Announcements

- HW9 due Thu at 11:55pm
- Grade cutoffs for HW1-6/PROJ1/MIDTERM1-2 posted
- FCQs on Thu
  - Volunteer? Pick up forms from Lesley McDowell (ECOT 721)
  - Turn in forms to the Dean’s office (ECAD 100)

Review

- Inclusion polymorphism
  - 

\[
\text{Object id(Object o) \{ return o; \} \quad 
\text{VI. TX: } \text{Object} \\
\text{class Rectangle \{ int \text{ width, \text{ height} \}; \}} \\
\text{Object o; \quad \text{id (o). \text{width} \}; \}} \\
\text{Rectangle r; \quad \text{id (r). \text{width} \}; \}} \\
\text{r.\text{m}() \quad \text{.m() \times type error}} \\
id (r). \text{m}() \quad \text{x type error by Java} \\
\]

- Parametric polymorphism

\[
\text{fun} \quad \text{(a x \rightarrow b \rightarrow \text{a})} \\
\text{because code} \\
\text{doesn’t look at the parameter} \\
\text{f++ } \text{(1, 2)} \\
\text{fun } \text{x = x} \\
\]

Issues with inclusion polymorphism

\[
\text{Object id(Object o) \{ return o; \} \quad \text{VI. TX: } \text{Object} \\
\text{class Rectangle \{ int \text{ width, \text{ height} \}; \}} \\
\text{Object o; \quad \text{id (o). \text{width} \}; \}} \\
\text{Rectangle r; \quad \text{id (r). \text{width} \}; \}} \\
\text{r.\text{m}() \quad \text{.m() \times type error}} \\
id (r). \text{m}() \quad \text{x type error by Java} \\
\]

Review Generics

Announcements

- HW9 due Thu at 11:55pm
- Grade cutoffs for HW1-6/PROJ1/MIDTERM1-2 posted
- FCQs on Thu
  - Volunteer? Pick up forms from Lesley McDowell (ECOT 721)
  - Turn in forms to the Dean’s office (ECAD 100)
Example

- Type of a search method over an Object array

```java
Object search(Object a[], Object k)

Rectangle r = ... architect a[].search(a[r].m(i));

c<T> T search(T a[], T k)
```

Example

- Sort method: type safe in Java

```java
<T> void sort(T a[], boolean g(T i, T j))
... g(a[i], a[j])...

Rectangle r = ...

c<T> Tengers(T a[], T m(i))

- Where T must T extends Comparable
- Combinaion of parametric and including polymorphism
```

Exercise

- Write a method that takes an array of objects and a value and return number of elements greater than that value.

```java
numgt: T extends Comparable,
r[0] x 0 => int

<T extends Comparable>
int numgt(T a[], T value) <
int num = 0
for i = 0 to a.length
if a[i].gt(value) & num++
3 3 int numgt(T extends Comparable a[], Comparable value) <
```
On to Data Abstraction

One-Slide Summary

- An abstract data type (ADT) is a type definition with a set of operations manipulating values of that type.
- The representation or implementation type is hidden from clients. This is known as information hiding.
- An ADT allows developers to impose a representation invariant—a property of the data structure preserved across operations.
- Objects give rise to a related but different form of data abstraction.

Simple Example

- Built-in types, like float, are abstract data types. Why?
  
  \[ 3.14 \quad \text{up to compiler, how to represent this} \]
  
  \[ + : \text{float} \times \text{float} \rightarrow \text{float} \]
  
  \[ \times : \text{float} \times \text{float} \rightarrow \text{float} \]
  
  \[ / : \text{float} \times \text{float} \rightarrow \text{float} \]

User-Defined ADTs

- Example: A stack of values
  
  an empty value
  
  push a new value
  
  pop a value

ADTs in SML

- Signatures and Structures

Data Abstraction in OO Languages

- Java-like languages permit information hiding through access control
For Next Time

- Reading
- Forum comment
- HW9