

Milestone: eXoa: Example-oriented API

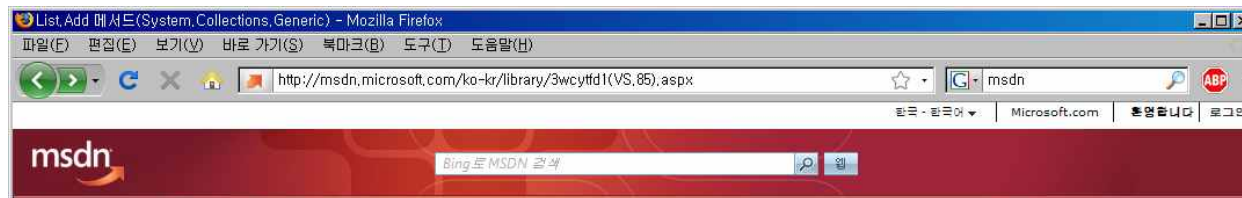
Sanghoon Lee
POSTECH

with
Jinhan Kim, Seung-won Hwang, Sunghun Kim

July 10, 2009

Contents

- Introduction
- eXoa
 - Overview
 - Using Source code repositories
 - Summarization
 - Diversification
 - Ranking
 - Embedding
- Evaluation
- Case Study
- Improvements



.NET Framework Developer Center

홈 라이브러리 학습 다운로드

인쇄용 버전 전송 콘텐츠 추가...

MSDN > MSDN Library > .NET 개발 > 이전 버전 > .NET Framework

Add 메서드

저대역폭 보기 설정

모두 축소 언어 필터: C++

List.Add 메서드

개체를 List의 끝 부분에 추가합니다.

네임스페이스: System.Collections.Generic

어셈블리: mscorlib(mscorlib.dll)

구분

```
C++  
public:  
virtual void Add (  
    T item  
) sealed
```

매개 변수

item
List의 끝에 추가할 개체입니다. 참조 형식에 대해 값은 null 참조(Visual Basic에서는 Nothing)일 수 있습니다.

설명

List은 null 참조(Visual Basic의 경우 Nothing)를 참조 형식에 대해 유효한 값으로 취급하지 않습니다. Count가 이미 Capacity와 같으면 내부 배열을 자동으로 재할당하여 List의 용량을 늘립니다. Count가 Capacity보다 작으면 이 메서드는 O(1) 연산입니다. 용량을 증가시키려면 Capacity를 늘려주세요.

