lecture 31

- packages
- modifiers
  - public, private, ...
  - final, static
- big picture

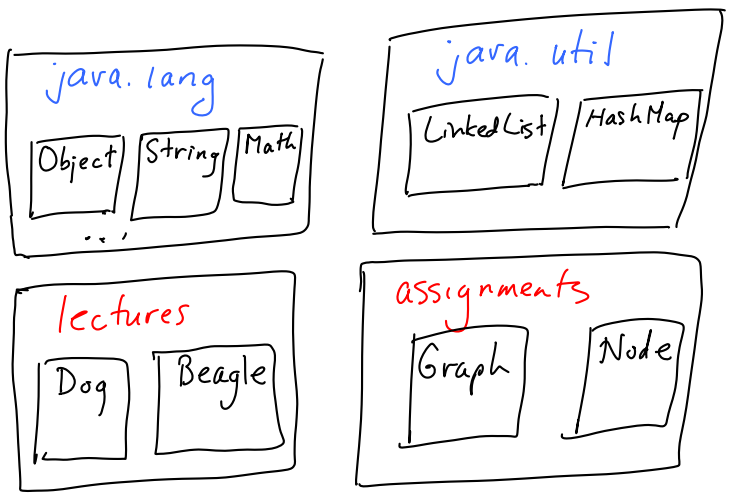
file Dog.java

```
package lectures

public class Dog {
    ⋮
}
```

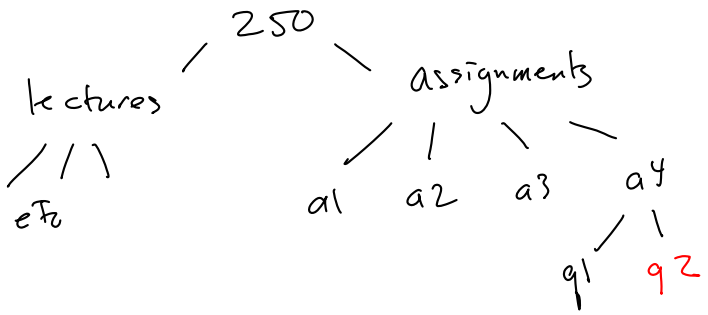
file Graph.java

```
package assignments

public class Graph {
    ⋮
}
```

Packages



package tree



e.g. 250.assignments.a4.q2

visibility / access modifiers

- public
- private
- [default]
- protected

Class Visibility

```
package lectures

public class Dog {
    ⋮
}
```

```
package assignments

public class Graph {
    ⋮
}
```

If we drop the modifier public then visibility is restricted to classes within the same package

```
package lectures
public class Dog {
    ⋮
}
```

```
package assignments
public class Graph {
    Dog myDog;
    ⋮
}
```

Compiler gives an error.

```
package lectures
public class Dog {
    ⋮
}
```

```
package assignments
public class Graph
    extends Dog {
    ⋮
}
```

Compiler gives an error.

Suppose class A is visible to class B.

e.g.
- public A
- A and B are in same package and A has package visibility

Q: can class B ...
- reference a field of A?
- invoke a method of A?

```
public class Dog {
* private String name;
  public Dog() {...}
  public String
      setName(...) {
      ... }
}
```

```
public class Person {
    ⋮
  Dog myDog;
    ⋮
  myDog = new
      Beagle();
  myDog.name
    = "Buddy"
}
```

Compiler error.
(Dog.name not visible to Person.)

```
package lectures
public class Dog {
    String name;
* // package visibility
  public Dog() {...}
  public String
      setName(...) {
      ... }
}
```

```
package lectures
public class Person {
    ⋮
  Dog myDog;
    ⋮
  myDog = new
      Beagle();
  myDog.name
    = "Buddy"
}
```

Dog.name is visible to Person.

```
package lectures
public class Dog {
  private String name;
  public Dog() {...}
    String setName(...) {
      * ... }
  private helper() {..}
```

compiler OK

```
myDog.setName(
    "Buddy")
```

compiler error

```
myDog.helper() {
    ... }.
```

```
package lectures
public class Person {
    ⋮
  Dog myDog;
    ⋮
  myDog = new
      Beagle();
```

Modifiers for methods/fields

least visible

private - only visible within class

(package) - only visible to classes in the same package.

protected - visible to classes in the same package and to subclasses in different packages

not on final exam

public - visible to every class

most visible

TODAY

- packages
- modifiers
  - public, private, ...
  - final, static

final class modifier

```
package lectures
class final Dog {
    ⋮
}
```

```
package lectures
class Beagle
  extends Dog{
    ⋮
}
```

Compiler error.
(Cannot extend a final class
e.g. Java.lang.String)

final method modifier

```
package lectures
class Dog {
void final bark(){
    ⋮
}
}
```

```
package lectures
class Beagle
  extends Dog{
  void bark(){
  }
}
```

Compiler error.
(Cannot override a final method.)

static modifier
(+ final field modifier)

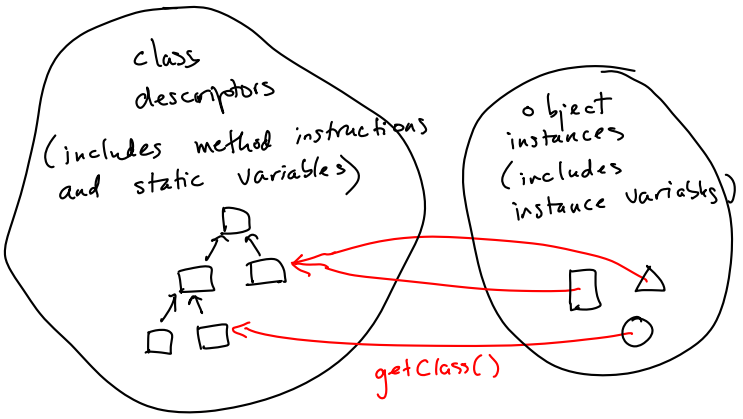
```
public static class Math{
    public static final double PI = 3.14
    public static final double E = 2.71
    ⋮
    public static double sqrt(
                double x ){
                ....
                }
}
```

```
class Dog {
    String name
    static int numDogs      // stored in the class, not in object
    Dog (){
        numDogs ++
        ⋮
    }
    static int getNumDogs () {
        return numDogs
    }
    ⋮
}
```

Big Picture of Java Virtual Machine

(Memory is 2^{32} bytes)

Motivation: this week, we have discussed many relationships between classes, methods, variables.



Q: Where are method arguments and local variables (runtime)?