Subtyping vs inheritance

Madhavan Mukund

https://www.cmi.ac.in/~madhavan

Programming Concepts using Java

Week 3

Subclasses, subtyping and inheritance

- Class hierarchy provides both subtyping and inheritance

Subclasses, subtyping and inheritance

- Class hierarchy provides both subtyping and inheritance
- Subtyping
  - Capabilities of the subtype are a superset of the main type
  - If `B` is a subtype of `A`, wherever we require an object of type `A`, we can use an object of type `B`
  - `Employee e = new Manager(...);` is legal

Subclasses, subtyping and inheritance

- Class hierarchy provides both subtyping and inheritance
- Subtyping
  - Capabilities of the subtype are a superset of the main type
  - If `B` is a subtype of `A`, wherever we require an object of type `A`, we can use an object of type `B`
  - `Employee e = new Manager(...);` is legal
- Inheritance
  - Subtype can reuse code of the main type
  - `B` inherits from `A` if some functions for `B` are written in terms of functions of `A`
  - `Manager.bonus()` uses `Employee.bonus()`

Subtyping vs inheritance

- Recall the following example
  - `queue`, with methods `insert-rear`, `delete-front`
  - `stack`, with methods `insert-front`, `delete-front`
  - `deque`, with methods `insert-front`, `delete-front`, `insert-rear`, `delete-rear`

Subtyping vs inheritance

- Recall the following example
    - `queue`, with methods `insert-rear`, `delete-front`
    - `stack`, with methods `insert-front`, `delete-front`
    - `deque`, with methods `insert-front`, `delete-front`, `insert-rear`, `delete-rear`
- What are the subtype and inheritance relationships between these classes?

Subtyping vs inheritance

- Recall the following example
  - `queue`, with methods `insert-rear`, `delete-front`
  - `stack`, with methods `insert-front`, `delete-front`
  - `deque`, with methods `insert-front`, `delete-front`, `insert-rear`, `delete-rear`
- What are the subtype and inheritance relationships between these classes?
- Subtyping
  - `deque` has more functionality than `queue` or `stack`
  - `deque` is a subtype of both these types

Subtyping vs inheritance

- Recall the following example
  - `queue`, with methods `insert-rear`, `delete-front`
  - `stack`, with methods `insert-front`, `delete-front`
  - `deque`, with methods `insert-front`, `delete-front`, `insert-rear`, `delete-rear`
- What are the subtype and inheritance relationships between these classes?
- Subtyping
  - `deque` has more functionality than `queue` or `stack`
  - `deque` is a subtype of both these types
- Inheritance
  - Can suppress two functions in a `deque` and use it as a `queue` or `stack`
  - Both `queue` and `stack` inherit from `deque`

Subclasses, subtyping and inheritance

- Class hierarchy represents both subtyping and inheritance

Subclasses, subtyping and inheritance

- Class hierarchy represents both subtyping and inheritance
- Subtyping
  - Compatibility of interfaces.
  - `B` is a subtype of `A` if every function that can be invoked on an object of type `A` can also be invoked on an object of type `B`.

Subclasses, subtyping and inheritance

- Class hierarchy represents both subtyping and inheritance
- Subtyping
  - Compatibility of interfaces.
  - `B` is a subtype of `A` if every function that can be invoked on an object of type `A` can also be invoked on an object of type `B`.
- Inheritance
  - Reuse of implementations.
  - `B` inherits from `A` if some functions for `B` are written in terms of functions of `A`.

Subclasses, subtyping and inheritance

- Class hierarchy represents both subtyping and inheritance
- Subtyping
  - Compatibility of interfaces.
  - `B` is a subtype of `A` if every function that can be invoked on an object of type `A` can also be invoked on an object of type `B`.
- Inheritance
  - Reuse of implementations.
  - `B` inherits from `A` if some functions for `B` are written in terms of functions of `A`.
- Using one idea (hierarchy of classes) to implement both concepts blurs the distinction between the two