Engineering Solutions Using Associative Collections

Zoran Horvat

CEO at Coding Helmet

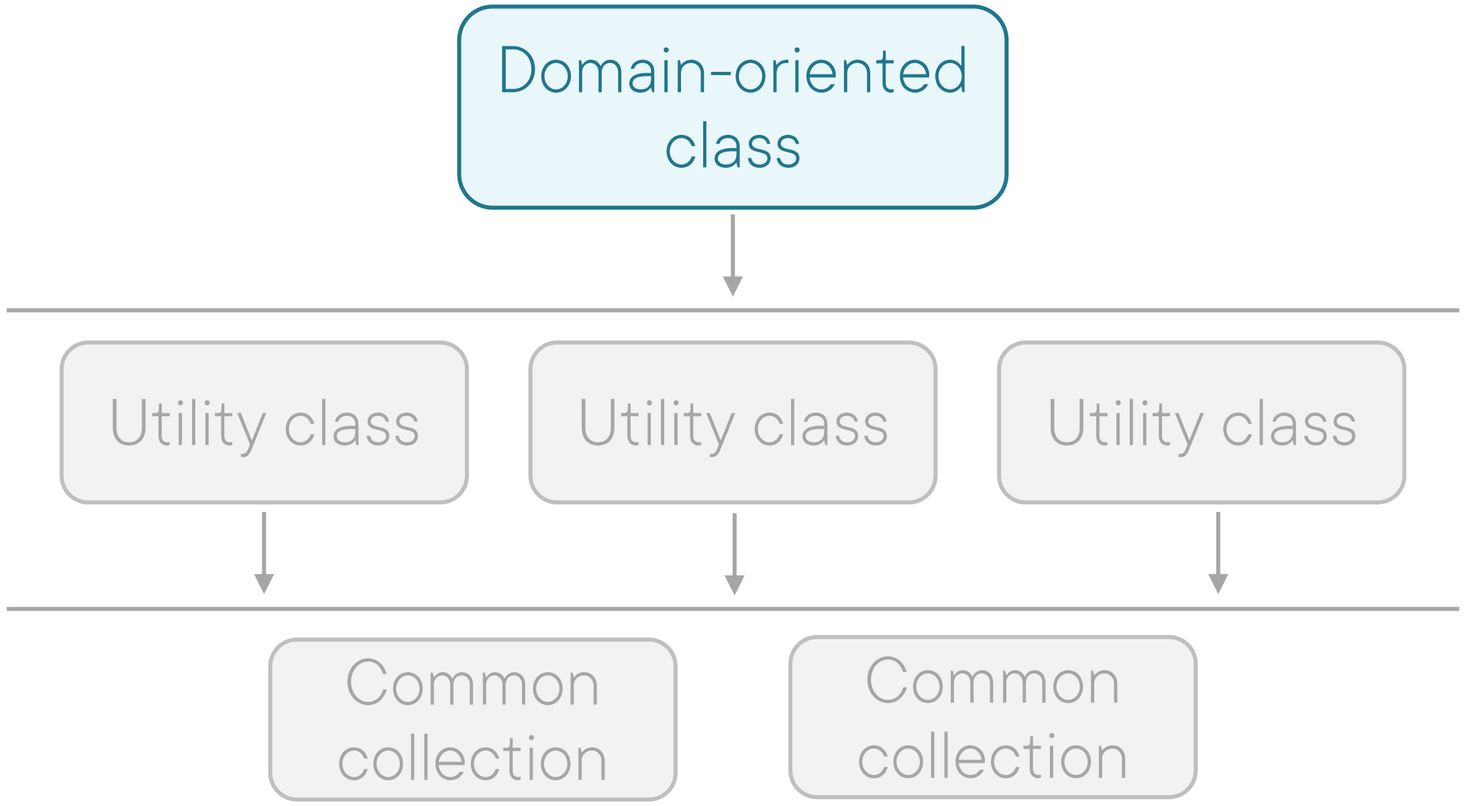
@zoranh75 https://codinghelmet.com

In This Module...

Demo 1

Transparent object cache

Demo 2



In This Module...