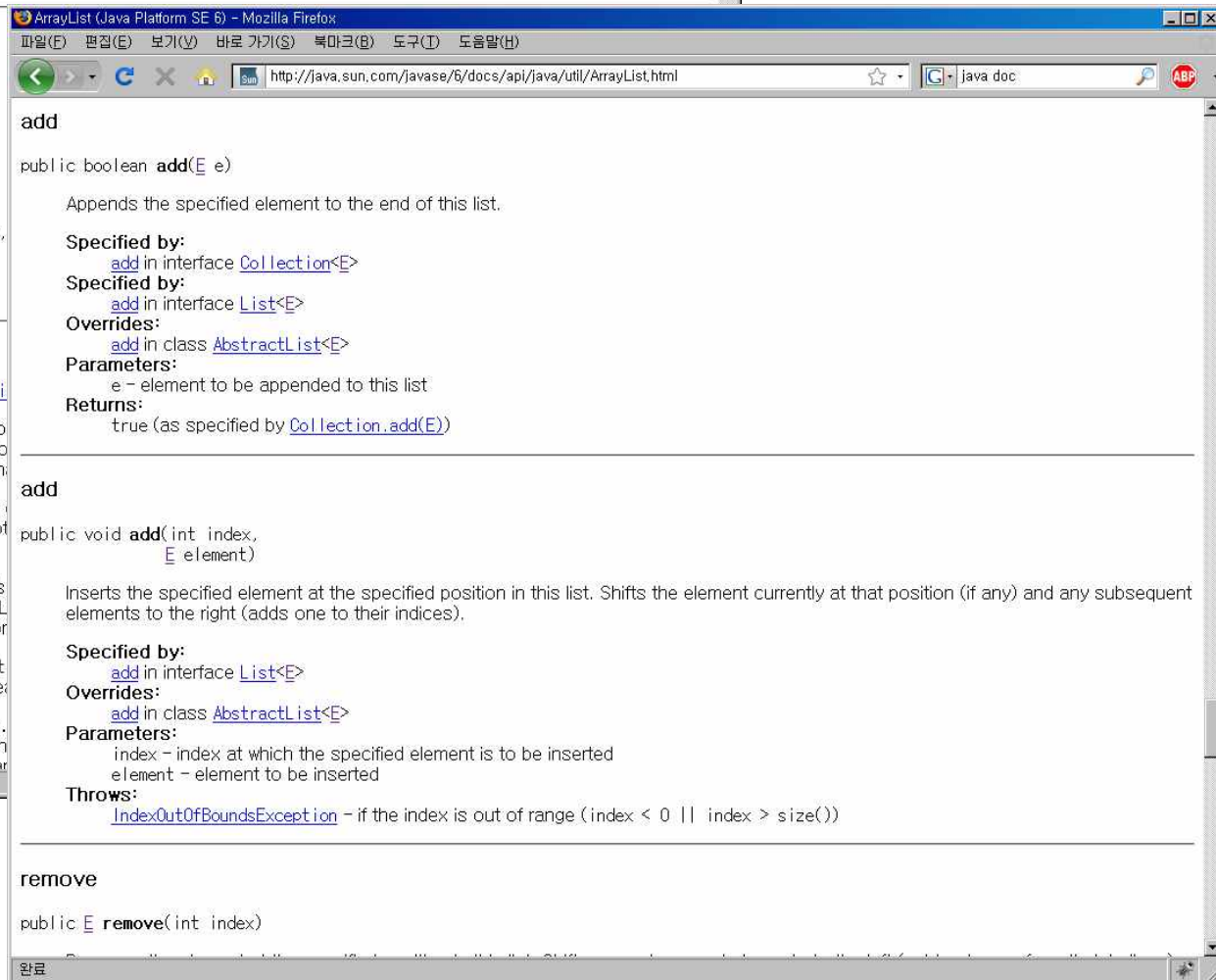
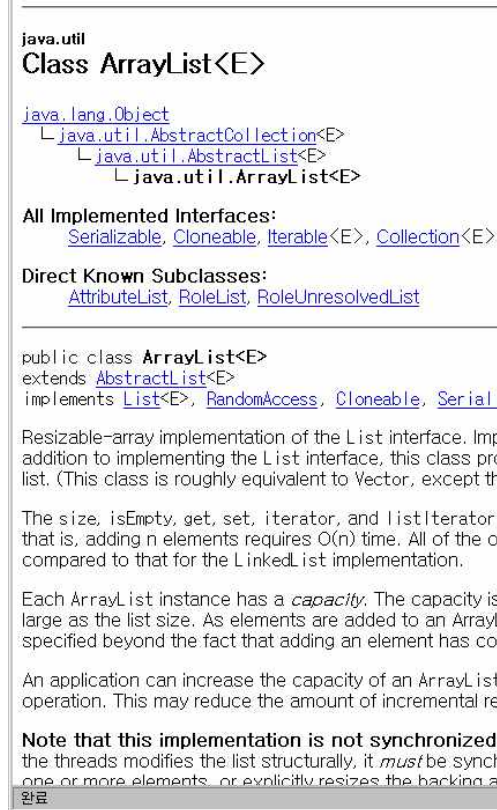
완료

```
C++  
using namespace System;  
using namespace System::Collections::Generic;  
  
void main()  
{  
    List<String^>^ dinosaurs = gcnew List<String^>();  
  
    Console::WriteLine("\nCapacity: {0}", dinosaurs->Capacity);  
  
    dinosaurs->Add("Tyrannosaurus");  
    dinosaurs->Add("Amargasaurus");  
    dinosaurs->Add("Mamenchisaurus");  
    dinosaurs->Add("Deinonychus");  
    dinosaurs->Add("Compsognathus");  
  
    Console::WriteLine();  
    for each(String^ dinosaur in dinosaurs )  
    {  
        Console::WriteLine(dinosaur);  
    }  
  
    Console::WriteLine("\nCapacity: {0}", dinosaurs->Capacity);  
    Console::WriteLine("Count: {0}", dinosaurs->Count);  
  
    Console::WriteLine("\nContains(\"Deinonychus\"): {0}",  
        dinosaurs->Contains("Deinonychus"));  
  
    Console::WriteLine("\nInsert(2, \"Compsognathus\")");  
    dinosaurs->Insert(2, "Compsognathus");  
  
    Console::WriteLine();  
}
```

완료

Javadoc

No examples !!!



Javadoc + Examples = Example Oriented API documentation

Javadoc

add

```
public boolean add(E e)
```

Appends the specified element to the end of this list.

Specified by:

[add](#) in interface [Collection<E>](#)

Specified by:

[add](#) in interface [List<E>](#)

Overrides:

[add](#) in class [AbstractList<E>](#)

Parameters:

e - element to be appended to this list

Returns:

true (as specified by [Collection.add\(E\)](#))

add

```
public void add(int index,  
                E element)
```

Inserts the specified element at the specified position in this list. Shifts the elements to the right (adds one to their indices).

Specified by:

[add](#) in interface [List<E>](#)

Overrides:

[add](#) in class [AbstractList<E>](#)

Parameters:

index - index at which the specified element is to be inserted

element - element to be inserted

Throws:

Examples

```
String name;  
ArrayList acc = new ArrayList(headers.size());  
for (Iterator i = headers.iterator(); i.hasNext();) {  
    InternetHeader header = (InternetHeader) i.next();  
    if (header.equals(name) && header.line != null) {  
        acc.add(header.getValue());  
    }  
}
```

Good Examples

- Simplicity

```
public String[] getHeader(String name) {  
    ArrayList acc = new ArrayList(headers.size());  
    for (Iterator i = headers.iterator(); i.hasNext();) {  
        InternetHeader header = (InternetHeader) i.next();  
        if (header.equals(name) && header.line != null) {  
            acc.add(header.getValue());  
        }  
    }  
    int size = acc.size();  
    if (size == 0) {  
        return null;  
    }  
    String[] h = new String[size];  
    acc.toArray(h);  
    return h;  
}
```



```
String name;  
ArrayList acc = new ArrayList(headers.size());  
for (Iterator i = headers.iterator(); i.hasNext();) {  
    InternetHeader header = (InternetHeader) i.next();  
    if (header.equals(name) && header.line != null) {  
        acc.add(header.getValue());  
    }  
}
```

