Lecture 6: Subtyping

- Def of subtype, is-a
- Primitive subtypes
- Casts & instances
- Object, Comparable, Comparator

Announcements

- P1 due today, see submission instructions, P2 out soon
- No lecture/discussion 7/4 or 7/5
- Guest lectures next week (Prof. G. traveling)
- Feedback form
- Anyone w/o partner?
class A { f(); }
class C extends A { g(); }
class B extends A { h(); }
every C is an A.
every B is an A.
anything I can do with an A object, I can do with a B.

C and B are subtypes of A
cwe write C< A or C< A or
B b = new B();
A a;
a = b; // ok. (to change)
a.f();
A a = new C();
b = a; // not ok.
b.h();
a.h(); // not ok.

// I know (I'm clever)
// that a actually
// points to a B.

b = (B) a; // (run-time)
check if object is really of type B
type B.

(B) a; h();
((C) a). g(); // crash
C c = (C) a; // crash

Never use!

o instanceof B

boolean expression (true or false)

if (o instanceof B) {
  B b = (B) o;
  
  ...

Don't cast!

Cast: converts
a reference
from a supertype
to a subtype.
Can fail!
"Subtyping" for primitive types

\[
\text{char} \subset \text{int} \subset \text{float} \subset \text{double}
\]

\[
\text{float} \quad f = 1.0j \quad // \text{ok.}
\]

\[
\text{float}
\]

\[
\text{int} \quad i = 1.0j \quad // \text{not ok} \quad \text{int} \not\in \text{float}
\]

\[
\text{int} \quad \text{double} \quad \text{(Java's convention)}
\]

\[
\text{int} \quad i = 3.2j \quad // \text{doesn't compile}
\]

\[
\text{int} \quad i = (\text{int}) 3.2j \quad // \text{ok, bad idea.}
\]

\[
// \text{doesn't give error, approximates}
\]

characters are integers (internally)