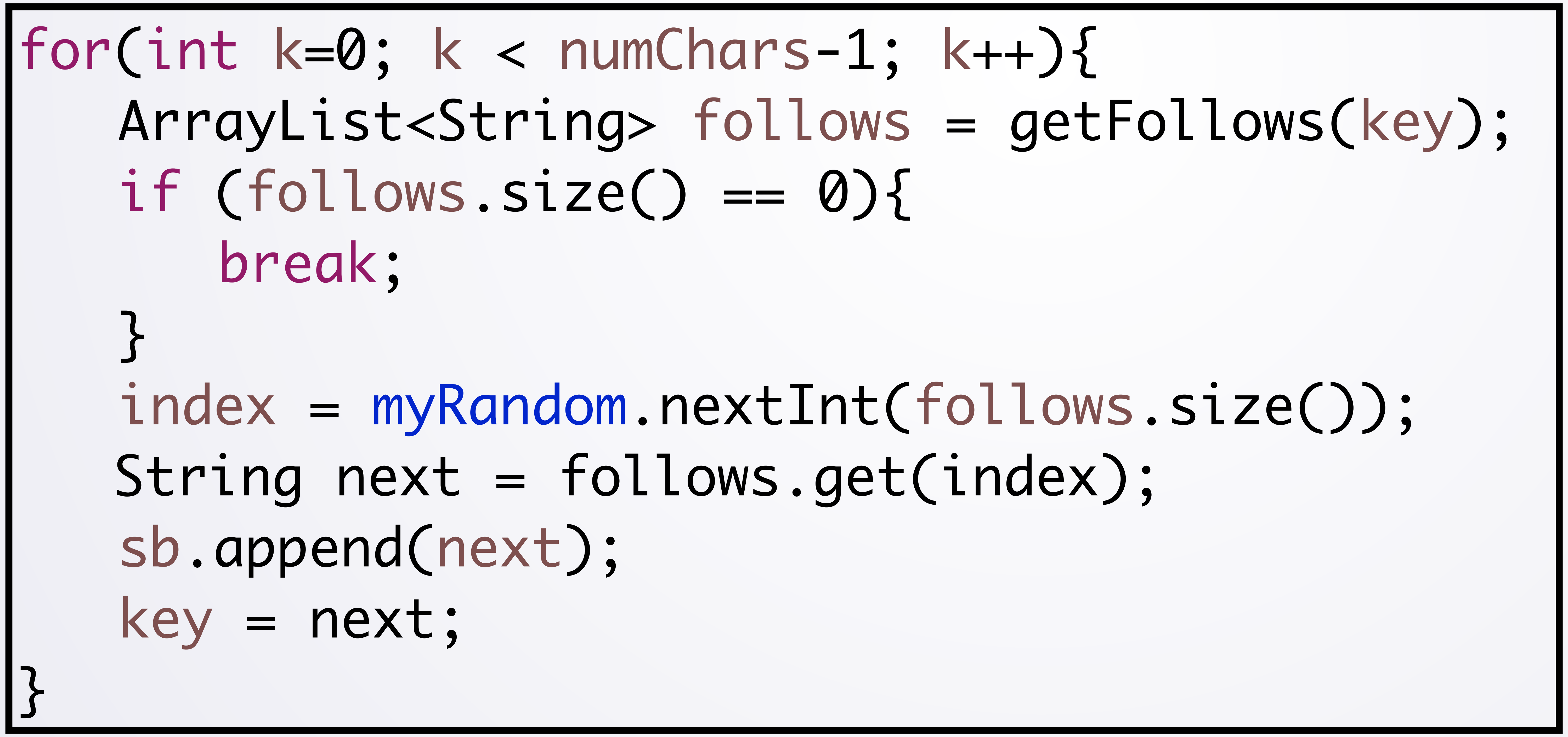
Generating Random Text

Implementing Order-Two

Duke UNIVERSITY

From MarkovOne to MarkovTwo

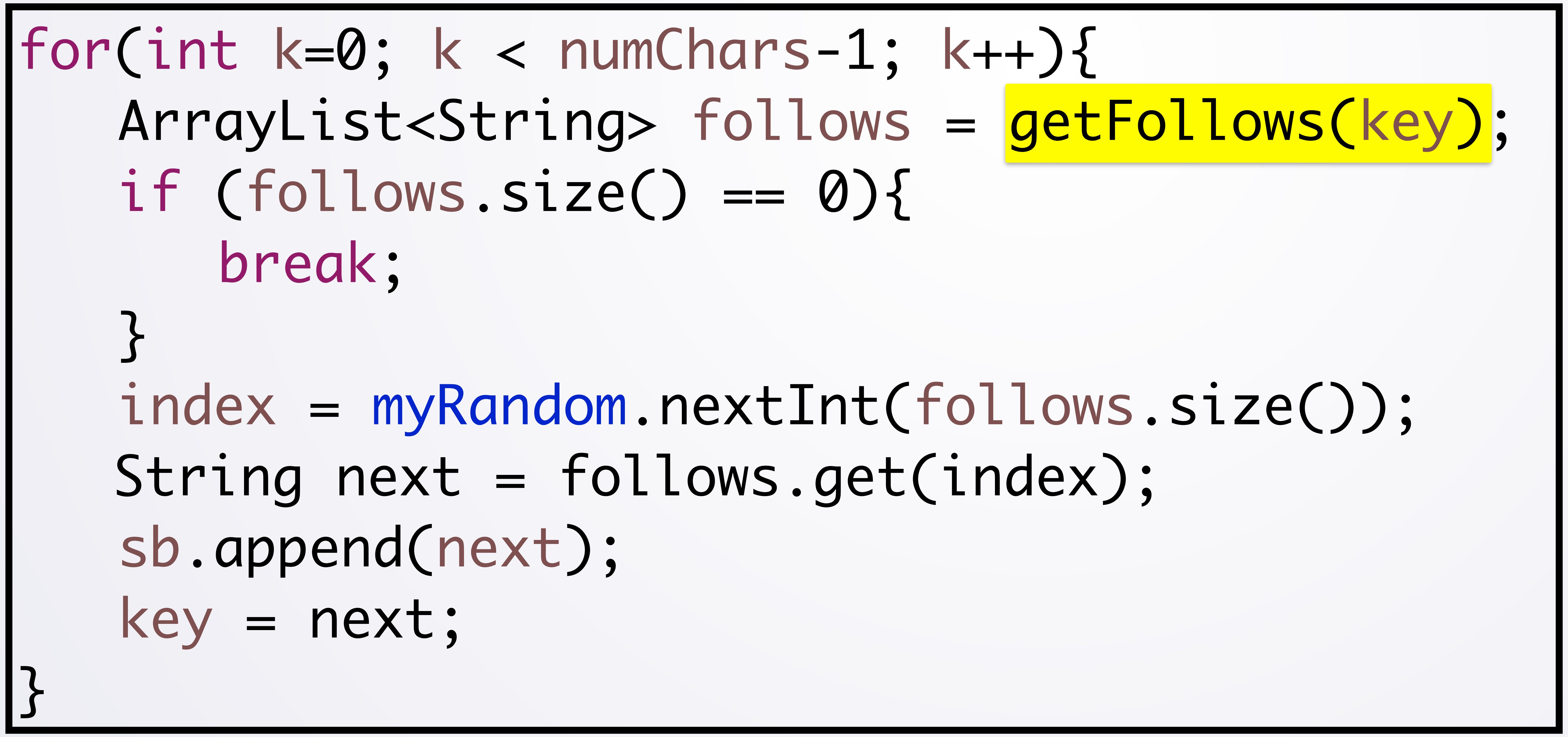
- MarkovOne **getRandomtext** calls **getFollows** method



```
for(int k=0; k < numChars-1; k++){
   ArrayList<String> follows = getFollows(key);
   if (follows.size() == 0){
      break;
   }
   index = myRandom.nextInt(follows.size());
   String next = follows.get(index);
   sb.append(next);
   key = next;
}
```

