Lecture 19 - Nov. 14

Inheritance

Polymorphism vs. Dynamic Binding Type Casts: Named vs. Anonymous Casts: Compilable vs. ClassCastException

Announcements/Reminders

- WrittenTest2 results to be released by Monday
- Lab4 due tomorrow at noon
- Lab5 to be released tomorrow
- ProgTest3 next Wednesday, November 20
 - + Lab4 grading tests
 - + Lab4 solution video
- Bonus Opportunity coming: Formal Course Evaluation



Rules of Substitutions (1)



Rules of Substitutions (2)



Rules of Substitutions (3)



Visualization: Static Type vs. Dynamic Type

<u>Static Type</u>: Expectation Dynamic Type: Accumulation of Code

s **new ResidentStudent**("Rachael");

S

Declaration:

Student s;

Substitution:



Change of Dynamic Type (1.1)



Change of Dynamic Type (1.2)



Change of Dynamic Type: Exercise (1)



Exercise 1: Android myPhone = new HuaweiP50Pro(...); myPhone = new GalaxyS21(...);

Change of **Dynamic** Type: Exercise (2)



Exercise 2: IOS myPhone = new HuaweiP50Pro(...); myPhone = new GalaxyS21(...);

Change of Dynamic Type (2.1)



Given:

Student jim = new Student(...);
ResidentStudent rs = new ResidentStudent(...);
NonResidentStudent nrs = new NonResidentStudent(...);

Example 1: jim = rs; , what's p(?) println(jim.getTuition()); jim = nrs; , wat's p(?); println(jim.getTuition());

Change of Dynamic Type (2.2)



Given:

Student jim = new Student(...);
ResidentStudent rs = new ResidentStudent(...);
NonResidentStudent nrs = new NonResidentStudent(...);

Example 2: rs ▲ jim; println(rs.getTuition()); nrs ▲ jim; println(nrs.getTuition());

Polymorphism and Dynamic Binding

Polymorphism:

An object's static type may allow multiple possible dynamic types.

⇒ Each dynamic type has its version of method.

Dynamic Binding:

An object's dynamic type determines the version of method being invoked.

Student jim = new ResidentStudent(...); jim.getTuition(); jim.getTuition(); jim.getTuition();



SmartPhone sp1 = new IPhone13Pro(...);
SmartPhone sp2 = new GalaxyS21(...);
sp1.surfWeb(); D : IP3 Po (falari)
sp1 = sp2;
sp1.surfWeb();



Type Cast: Motivation





Anatomy of a Type Cast

Student jim = new ResidentStudent("Jim");



Type Cast: Named vs. Anonymous





Compilable Type Cast May Fail at Runtime (1)



Compilable Type Cast May Fail at Runtime (2)



Exercise: Compilable Type Cast? Fail at Runtime? (1)



Compilable? ClassCastException at runtime?

Exercise: Compilable Type Cast? Fail at Runtime? (2)



Compilable? ClassCastException at runtime?

Compilable Cast vs. Exception-Free Cast