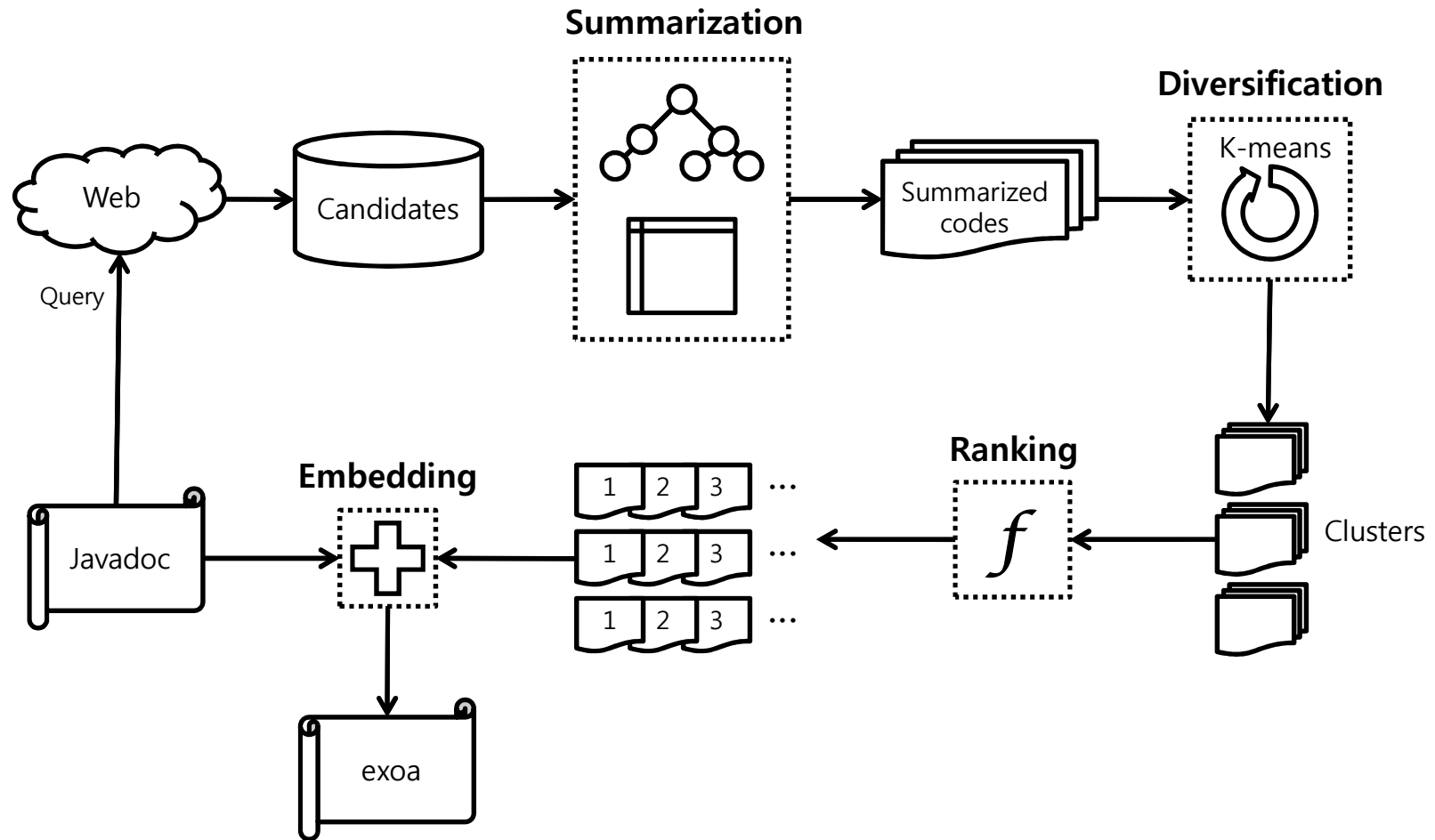
- Various Usages

```
ArrayList list = new ArrayList();  
list.add("one");  
list.add("two");  
list.add("three");
```

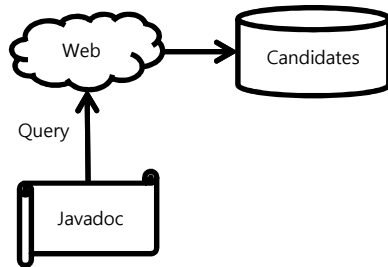


```
ArrayList list = new ArrayList();  
for (int i = 0; i < original.length; i++) {  
    if (!original[i].equals(element)) {  
        list.add(original[i]);  
    }  
}
```

Overview of eXoa



Using Source code repositories



Javadoc

```
ArrayList class
boolean add(E e)
void add(int index, E element)
boolean addAll(Collection c)
boolean addAll(int index, Collection c)
void clear()
...
```



SortedList.java

```
}
/**
 * Adds the specified object in proper location to keep list sorted.
 * @param elementToAdd the element to be added
 */
```

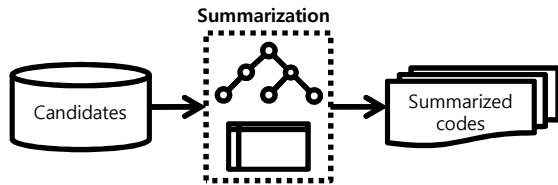
SSLSocket.java

```
import java.io.PrintStream;
import java.math.BigInteger;
import java.net.InetAddress;
import java.net.Socket;
import java.net.SocketAddress;
```