Demo 1

Transparent object cache

Caching solution 1

Use a `HashSet<T>`

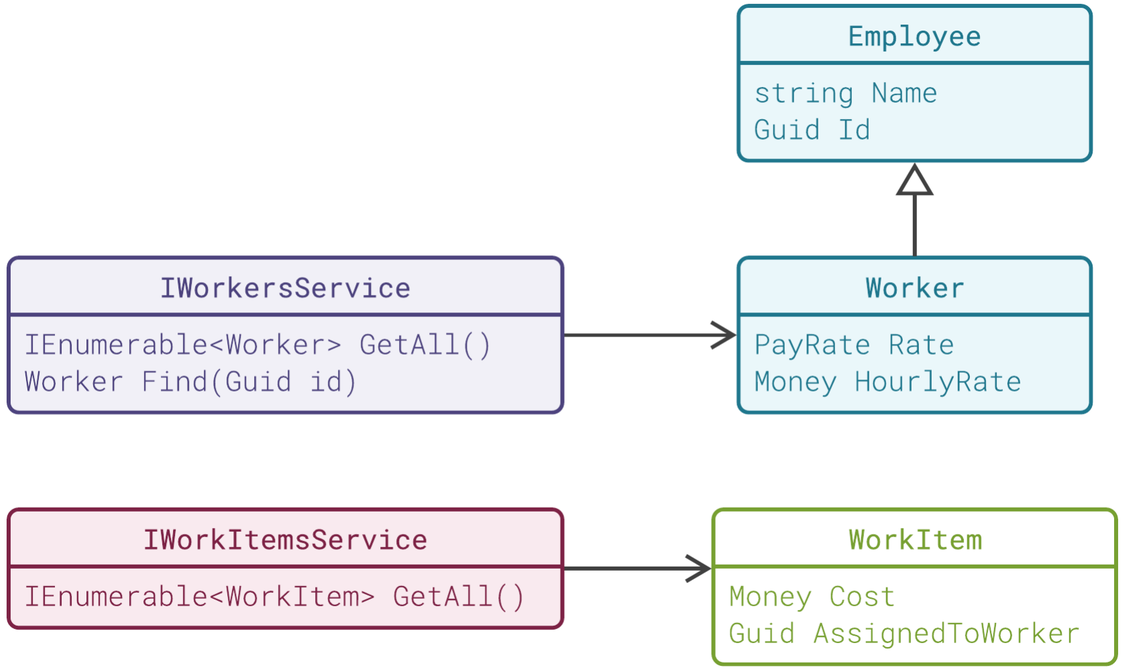
Test if an object is a duplicate

Caching solution 2

Use a `Dictionary<TKey, TValue>`

