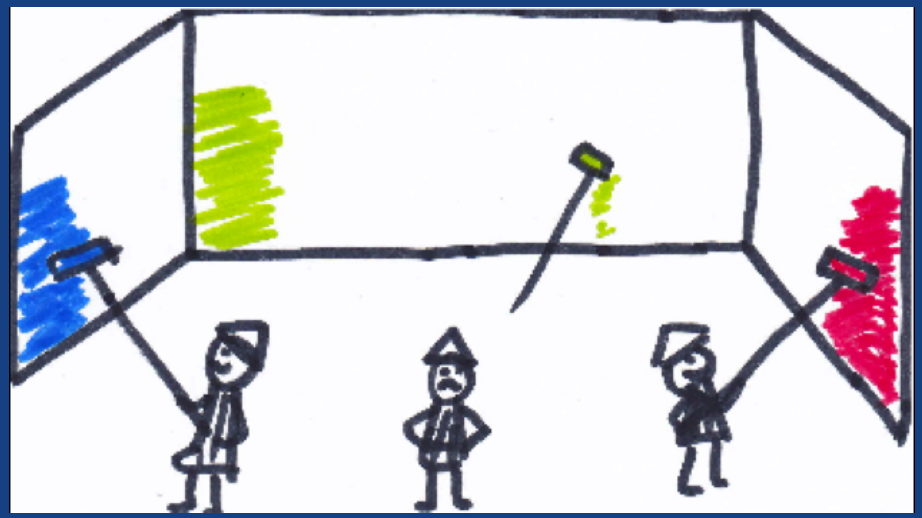
Is This Class Thread-Safe? Inferring Documentation using Graph-Based Learning