From MarkovOne to MarkovTwo

- MarkovOne **`getRandomtext`** calls **`getFollows`** method



```
for(int k=0; k < numChars-1; k++){
   ArrayList<String> follows = getFollows(key);
   if (follows.size() == 0){
      break;
   }
   index = myRandom.nextInt(follows.size());
   String next = follows.get(index);
   sb.append(next);
   key = next;
}
```

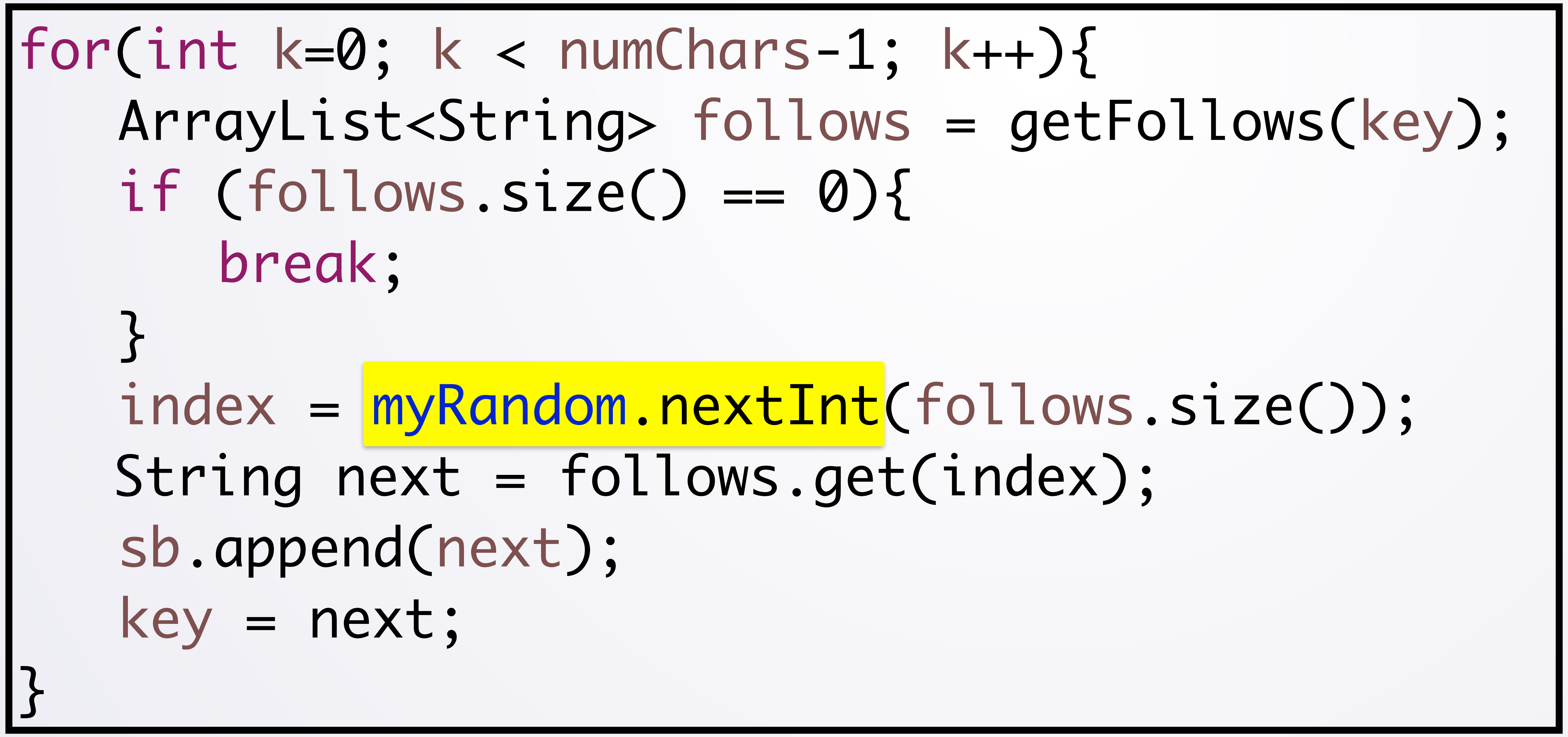
From MarkovOne to MarkovTwo

- MarkovOne **`getRandomtext`** calls **`getFollows`** method
    - Exit or add random follow to String

```
for(int k=0; k < numChars-1; k++){
    ArrayList<String> follows = getFollows(key);
    if (follows.size() == 0){
        break;
    }
    index = myRandom.nextInt(follows.size());
    String next = follows.get(index);
    sb.append(next);
    key = next;
}
```

From MarkovOne to MarkovTwo

- MarkovOne **`getRandomtext`** calls **`getFollows`** method
    - Exit or add random follow to String



```
for(int k=0; k < numChars-1; k++){
    ArrayList<String> follows = getFollows(key);
    if (follows.size() == 0){
        break;
    }
    index = myRandom.nextInt(follows.size());
    String next = follows.get(index);
    sb.append(next);
    key = next;
}
```