Use constructor arguments as the key

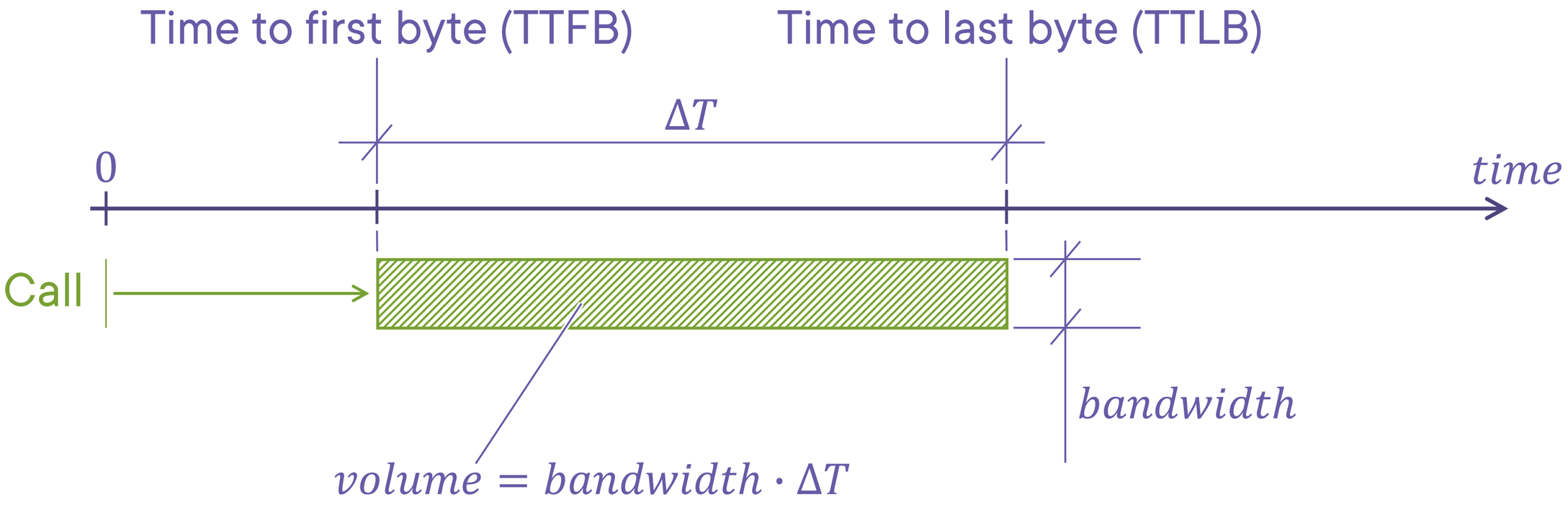
Demo



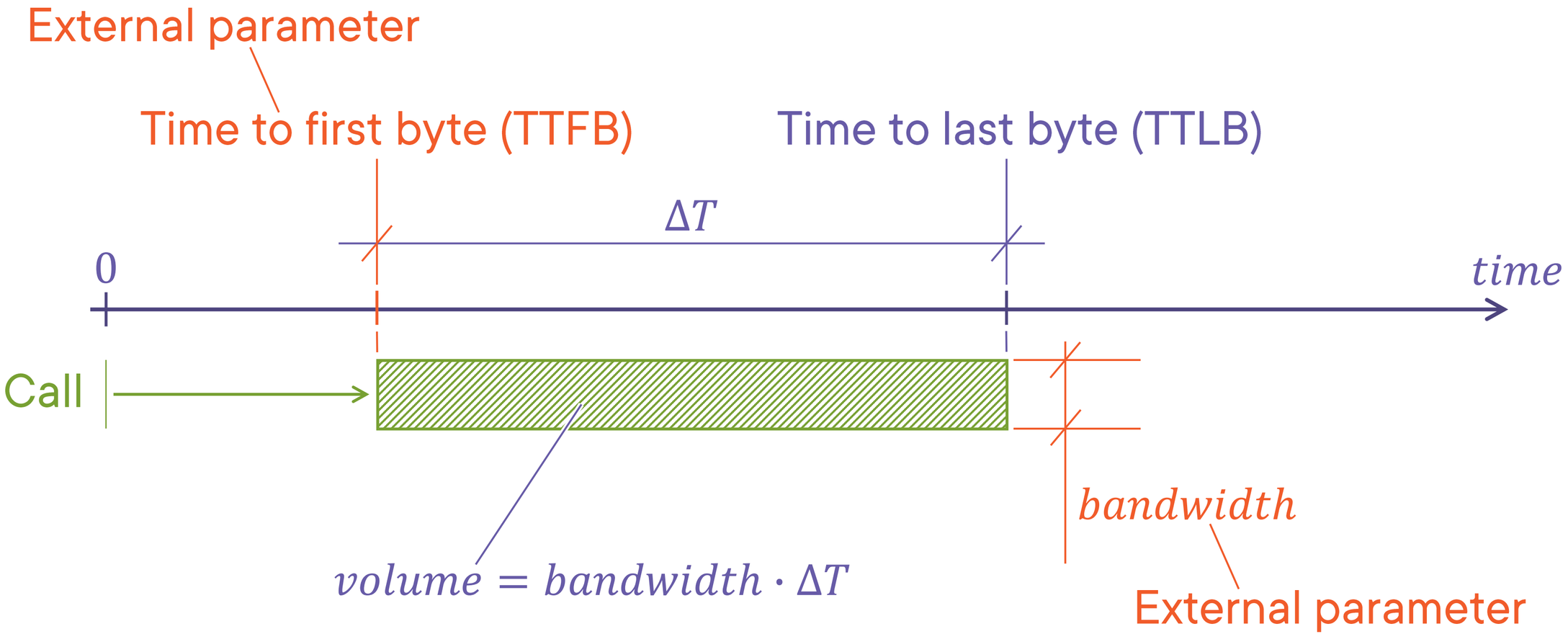
Use the `IWorkItemsService` to fetch work items