TopologicalSorting.java

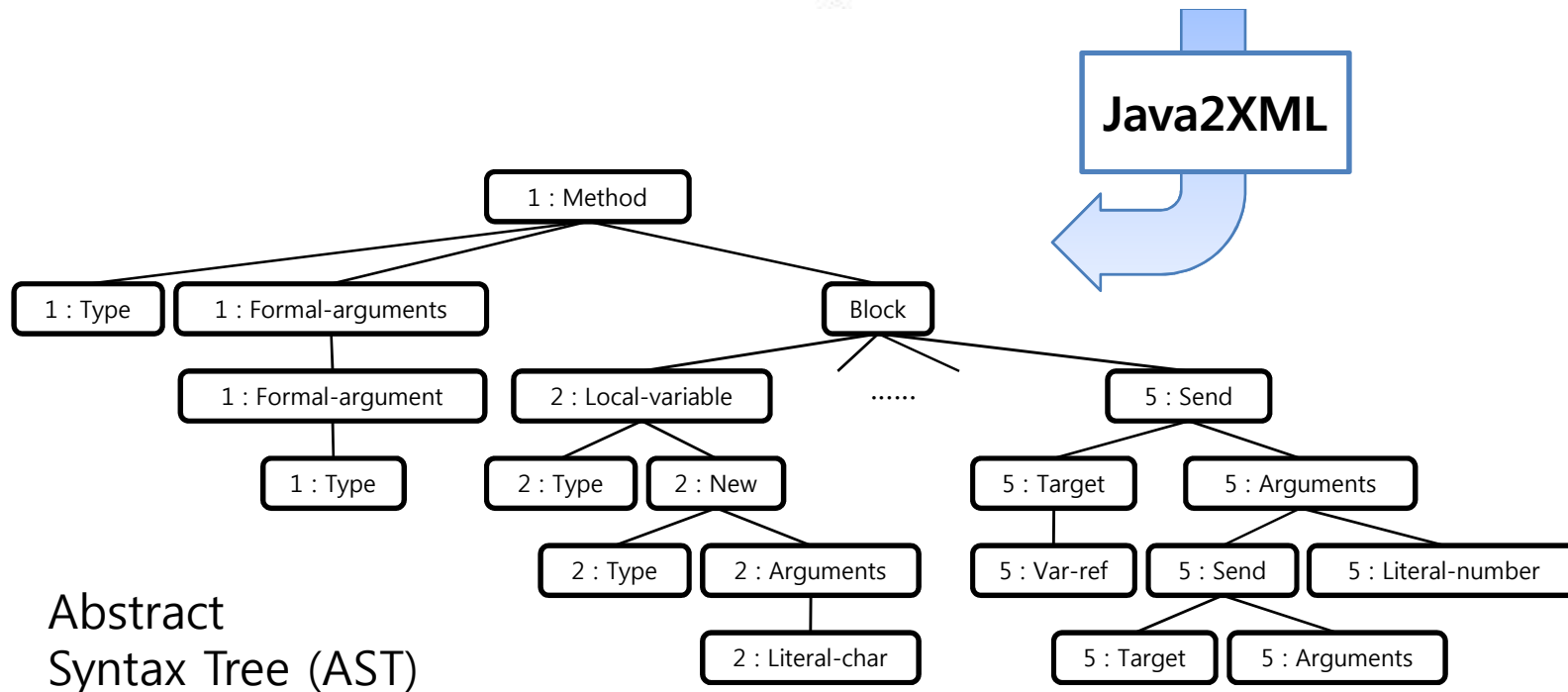
```
}
// We're guaranteed a single root by the above construction.
List rootList = new ArrayList();
rootList.add(startat);
// Finally, perform the topological sort.
```


Summarization



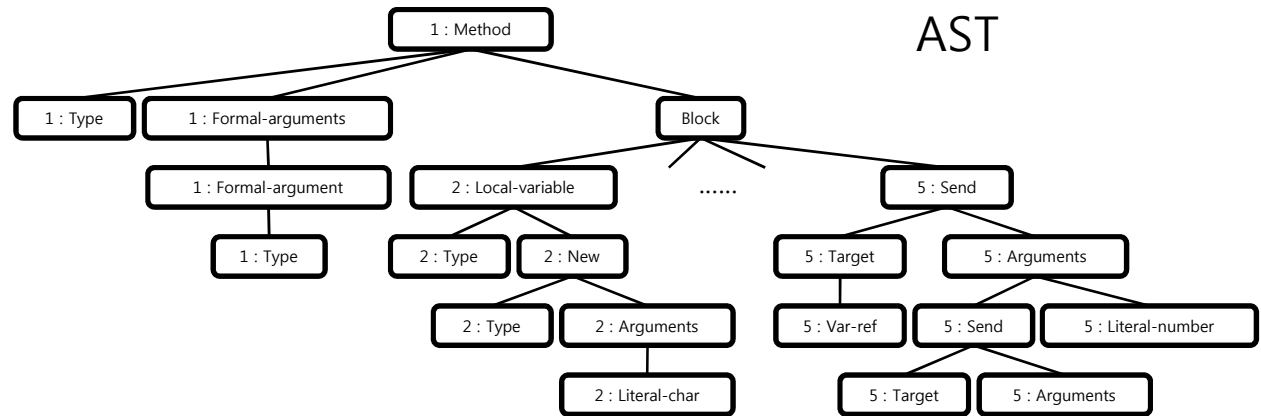
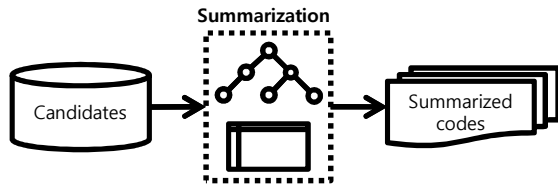
Method-based sliced code

```
public void test(TestHarness harness) {  
    Character a = new Character('\uffda');  
    Character b = new Character('Z');  
    harness.check(a.hashCode(), 65498);  
    harness.check(b.hashCode(), 90);  
}
```



