

Iteradores

La idea detrás de los iteradores es el patrón de recorrido de colecciones. Pensemos en las aplicaciones móviles más usadas como un cliente de mensajería, o un cliente de redes sociales y notaremos que su interfaz está compuesta principalmente por listas.

Esta necesidad motiva la implementación de estructuras repetitivas “for each” en los principales lenguajes de programación, las cuales nos permiten recorrer diferentes tipos de colecciones como arreglos o listas de manera transparente.

For each

Considerando la siguiente variable de tipo vector:

```
vector<string> cadenas;
```

Podemos recorrer esta colección haciendo uso de la estructura “for each”

```
for (string s : cadenas) {  
    cout << s << endl;  
}
```

Esta estructura hace lo mismo que la siguiente implementación usando la estructura repetitiva “for” clásica:

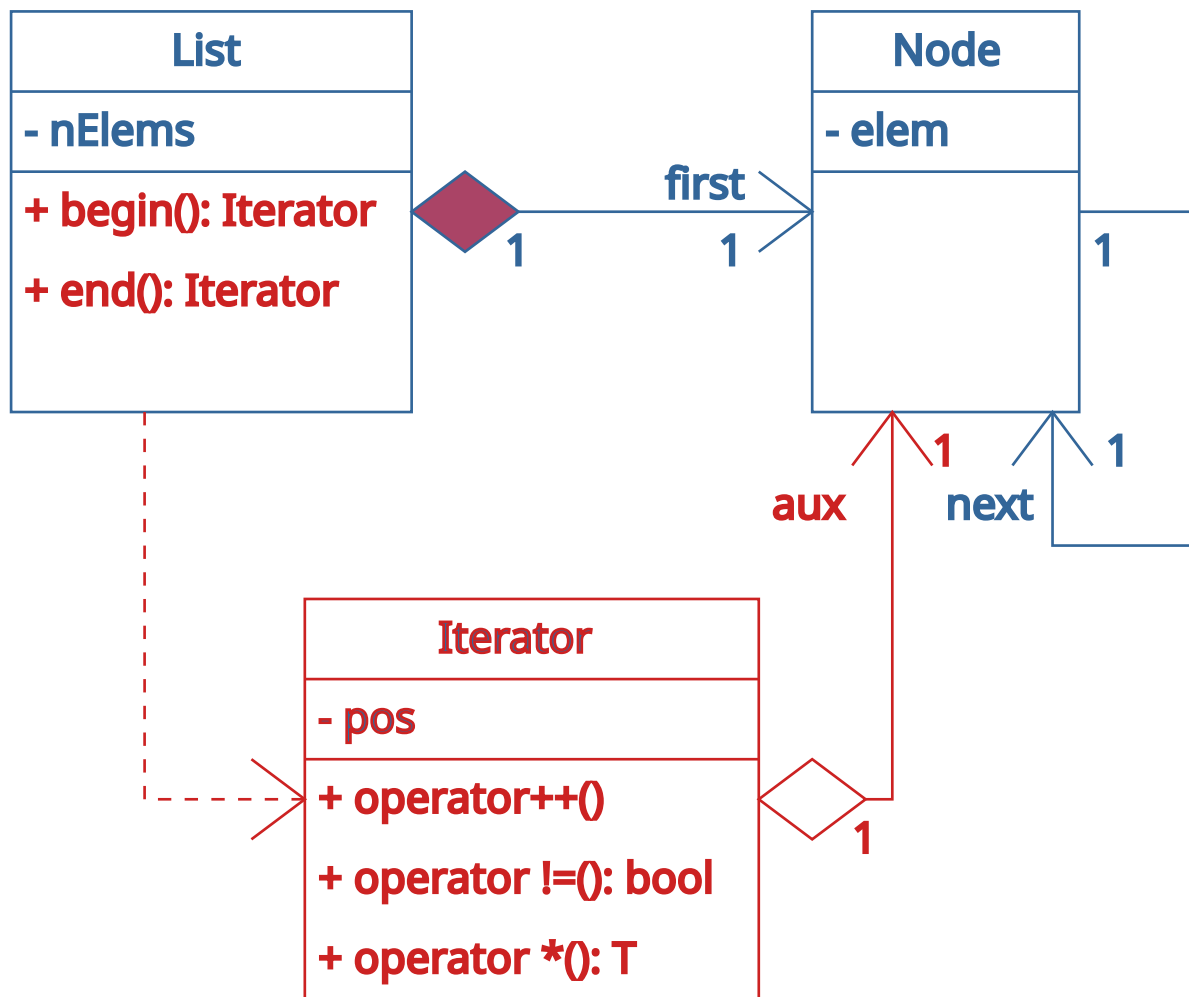
```
for (int i = 0; i < cadenas.size(); i++) {  
    string s = cadenas[i];  
    cout << s << endl;  
}
```

La estructura “for each” hace uso de **Iteradores** para realizar su trabajo e internamente oculta (a esto se le llama *syntax*) la siguiente estructura equivalente:

```
for (vector<string>::iterator it = cadenas.begin(); it != cadenas.end(); ++it) {  
    string s = *it;  
    cout << s << endl;  
}
```

Diagrama de clase

El código anterior sugiere el siguiente diagrama de clases:



- La clase Iterator, de manera similar a Node, una clase embebida pero se diferencia por ser un pública.

Código

```

class Iterator {
    unsigned int    pos;
    Node*          aux;
public:
    Iterator(unsigned int pos, Node* aux): pos(pos), aux(aux) {}
    void operator ++() { pos++; aux = aux->next; }
    bool operator !=(Iterator x) { return pos != x.pos; }
    T operator *() { return aux->elem; }
};

Iterator begin() {
    return Iterator(0, first);
}

Iterator end() {
    return Iterator(nElem, nullptr);
}
  
```

Actividad en parejas

- Diseñe los diagramas de clases incluyendo iteradores para todos los tipos de lista vistos.
- Implemente los iteradores para las listas según indicación del profesor.
- Presente y explique su implementación a la clase.

- Con ayuda del delegado, compartan todas las implementaciones con toda la clase.