Use the `IWorkersService` to fetch the worker for each work item

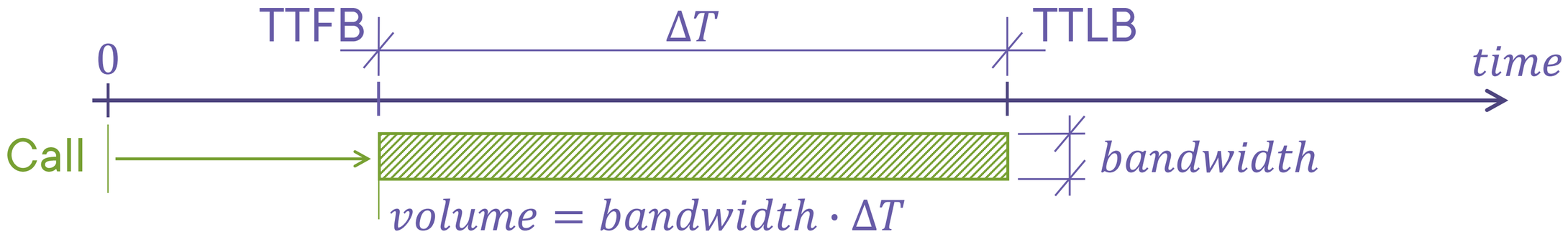
Service Communication Metrics



Service Communication Metrics



Service Communication Metrics



Chunky communication — Large volume implies $\Delta T \gg TTFB$

$$effective\ bandwidth \approx \frac{volume}{\Delta T} = bandwidth$$

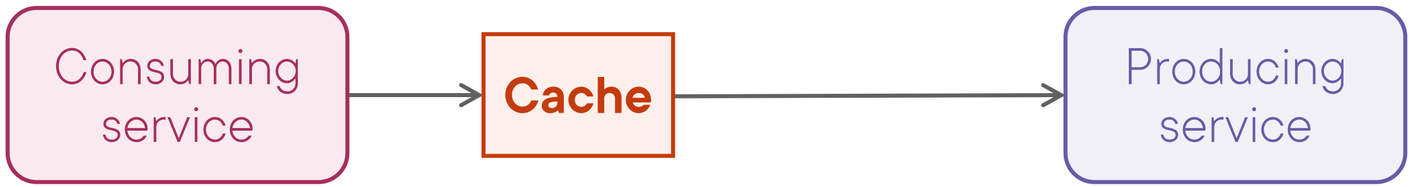
Chatty communication — Small volume implies $TTFB \gg \Delta T$

$$effective\ bandwidth \approx \frac{volume}{TTFB} \ll bandwidth$$

```
1  using System.Diagnostics;
2
3  try
4  {
5      (IWorkersService workers, IWorkItemsService workItems) = CreateServices();
6
7      int batchSize = 1000;
8
9      Stopwatch time = Stopwatch.StartNew();
10     foreach (WorkItem workItem in workItems.GetAll().Take(batchSize))
11     {
12         workers.Find(workItem.AssignedToWorker);
13     }
14     Console.WriteLine($"Processed {batchSize} item(s) in {time.Elapsed}");
15 }
16
17
18
19
20
21
22
23
24
25
```