Andrew Habib, Michael Pradel

TU Darmstadt, Germany

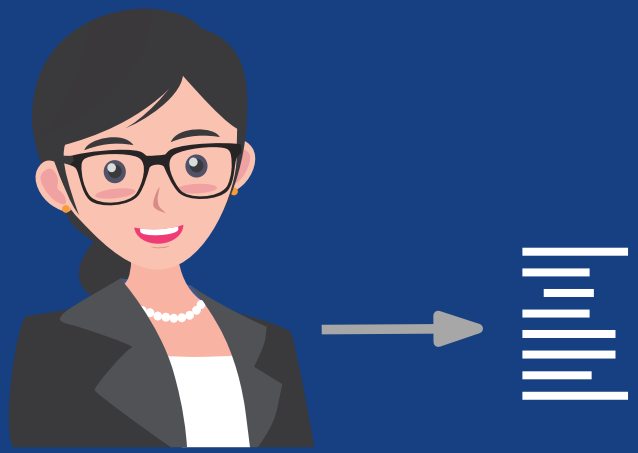
software-lab.org

Thread-Safe Classes



Thread-Safe Classes

Thread-Safe Classes



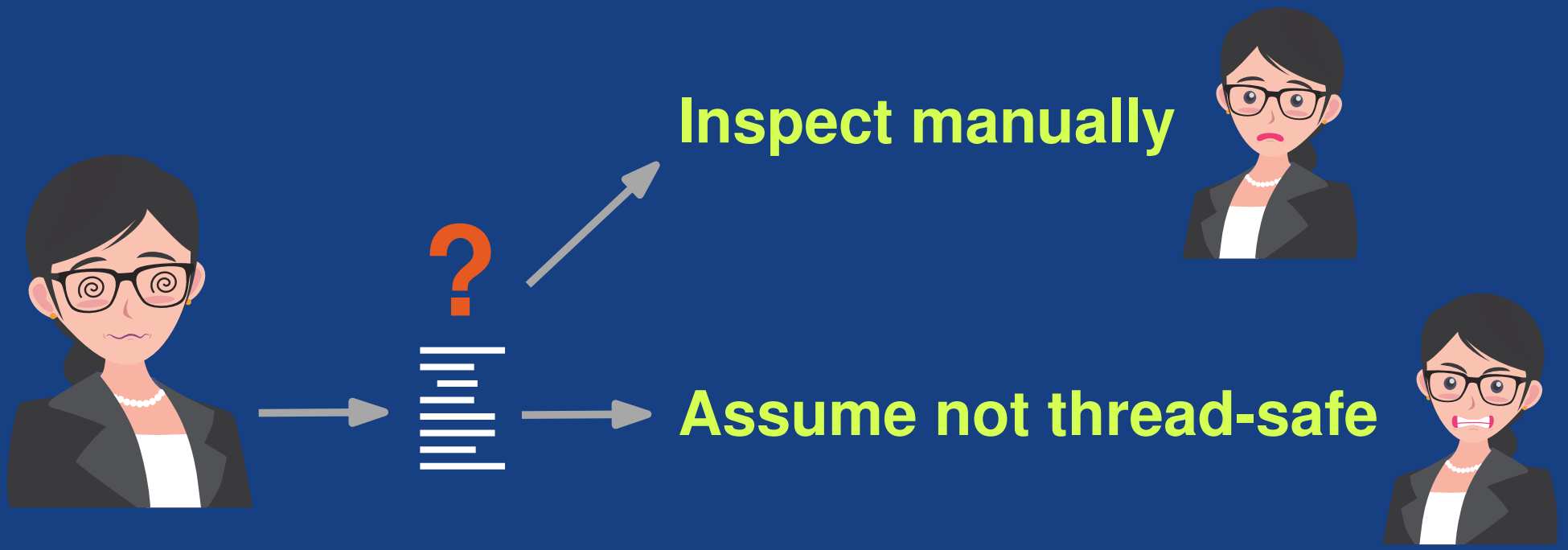
Thread-Safe Classes

Is this class thread-safe?

Thread-Safe Classes

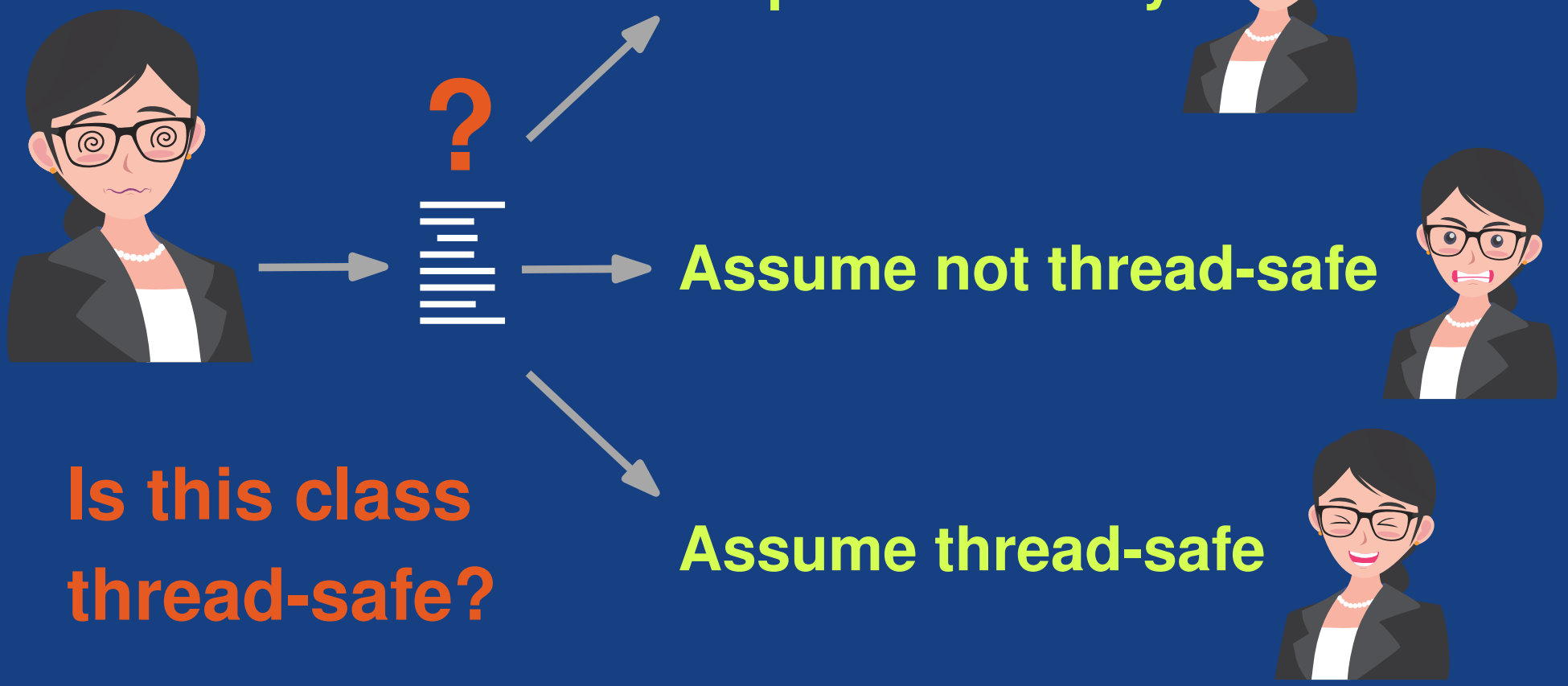
Is this class thread-safe?

