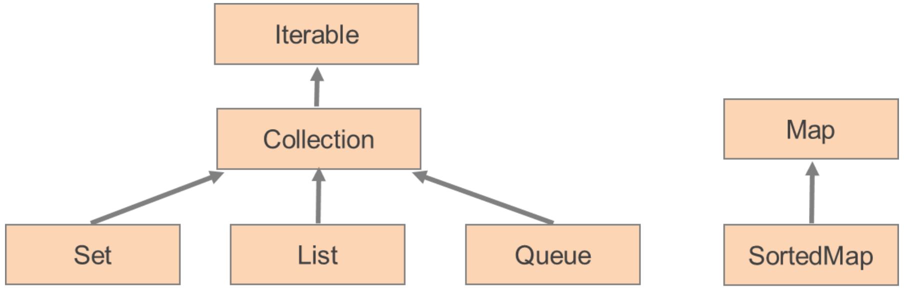
Coleções de dados (Introdução)

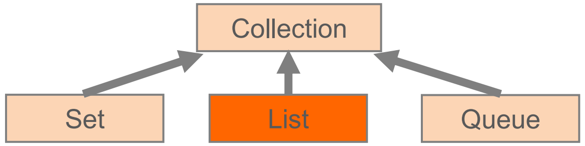
Collections *(Introdução)*

- No java existem collections
  - Similares à List, Dictionary, Set, …   do python

- **List:** *sequências com noção de ordem, com repetição*
  - ArrayList; LinkedList; Vector
- **Set:** sem noção de posição (sem ordem), sem repetição
  - HashSet; LinkedHashSet; TreeSet
- **Queue:** são as filas do tipo *First in First Out*
  - PriorityQueue; LinkedList
- **Deque**
  - ArrayDeque; LinkedList
- **Map (Interface) :** estruturas associativas onde os objectos são representados por um par chave-valor.
  - HashMap; LinkedHashMap ; TreeMap; Hashtable



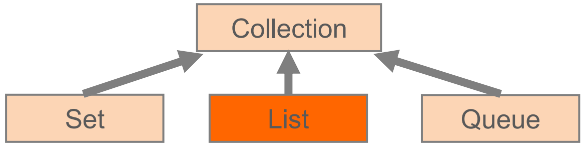
List



```
import java.util.*;

public class ListDemo {
    public static void main(String[] args) {

        List<String> list = new ArrayList<>();
        list.add("A"); list.add("B"); list.add("C");

        System.out.println(list);         // display
        System.out.println(list.get(0)); // access

        list.set(1, "X");                // update
        list.remove("C");                // remove

        for (String s : list)            // iterate
            System.out.println(s);
    }
}
```

```
def main():
    lst = [] # List<String> list = new ArrayList<>();
    lst.append("A")
    lst.append("B")
    lst.append("C")

    print(lst)      # display
    print(lst[0])   # access

    lst[1] = "X"    # update (like set)
    lst.remove("C") # remove by value (like remove("C"))

    for s in lst:   # iterate (for-each)
        print(s)

if __name__ == "__main__":
    main()
```

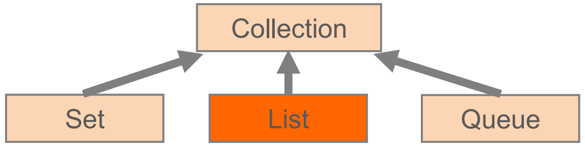
List



```java
import java.util.*;

public class ListDemo {
    public static void main(String[] args) {

        List<String> list = new ArrayList<>();
        list.add("A"); list.add("B"); list.add("C");

        System.out.println(list);         // display
        System.out.println(list.get(0)); // access

        list.set(1, "X");                 // update
        list.remove("C");                 // remove

        for (String s : list)             // iterate
            System.out.println(s);
    }
}
```

```python
def main():
    lst = [] # List<String> list = new ArrayList<>();
    lst.append("A")
    lst.append("B")
    lst.append("C")

    print(lst)      # display
    print(lst[0])   # access

    lst[1] = "X"    # update (like set)
    lst.remove("C") # remove by value (like remove("C"))

    for s in lst:   # iterate (for-each)
        print(s)

if __name__ == "__main__":
    main()
```