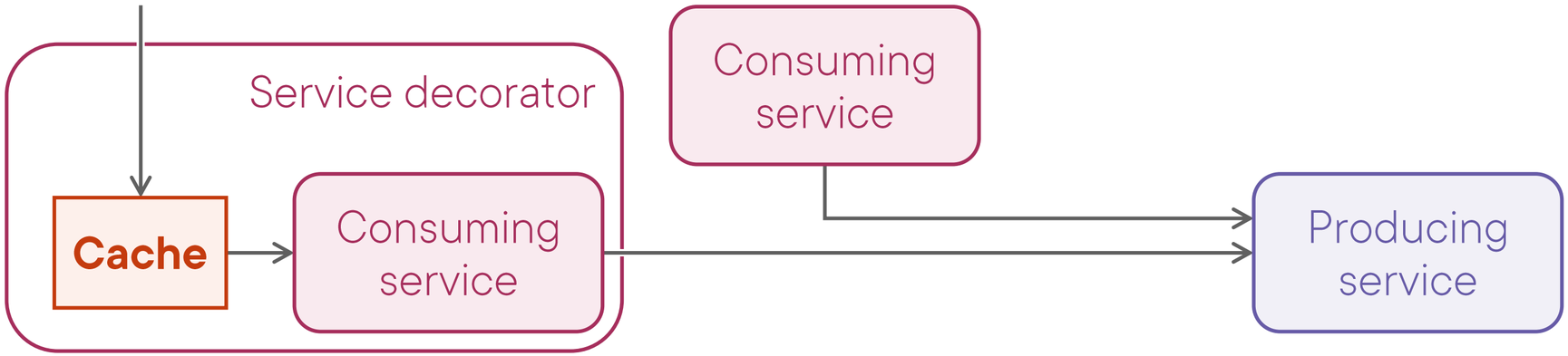
.vscode
ConsoleDemo
ConsoleDemo.csproj
CurrencyCreationBen...
Operators.cs
Program.cs
Models
Collections
FullySortedList.cs
IOrderedList.cs
LruCache.cs
TransparentCache.cs
Common
Currency.cs
CurrencyCodeEqualit...
Employee.cs
FinanceExtensions.cs
IWorkersService.cs
IWorkItemsService.cs
Models.csproj
Money.cs
MoneyBag.cs
PayRate.cs
Worker.cs
WorkItem.cs
Models.Tests
Data
Services
DelayingWorkersSer...
TestWorkItemsServic...
Models.Tests.csproj

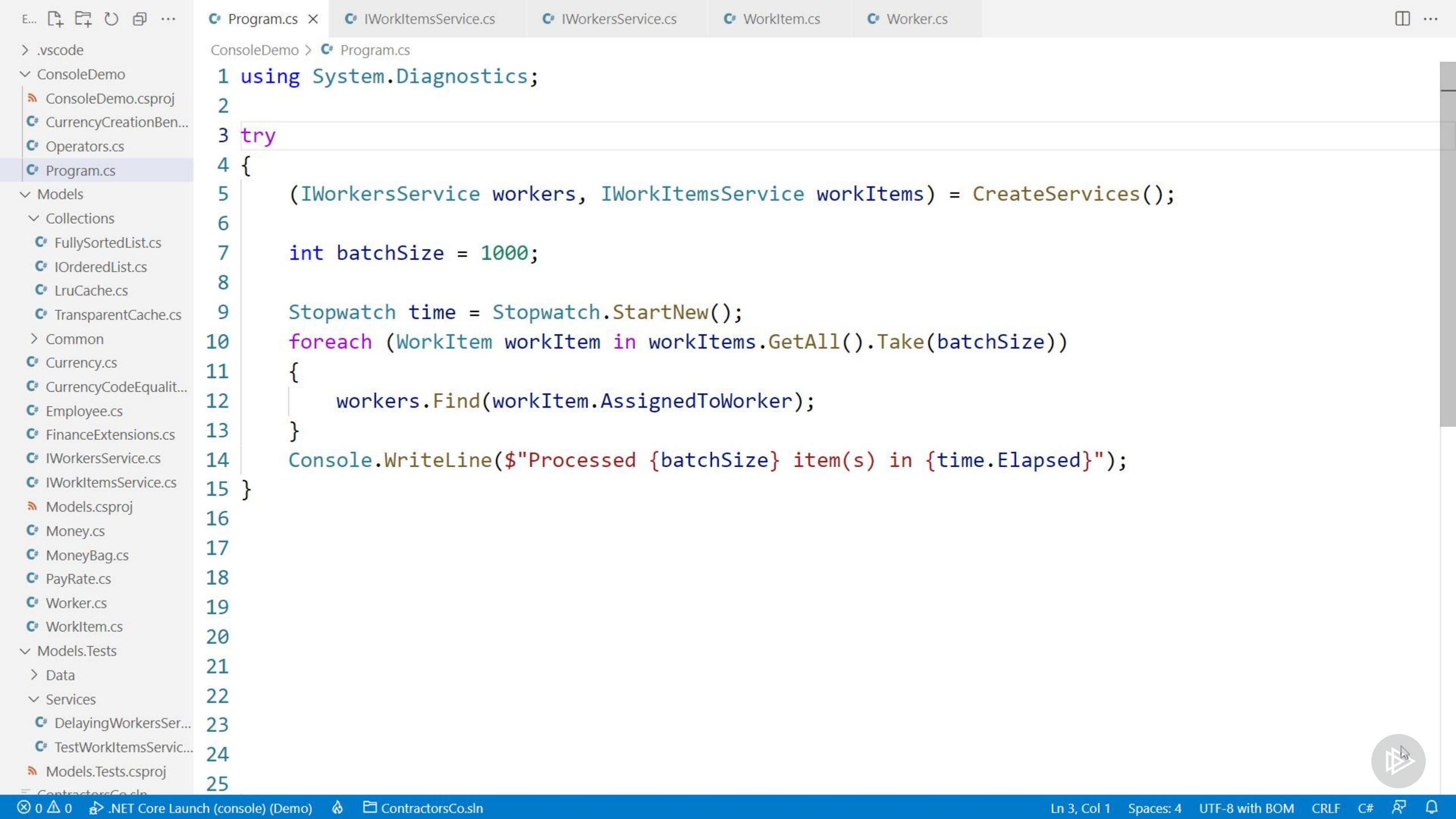
```
1  using System.Diagnostics;
2
3  try
4  {
5      (IWorkersService workers, IWorkItemsService workItems) = CreateServices();
6
7      int batchSize = 1000;
8
9      Stopwatch time = Stopwatch.StartNew();
10     foreach (WorkItem workItem in workItems.GetAll().Take(batchSize))
11     {
12         workers.Find(workItem.AssignedToWorker);
13     }
14     Console.WriteLine($"Processed {batchSize} item(s) in {time.Elapsed}");
15 }
16
17
18
19
20
21
22
23
24
25
```