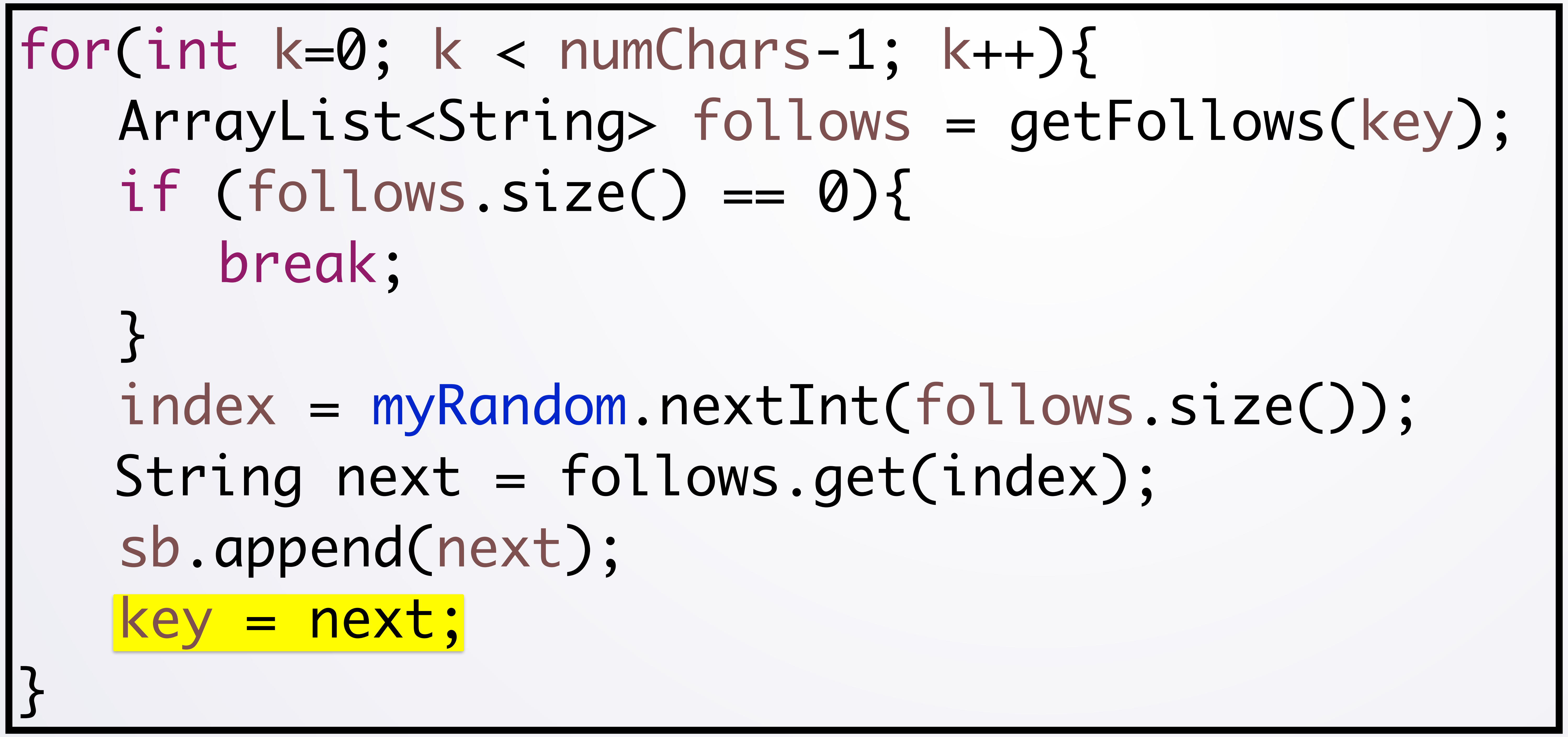
From MarkovOne to MarkovTwo

- MarkovOne **`getRandomtext`** calls **`getFollows`** method
    - Exit or add random follow to String

```
for(int k=0; k < numChars-1; k++){
   ArrayList<String> follows = getFollows(key);
   if (follows.size() == 0){
      break;
   }
   index = myRandom.nextInt(follows.size());
   String next = follows.get(index);
   sb.append(next);
   key = next;
}
```