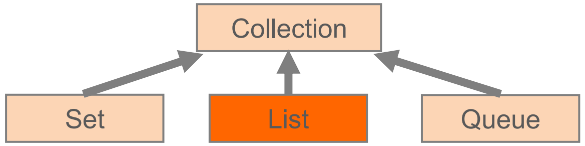
List



```
import java.util.*;

public class ListDemo {
    public static void main(String[] args) {

        List<String> list = new ArrayList<>();
        list.add("A"); list.add("B"); list.add("C");

        System.out.println(list);        // display
        System.out.println(list.get(0)); // access

        list.set(1, "X");                // update
        list.remove("C");                // remove

        for (String s : list)            // iterate
            System.out.println(s);
    }
}
```

```
def main():
    lst = [] # List<String> list = new ArrayList<>();
    lst.append("A")
    lst.append("B")
    lst.append("C")

    print(lst)      # display
    print(lst[0])   # access

    lst[1] = "X"    # update (like set)
    lst.remove("C") # remove by value (like remove("C"))

    for s in lst:   # iterate (for-each)
        print(s)

if __name__ == "__main__":
    main()
```

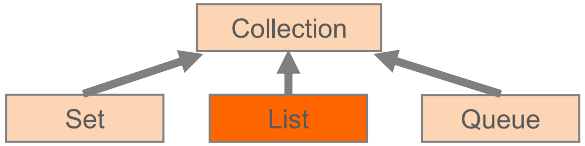
List



```java
import java.util.*;

public class ListDemo {
    public static void main(String[] args) {

        List<String> list = new ArrayList<>();
        list.add("A"); list.add("B"); list.add("C");

        System.out.println(list);        // display
        System.out.println(list.get(0)); // access

        list.set(1, "X");                // update
        list.remove("C");                // remove

        for (String s : list)            // iterate
            System.out.println(s);
    }
}
```

```python
def main():
    lst = [] # List<String> list = new ArrayList<>();
    lst.append("A")
    lst.append("B")
    lst.append("C")

    print(lst)      # display
    print(lst[0])   # access

    lst[1] = "X"    # update (like set)
    lst.remove("C") # remove by value (like remove("C"))

    for s in lst:   # iterate (for-each)
        print(s)

if __name__ == "__main__":
    main()
```

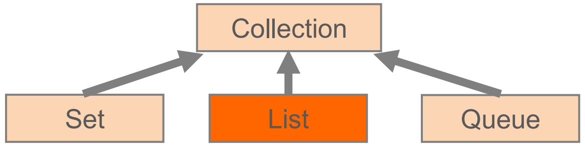
List



```java
import java.util.*;

public class ListDemo {
    public static void main(String[] args) {

        List<String> list = new ArrayList<>();
        list.add("A"); list.add("B"); list.add("C");

        System.out.println(list);         // display
        System.out.println(list.get(0)); // access

        list.set(1, "X");                // update
        list.remove("C");                // remove

        for (String s : list)            // iterate
            System.out.println(s);
    }
}
```

```python
def main():
    lst = [] # List<String> list = new ArrayList<>();
    lst.append("A")
    lst.append("B")
    lst.append("C")

    print(lst)      # display
    print(lst[0])   # access

    lst[1] = "X"    # update (like set)
    lst.remove("C") # remove by value (like remove("C"))

    for s in lst:   # iterate (for-each)
        print(s)

if __name__ == "__main__":
    main()
```