Java2XML

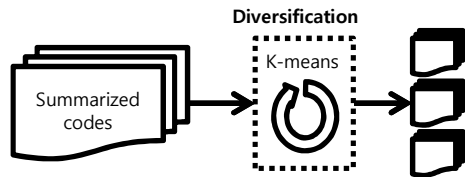
Summarization



Intra
Analysis
Table
(IAT)

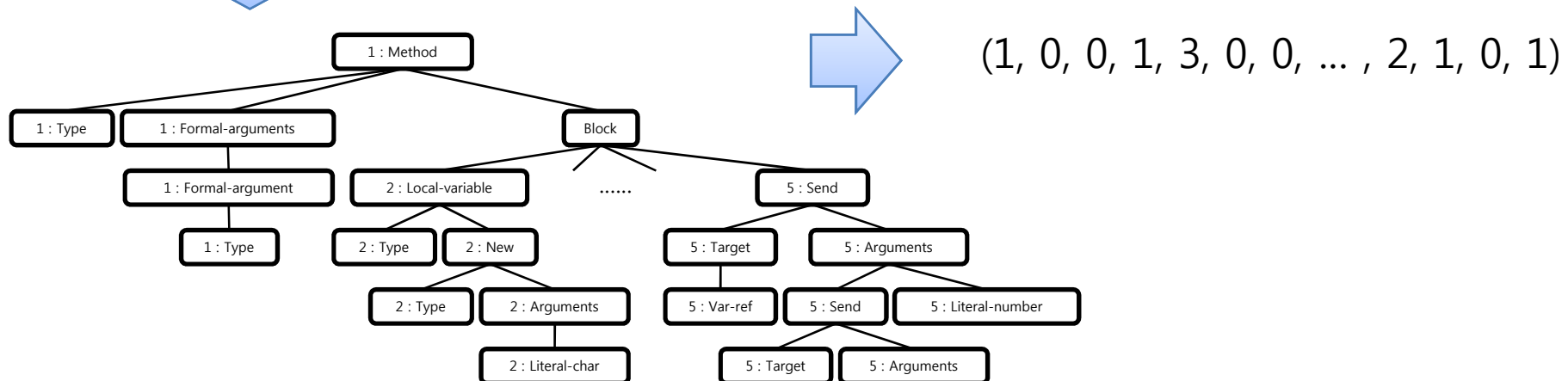
| Line | Class | API name | Arguments |
|------|---------------------------|----------|-----------|
| 4 | a : Character : 2 | hashCode | |
| 5 | b : Character : 3 | hashCode | |
| 4 | harness : TestHarness : 1 | check | int, int |
| 5 | harness : TestHarness : 1 | check | int, int |
| ... | ... | ... | ... |

Diversification

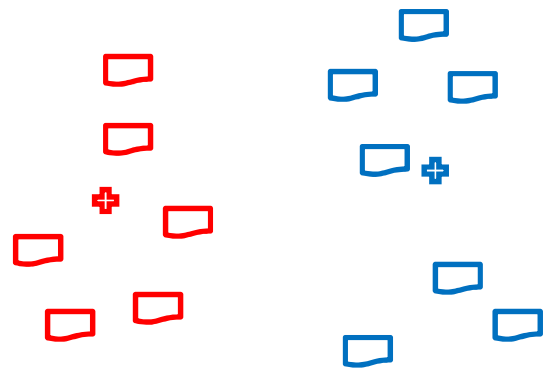
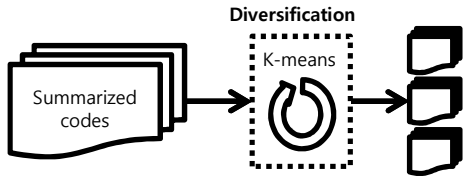


- Similarity comparison
 - String-based matching
 - Tree-based matching
 - Characteristic vector-based matching (Deckard, ICSE07)

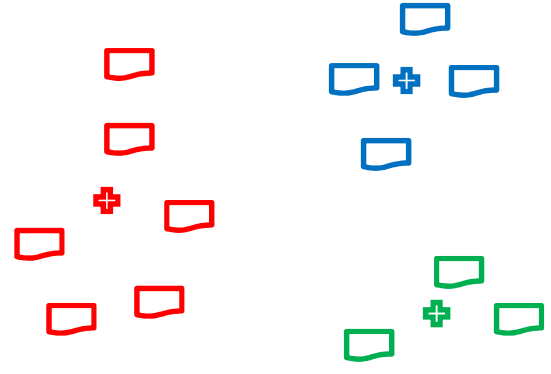
```
public void test(TestHarness harness) {  
    Character a = new Character('\uffda');  
    Character b = new Character('Z');  
    harness.check(a.hashCode(), 65498);  
    harness.check(b.hashCode(), 90);  
}
```



