Private classes

Madhavan Mukund

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

Programming Concepts using Java

Week 4

Nested objects

- An instance variable can be a user defined type
  - `Employee` uses `Date`

```
public class Employee{
  private String name;
  private double salary;
  private Date joindate;

  ...

}

public class Date {
  private int day, month year;

  ...
}
```

Nested objects

- An instance variable can be a user defined type
  - `Employee` uses `Date`
- `Date` is a public class, also available to other classes

```
public class Employee{
  private String name;
  private double salary;
  private Date joindate;

  ...

}

public class Date {
  private int day, month year;

  ...
}
```

Nested objects

- An instance variable can be a user defined type
  - `Employee` uses `Date`
- `Date` is a public class, also available to other classes
- When could a private class make sense?

```
public class Employee{
  private String name;
  private double salary;
  private Date joindate;

  ...

}

public class Date {
  private int day, month year;

  ...
}
```

Nested objects

- LinkedList is built using Node

```
public class Node {
  public Object data;
  public Node next;
  ...
}

public class LinkedList{
  private int size;
  private Node first;

  public Object head(){
    Object returnval = null;
    if (first != null){
      returnval = first.data;
      first = first.next;
    }
    return(returnval);
  }
}
```

Nested objects

- LinkedList is built using Node
- Why should Node be public?
  - May want to enhance with prev field, doubly linked list
  - Does not affect interface of LinkedList

```
public class Node {
  public Object data;
  public Node next;
  ...
}

public class LinkedList{
  private int size;
  private Node first;

  public Object head(){
    Object returnval = null;
    if (first != null){
      returnval = first.data;
      first = first.next;
    }
    return(returnval);
  }
}
```

Nested objects

- LinkedList is built using Node
- Why should Node be public?
  - May want to enhance with prev field, doubly linked list
  - Does not affect interface of LinkedList
- Instead, make Node a private class
  - Nested within LinkedList
  - Also called an inner class

```
public class LinkedList{
  private int size;
  private Node first;

  public Object head(){ ... }

  public void insert(Object newdata){
    ...
  }

  private class Node {
    public Object data;
    public Node next;
    ...
  }
}
```

Nested objects

- LinkedList is built using Node
- Why should Node be public?
  - May want to enhance with prev field, doubly linked list
  - Does not affect interface of LinkedList
- Instead, make Node a private class
  - Nested within LinkedList
  - Also called an inner class
- Objects of private class can see private components of enclosing class

```
public class LinkedList{
  private int size;
  private Node first;

  public Object head(){ ... }

  public void insert(Object newdata){
    ...
  }

  private class Node {
    public Object data;
    public Node next;
    ...
  }
}
```

Summary

- An object can have nested objects as instance variables
- In some situations, the structure of these nested objects need not be exposed
- Private classes allow an additional degree of data encapsulation

Summary

- An object can have nested objects as instance variables
- In some situations, the structure of these nested objects need not be exposed
- Private classes allow an additional degree of data encapsulation
- Combine private classes with interfaces to provide controlled access to the state of an object