- .vscode
- ConsoleDemo
  - ConsoleDemo.csproj
  - CurrencyCreationBen...
  - Operators.cs
  - Program.cs
- Models
  - Collections
    - FullySortedList.cs
    - IOrderedList.cs
    - LruCache.cs
    - TransparentCache.cs
  - Common
  - Currency.cs
  - CurrencyCodeEqualit...
  - Employee.cs
  - FinanceExtensions.cs
  - IWorkersService.cs
  - IWorkItemsService.cs
  - Models.csproj
  - Money.cs
  - MoneyBag.cs
  - PayRate.cs
  - Worker.cs
  - WorkItem.cs
- Models.Tests
  - Data
  - Services
    - DelayingWorkersSer...
    - TestWorkItemsServic...
  - Models.Tests.csproj

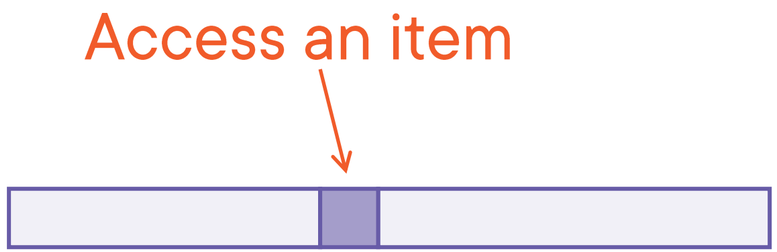

```
1  using System.Diagnostics;
2
3  try
4  {
5      (IWorkersService workers, IWorkItemsService workItems) = CreateServices();
6
7      int batchSize = 1000;
8
9      Stopwatch time = Stopwatch.StartNew();
10     foreach (WorkItem workItem in workItems.GetAll().Take(batchSize))
11     {
12         workers.Find(workItem.AssignedToWorker);
13     }
14     Console.WriteLine($"Processed {batchSize} item(s) in {time.Elapsed}");
15 }
16
17
18
19
20
21
22
23
24
25
```