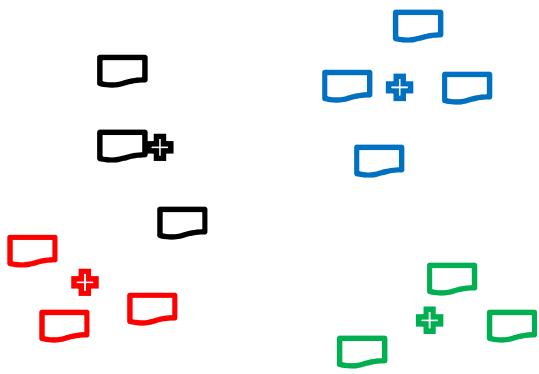
Diversification



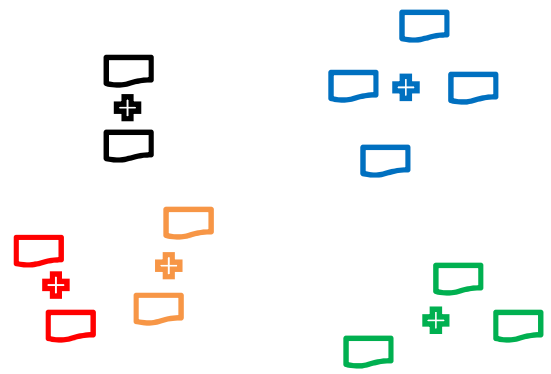
K=2



K=3

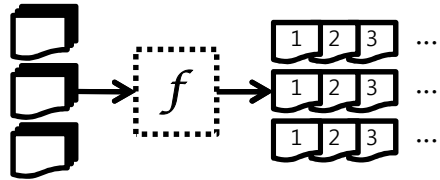


K=4



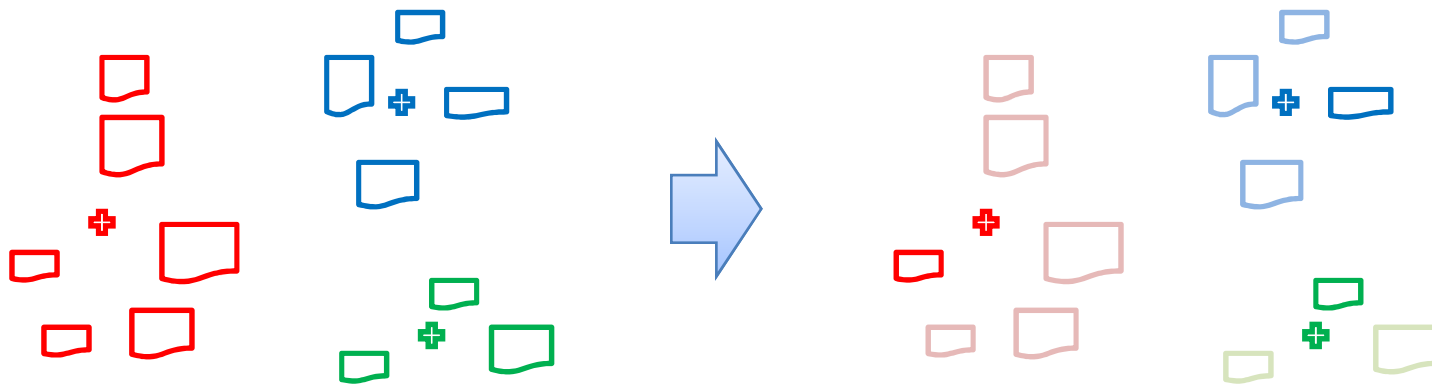
K=5

Ranking

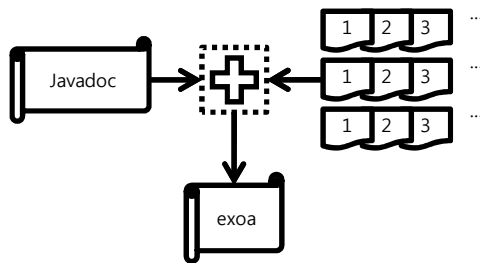


- The most representative example in a cluster
- Measures
 - Distance from centroid
 - Conciseness
 - Correctness
 - + User Feedback

K=3



Embedding



(a)

Popularity

| Method Summary | |
|--|--|
| <code>clone</code> | Creates and returns a copy of this object. |
| <code>format</code> | Formats an object and appends the resulting text to a given string buffer. |
| <code>formatToCharacterIterator</code> | Formats an Object producing an <code>IterableCharacterIterator</code> . |
| <code>parseObject</code> | Parses text from the beginning of the given string to produce an object. |
| <code>parseObject(String source, ParsePosition pos)</code> | Parses text from a string to produce an object. |

Methods inherited from class `java.lang.Object`

`equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Method Detail

(b)

Code example

```

format.parse
format(parsed)

public static boolean parseFormat(Object value, Format FORMATTER) {
    //...
    Object parsed = FORMATTER.parseObject(value);
    //...
    return FORMATTER.FORMATTER(parsed).equals(value);
}
    
```

(c)

User Feedback

Evaluation

- Comparison with Koders
 - Accuracy

| | Relevant code and snippet | Relevant code | Unrelated code |
|----------|---------------------------|---------------|----------------|
| Koders | 22% | 8% | 70% |
| eXoaDocs | 92% | 6% | 2% |

Table I

TOP-5 QUERY RESULTS COMPARISON BETWEEN KODERS AND
eXoaDOCS USING RANDOMLY SELECTED 100 APIs

- Diversification
 - Koders : 30.8% are duplicated (among 22%)
 - eXoa : only 8.7% are duplicated (among 92%)

Case Study

- Productivity

| Group | Task ₁ | Task ₂ | Task ₃ & Task ₄ |
|----------|-------------------|-------------------|---------------------------------------|
| eXoaDocs | 8:53 | 23:34 | 30:32 |
| JavaDocs | 14:40 | 25:03 | 32:03 |

