mathematica system, there are statements that cannot be proven either true or false.

Gödel's Theorem

In any interesting rigid system, there are statements that cannot be proven either true or false.

Gödel's Theorem

All logical systems of any complexity are incomplete: there are statements that are *true* that cannot be proven within the system.

Proof – General Idea

- Theorem: In the Principia Mathematica system, there are statements that cannot be proven either true or false.
- Proof: Find such a statement

Gödel's Statement

G: This statement does not have any proof in the system of *Principia Mathematica*.

G is unprovable, but true!

Gödel's Proof Idea

G: This statement does not have any proof in the system of *PM*.

If *G* is provable, PM would be inconsistent.

If *G* is unprovable, PM would be incomplete.

Thus, **PM cannot be complete and consistent!**

Charge

- Wednesday:
  - Finish the proof: show we can express *G*
  - What is the equivalent to the Gödel sentence for computation?
- Friday:
  - How to prove a problem has no solving procedure
- Next Monday:
  - History of Object-Oriented Programming