Program.cs
IWorkItemsService.cs
IWorkersService.cs
WorkItem.cs
Worker.cs
ConsoleDemo > Program.cs
.vscode
ConsoleDemo
ConsoleDemo.csproj
CurrencyCreationBen...
Operators.cs
Program.cs
Models
Collections
FullySortedList.cs
IOrderedList.cs
LruCache.cs
TransparentCache.cs
Common
Currency.cs
CurrencyCodeEqualit...
Employee.cs
FinanceExtensions.cs
IWorkersService.cs
IWorkItemsService.cs
Models.csproj
Money.cs
MoneyBag.cs
PayRate.cs
Worker.cs
WorkItem.cs
Models.Tests
Data
Services
DelayingWorkersSer...
TestWorkItemsServic...
Models.Tests.csproj
using System.Diagnostics;
try
{
(IWorkersService workers, IWorkItemsService workItems) = CreateServices();
int batchSize = 1000;
Stopwatch time = Stopwatch.StartNew();
foreach (WorkItem workItem in workItems.GetAll().Take(batchSize))
{
workers.Find(workItem.AssignedToWorker);
}
Console.WriteLine($"Processed {batchSize} item(s) in {time.Elapsed}");
}
0 0
.NET Core Launch (console) (Demo)
ContractorsCo.sln
Ln 3, Col 1
Spaces: 4
UTF-8 with BOM
CRLF
C#

- .vscode
- ConsoleDemo
  - ConsoleDemo.csproj
  - CurrencyCreationBen...
  - Operators.cs
  - Program.cs
- Models
  - Collections
    - FullySortedList.cs
    - IOrderedList.cs
    - LruCache.cs
    - TransparentCache.cs
  - Common
    - Formatting
    - Pagination
    - Shuffling
    - ArgumentExtension...
    - IPage.cs
    - IPaginated.cs
    - Operators.cs
    - SinglePassSequence...
    - TransparentCaching....
  - Currency.cs
  - CurrencyCodeEqualit...
  - Employee.cs
  - FinanceExtensions.cs
  - IWorkersService.cs
  - IWorkItemsService.cs
  - Models.csproj
  - Money.cs
  - MoneyBag.cs


```
1 namespace Models.Collections;
2
3 public class LruCache<TKey, TValue> where TKey : IEquatable<TKey>
4 {
5 }
```

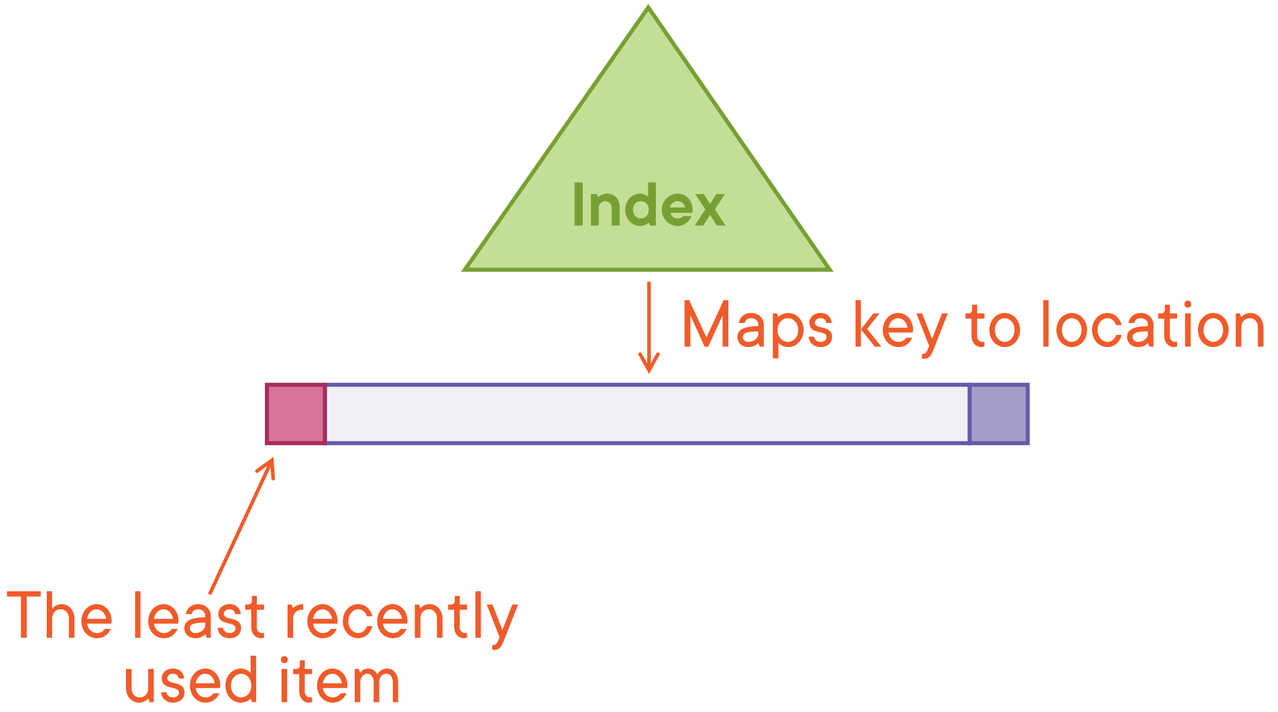


```
1 namespace Models.Collections;
2
3 public class LruCache<TKey, TValue> where TKey : IEquatable<TKey>
4 {
5 }
```