Table II

AVERAGING THE CUMULATIVE COMPLETION TIME OF ONLY THOSE WHO COMPLETED THE TASK (MIN:SEC).

| Group | Total lookups | Distinct lookups | Relevant lookups | $\frac{relevant}{distinct}$ |
|----------|---------------|------------------|------------------|-----------------------------|
| eXoaDocs | 5.67 | 3.25 | 2.33 | 0.72 |
| JavaDocs | 17.58 | 7.5 | 3.25 | 0.43 |

Table III

THE AVERAGE NUMBER OF DOCUMENT LOOKUPS

- Code quality

| Group | Task ₁ | Task ₂ | Task ₃ | Task ₄ |
|----------|-------------------|-------------------|-------------------|-------------------|
| eXoaDocs | 11 | 11 | 3 | 3 |
| JavaDocs | 10 | 8 | 4 | 1 |

Table IV

THE NUMBER OF SUBJECTS PASSING THE TEST SUITE FOR EACH TASK

- Cognitive load

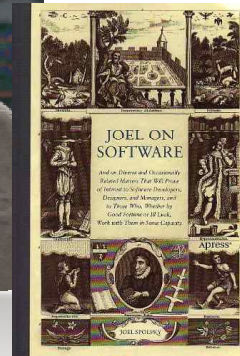
| Group | Extremely useful | Very useful | Useful | Somewhat useful | Not useful |
|----------|------------------|-------------|--------|-----------------|------------|
| eXoaDocs | 3 | 4 | 3 | 2 | 0 |
| JavaDocs | 1 | 2 | 5 | 2 | 2 |

Table V

POST-STUDY SURVEY RESULT

Developer Feedback

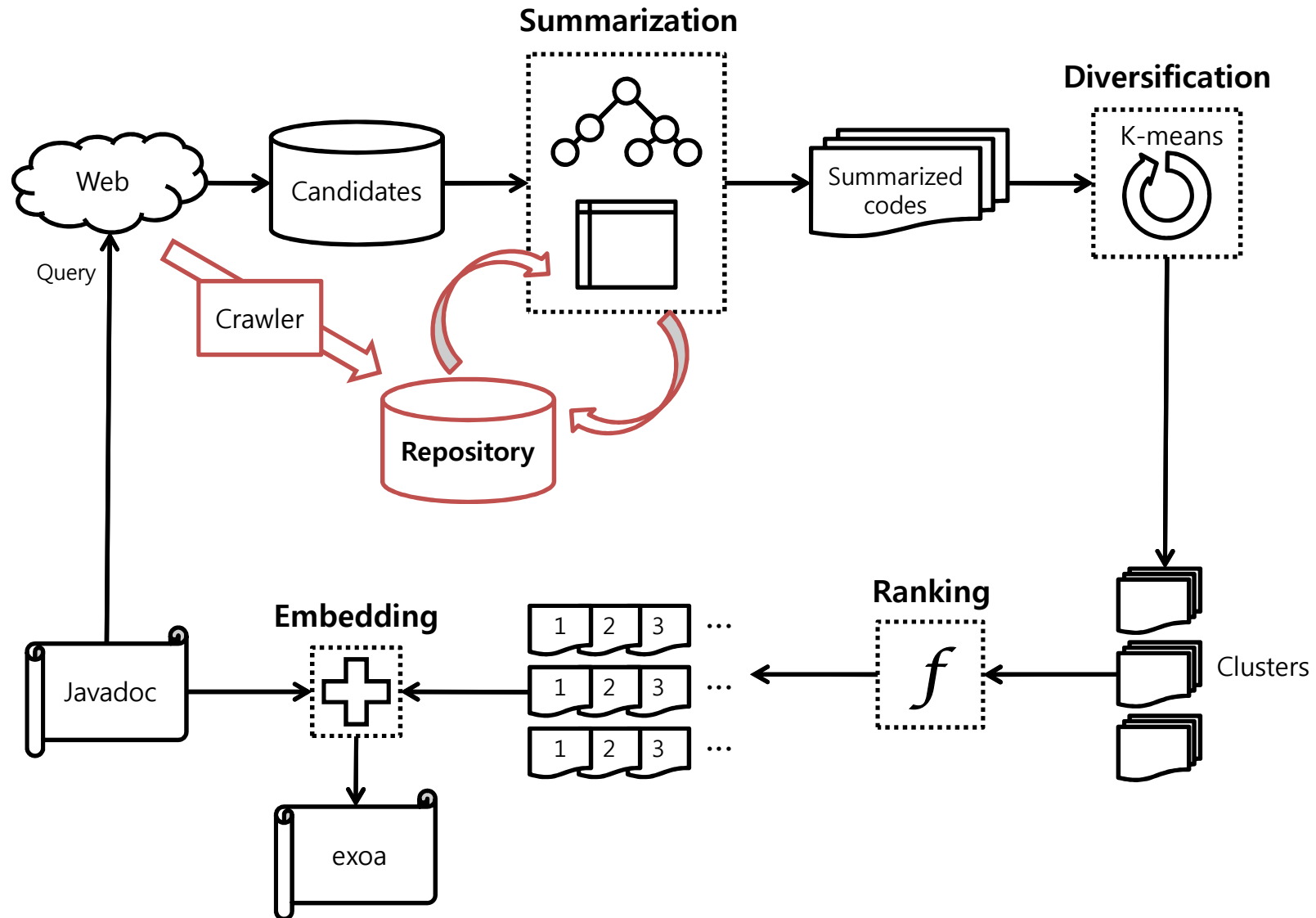
Joel Spolsky: "I think this is a **fantastic idea**. Just yesterday, I was facing this exact problem... the API documentation wasn't good enough, and I would have killed for a couple of examples. It sounds like a **very smart idea**..."



