Core Java

Trainer: Nilesh Ghule

Generic programming

```
static void swap(Object x, Object y)
{
    Object t = x;
    x = y;
    y = t;
    sysout ("x = " + x + ", y = ", + y);
}
p.s.v. main(String[] args){
    int i1 = 10, i2 = 20;
    swap(i1, i2);
    double d1 = 1.1, d2 = 2.2;
    swap (d1, d2);
    Person p1 = new ...(), p2 = new ...();
    swap (p1, p2);
    swap (P1, d2);
}
```

```
static <T> void swap( T x, T y)
{   T  t = x;
    x = y;
    y = z;
    sysout ("x = " + x + ", y = ", + y)
}
p.s.v. main(String[] args){
    int i1 = 10, i2 = 20;
    swap(i1, i2);
    double d1 = 1.1, d2 = 2.2;
    swap (d1, d2);
    Person p1 = new ...(), p2 = new ...();
    swap (p1, p2);
    <Double>swap (P1, d2); // error
}
```

T is placeholder for data-type.

both args should be double

C++ templates

.cpp → g++ → .o → .out

```
template <class T>
class box {
    private: T obj;
    //...
    public: T get() {
        return obj;
    }
};
int main() {
    box<int> b1;
    box<double> b2;
    box<String> b3;
    cout << b1.get();
    cout << b2.get();
    cout << b3.get();
    return 0;
}
```

```
class box<int> {
    private: int obj;
    //...
    public: int get() {
        return obj;
    }
};
class box<double> {
    private: double obj;
    //...
    public: double get() {
        return obj;
    }
};
class box<String> {
    private: string obj;
    //...
    public: String get() {
        return obj;
    }
};
```

Java generics

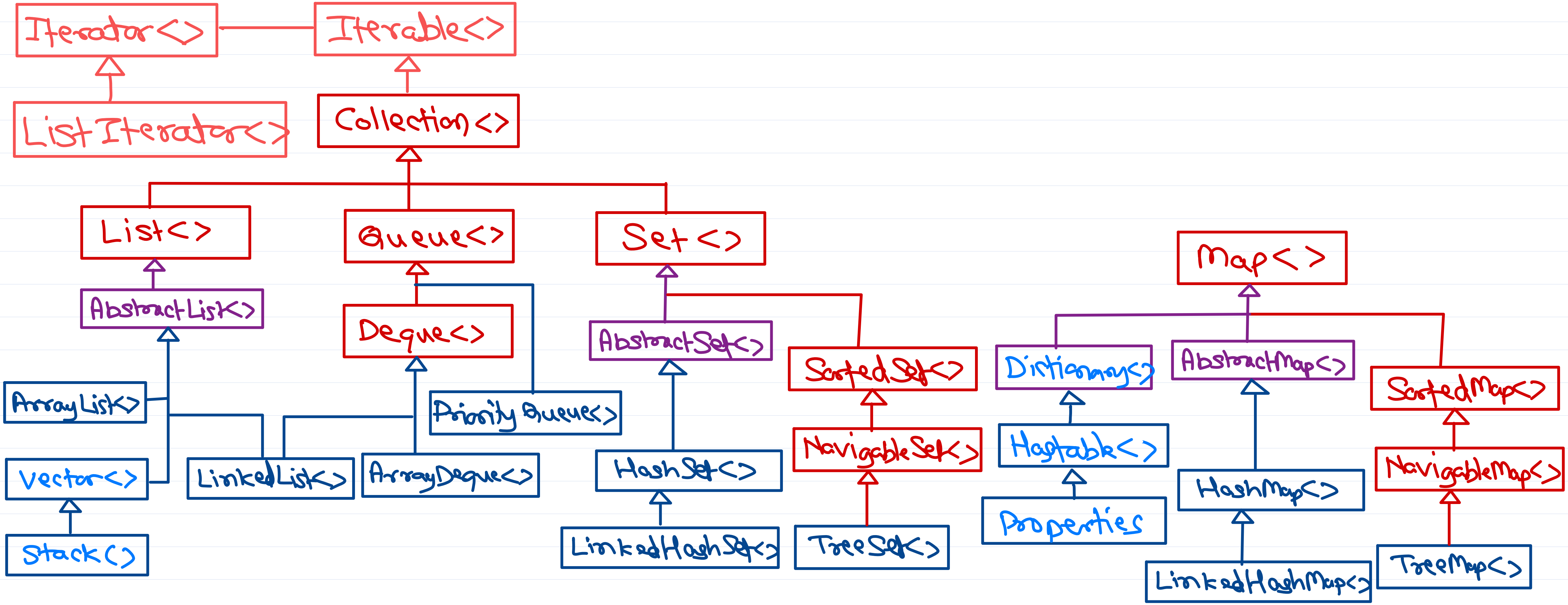
.java → javac → .class → JVM

```
class Box<T> {
    private T obj;
    //...
    public T get() {
        return obj;
    }
}
class Tester {
    p.s.v. main(—) {
        Box<Integer> b1
            = new Box<>();
        Box <Double> b2
            = new Box<>();
        Box <String> b3
            = new Box<>();
        SOP(b1.get());
        SOP(b2.get());
        SOP(b3.get());
    }
}
```

```
class Box {
    private Object obj;
    //...
    public Object get() {
        return obj;
    }
}
```

Compiler provides all required compile time checks to ensure "type-safety".

Java Collection Framework



Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>