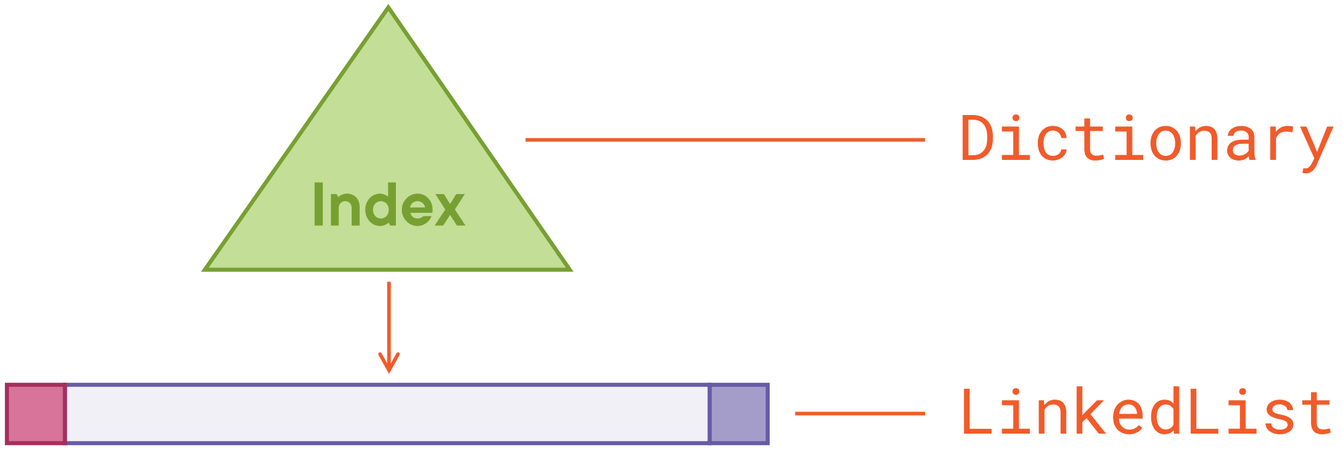
```
1 namespace Models.Collections;
2
3 public class LruCache<TKey, TValue> where TKey : IEquatable<TKey>
4 {
5 }
```

The least recently used item

```
1 namespace Models.Collections;
2
3 public class LruCache<TKey, TValue> where TKey : IEquatable<TKey>
4 {
5 }
```



```
1 namespace Models.Collections;
2
3 public class LruCache<TKey, TValue> where TKey : IEquatable<TKey>
4 {
5 }
```



Summary

Applied OOD to associative collections

Used a `HashSet` to share equatable objects

- Caused lower memory footprint
- Caused shorter execution time
- Designed a cache based on a `HashSet`

Designed a general-purpose LRU cache

- Cache keeps record of costly objects
- Used LRU cache as a private component

Summary

Caching is a matter of configuration

- Used object composition to inject a cache
- Consumer never depends on a collection

A digression on communication protocols

- Identified chatty as a problem
- Applied a local cache

Summary

Is `Dictionary` only a cache?

- Promises $O(1)$ time for operations
- Outperforms any linear collection
- Its unique ability is to *seek* by the key
- Try to view dictionary as a temporary cache

Up Next: Engineering Custom Linear Collections