Developer 2: "Automatic example finding sounds really good. This would help developers significantly. In fact, **I was struggled many times to understand APIs without examples.**"

Developer 3: "API documents that have code example is really helpful... Using these code examples, **I can reduce develop time significantly.** However, JavaDocs provides very few code examples, so I need to find additional code examples. I like MSDN because most functions and methods contain code examples. For this reason, **automatically finding and adding code examples** seems a **wonderful idea**..."

Improvements - 1

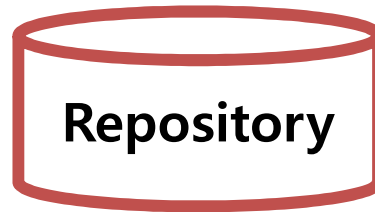


Improvements - 2

<http://koders.com/default.aspx?s=ArrayList+add&btn=&la=Java>



Web services



Open source
Repositories

Improvements - 3

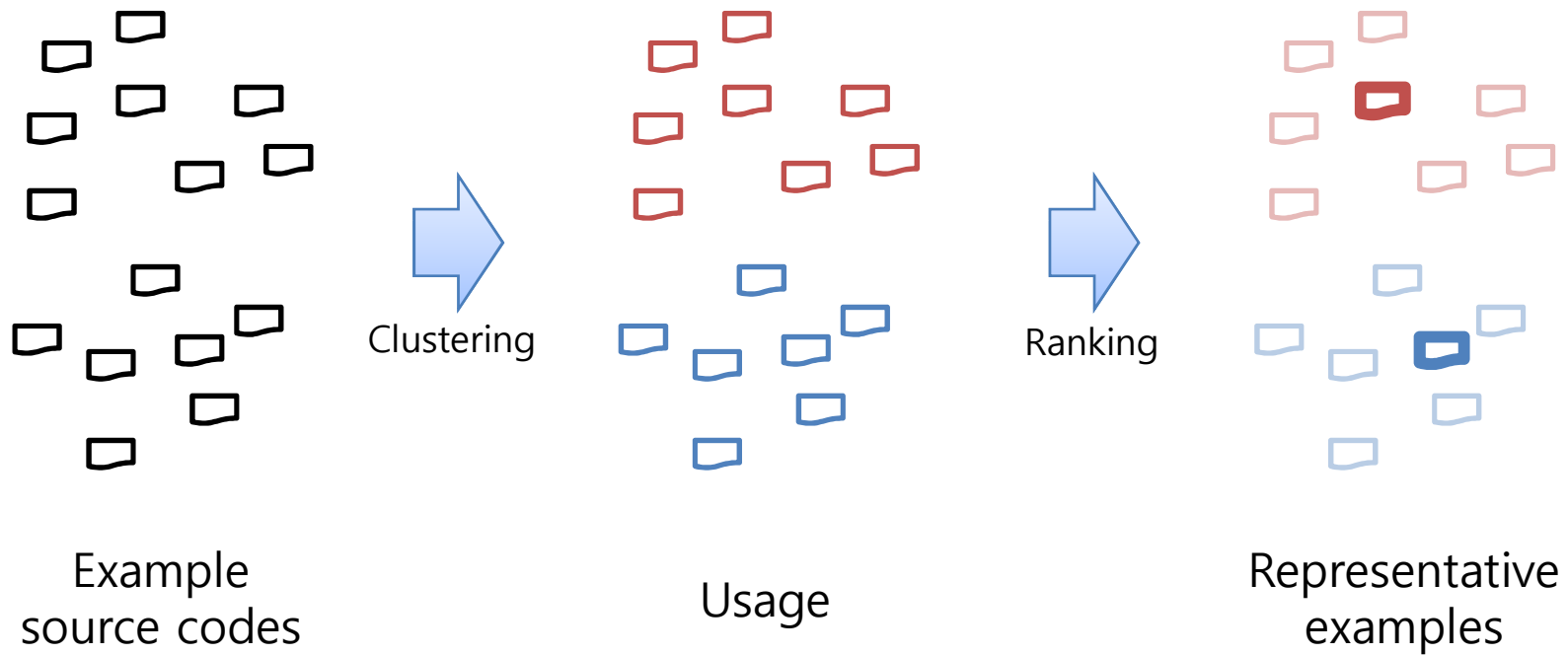
- Simplify examples on a method call

```
public String[] getHeader(String name) {  
    ArrayList acc = new ArrayList(headers.size());  
    for (Iterator i = headers.iterator(); i.hasNext();) {  
        InternetHeader header = (InternetHeader) i.next();  
        if (header.equals(name) && header.line != null) {  
            acc.add(header.getValue());  
        }  
    }  
    int size = acc.size();  
    if (size == 0) {  
        return null;  
    }  
    String[] h = new String[size];  
    acc.toArray(h);  
    return h;  
}
```



```
String name;  
ArrayList acc = new ArrayList(headers.size());  
for (Iterator i = headers.iterator(); i.hasNext();) {  
    InternetHeader header = (InternetHeader) i.next();  
    if (header.equals(name) && header.line != null) {  
        acc.add(header.getValue());  
    }  
}
```

Improvements - 4



To be continued in 'Lightning talk'...

Q & A

Thank you for listening ...