From MarkovOne to MarkovTwo

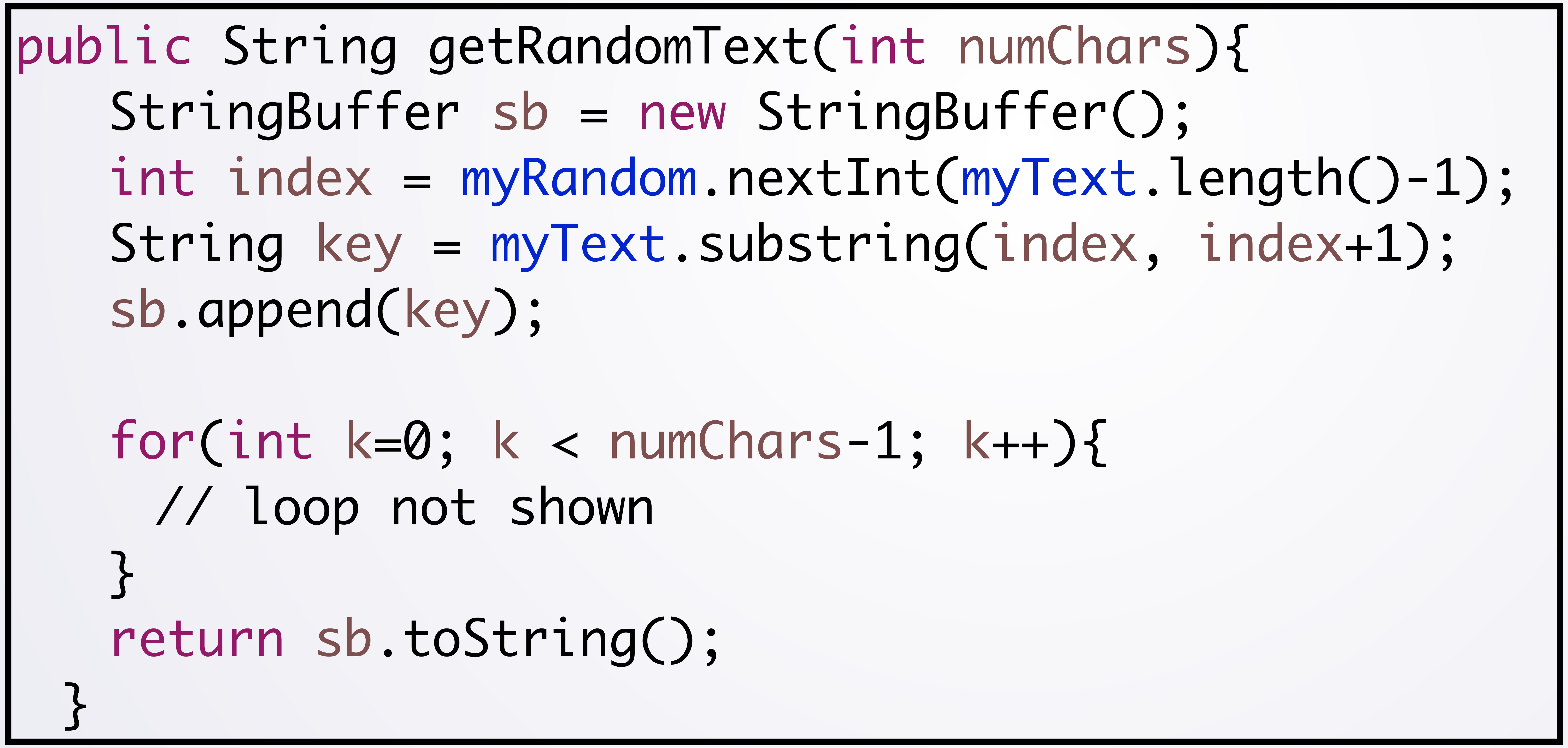
- MarkovOne **`getRandomtext`** calls **`getFollows`** method
    - Exit or add random follow to String
    - Repeat with random String as next key



```
for(int k=0; k < numChars-1; k++){
    ArrayList<String> follows = getFollows(key);
    if (follows.size() == 0){
        break;
    }
    index = myRandom.nextInt(follows.size());
    String next = follows.get(index);
    sb.append(next);
    key = next;
}
```

Details of getRandomText

- MarkovOne **`getRandomText`** code



```
public String getRandomText(int numChars){
   StringBuffer sb = new StringBuffer();
   int index = myRandom.nextInt(myText.length()-1);
   String key = myText.substring(index, index+1);
   sb.append(key);

   for(int k=0; k < numChars-1; k++){
     // loop not shown
   }
   return sb.toString();
 }
```

Details of getRandomText

- MarkovOne **getRandomText** code
  - **key** is one-character String

```
public String getRandomText(int numChars){
    StringBuffer sb = new StringBuffer();
    int index = myRandom.nextInt(myText.length()-1);
    String key = myText.substring(index, index+1);
    sb.append(key);

    for(int k=0; k < numChars-1; k++){
      // loop not shown
    }
    return sb.toString();
  }
```

Details of getRandomText

- MarkovOne **`getRandomText`** code
    - **`key`** is one-character String

```
public String getRandomText(int numChars){
   StringBuffer sb = new StringBuffer();
   int index = myRandom.nextInt(myText.length()-1);
   String key = myText.substring(index, index+1);
   sb.append(key);

   for(int k=0; k < numChars-1; k++){
    // loop not shown
   }
   return sb.toString();
 }
```

Details of getRandomText

- MarkovOne **`getRandomText`** code
  - **`key`** is one-character String
  - Think about changes to two-character

```
public String getRandomText(int numChars){
   StringBuffer sb = new StringBuffer();
   int index = myRandom.nextInt(myText.length()-1);
   String key = myText.substring(index, index+1);
   sb.append(key);

   for(int k=0; k < numChars-1; k++){
     // loop not shown
   }
   return sb.toString();
 }
```