List

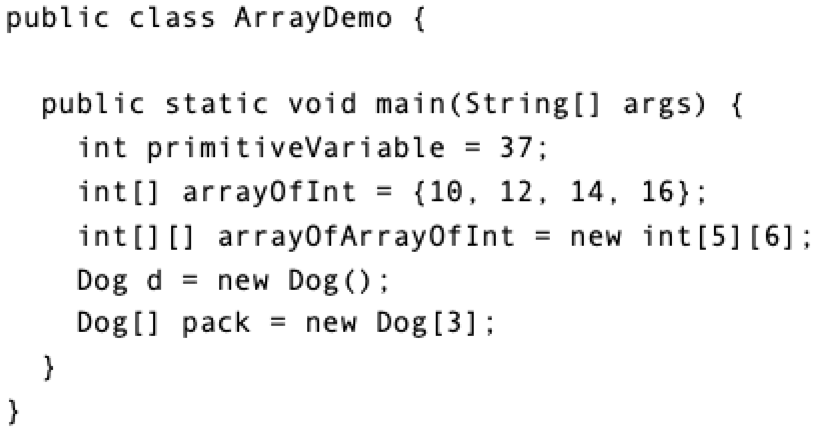


```java
import java.util.*;

public class ListDemo {
    public static void main(String[] args) {

        List<String> list = new ArrayList<>();
        list.add("A"); list.add("B"); list.add("C");

        System.out.println(list);        // display
        System.out.println(list.get(0)); // access

        list.set(1, "X");                // update
        list.remove("C");                // remove

        for (String s : list)            // iterate
            System.out.println(s);
    }
}
```

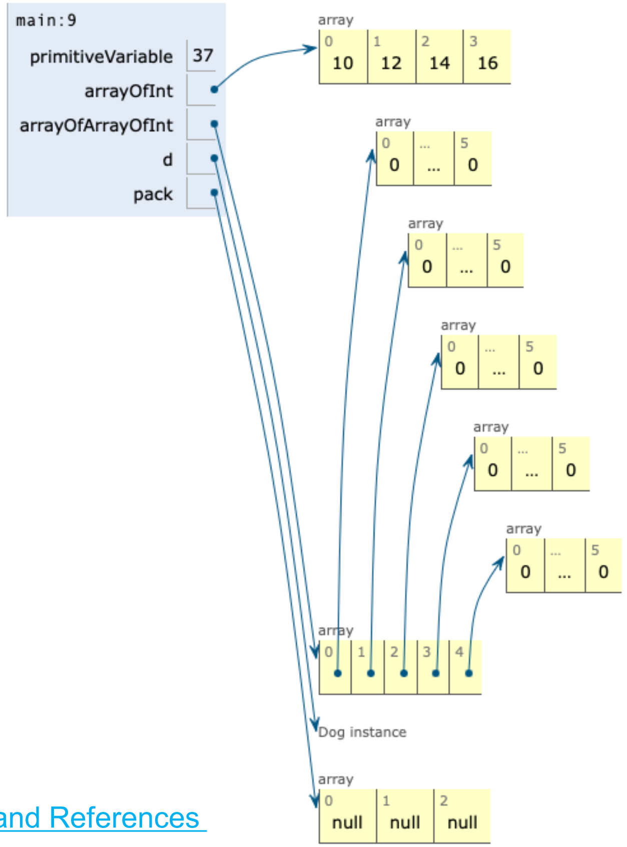
```python
def main():
    lst = [] # List<String> list = new ArrayList<>();
    lst.append("A")
    lst.append("B")
    lst.append("C")

    print(lst)      # display
    print(lst[0])   # access

    lst[1] = "X"    # update (like set)
    lst.remove("C") # remove by value (like remove("C"))

    for s in lst:   # iterate (for-each)
        print(s)

if __name__ == "__main__":
    main()
```

Tipos referenciados

- Variáveis destes tipos não contêm os valores mas os endereços para acesso aos valores efetivos



```
public class ArrayDemo {

  public static void main(String[] args) {
    int primitiveVariable = 37;
    int[] arrayOfInt = {10, 12, 14, 16};
    int[][] arrayOfArrayOfInt = new int[5][6];
    Dog d = new Dog();
    Dog[] pack = new Dog[3];
  }
}
```



- Incluem:
  - Vetores (arrays)
  - Objetos

Saber mais: Head First Java, Chapter 3. Know Your Variables: Primitives and References

Sumário

- Java Collections
  - Como usar Listas
- Tipos de dados referenciados

- Mais detalhes em aula posterior.