Thread-Safe Classes



Is this class thread-safe?

Thread-Safe Classes



Documentation of Thread Safety

Case study: The Qualitas Corpus

- 112 Java projects
- 179,239 classes

Documentation of Thread Safety

Case study: The Qualitas Corpus

- **112 Java projects**
- **179,239 classes**
- **Search: concu, thread, sync, parallel**
  - 8,655 search hits
  - Randomly sample 120 hits
  - Manually inspect random sample

Documentation of Thread Safety

Case study: The Qualitas Corpus

- **Search: concu, thread, sync, parallel**
  - 8,655 search hits (from 179,239 classes)
  - Manually inspect random sample of 120 hits

Documented as:	Count	%
Thread-safe	11	9.2%
Not thread-safe	12	10.0%
Conditionally thread-safe	2	1.7%
No documentation	95	79.2%
Total inspected classes	**120**	**100.0%**

Documentation of Thread Safety

Case study: The Qualitas Corpus

- **Search: concu, thread, sync, parallel**
  - 8,655 search hits (from 179,239 classes)
  - Manually inspect random sample of 120 hits

Documented as:	Count	%
Thread-safe	11	9.2%
Not thread-safe	12	10.0%
Conditionally thread-safe	2	1.7%
No documentation	95	79.2%
Total inspected classes	**120**	**100.0%**

21%

Documentation of Thread Safety

Case study: The Qualitas Corpus

- **Search: concu, thread, sync, parallel**
  - 8,655 search hits (from 179,239 classes)
  - Manually inspect random sample of 120 hits

Documented as:	Count	%
Thread-safe	11	9.2%
Not thread-safe	12	10.0%
Conditionally thread-safe	2	1.7%

21%

By extrapolation:

% of documented classes = 1.004%

Is This Class Thread-Safe?

Given an object-oriented class with unknown multi-threading behaviour, infer whether it is supposed to be thread-safe or not

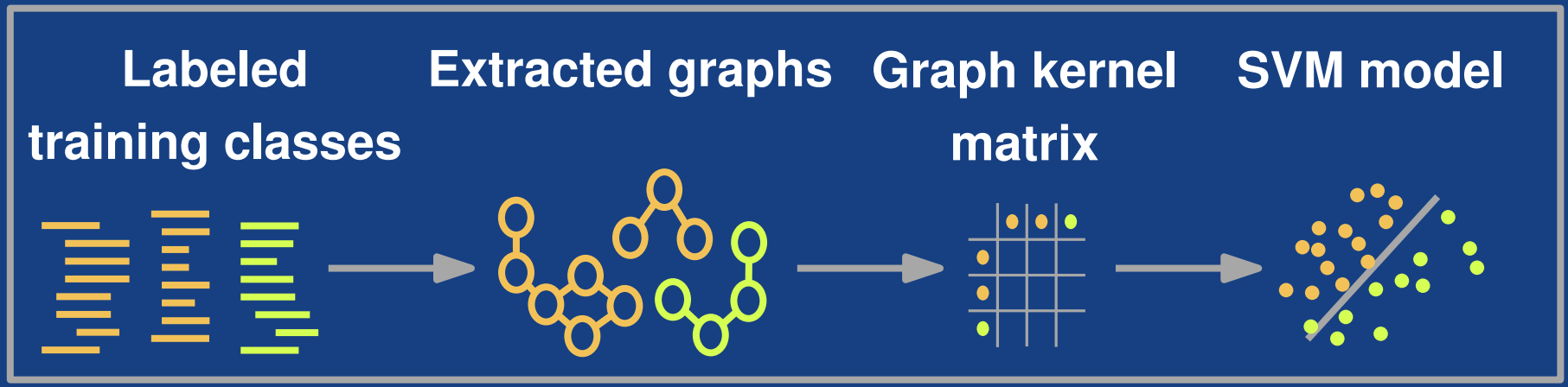
Is This Class Thread-Safe?

Given an object-oriented class with unknown multi-threading behaviour, infer whether it is supposed to be thread-safe or not

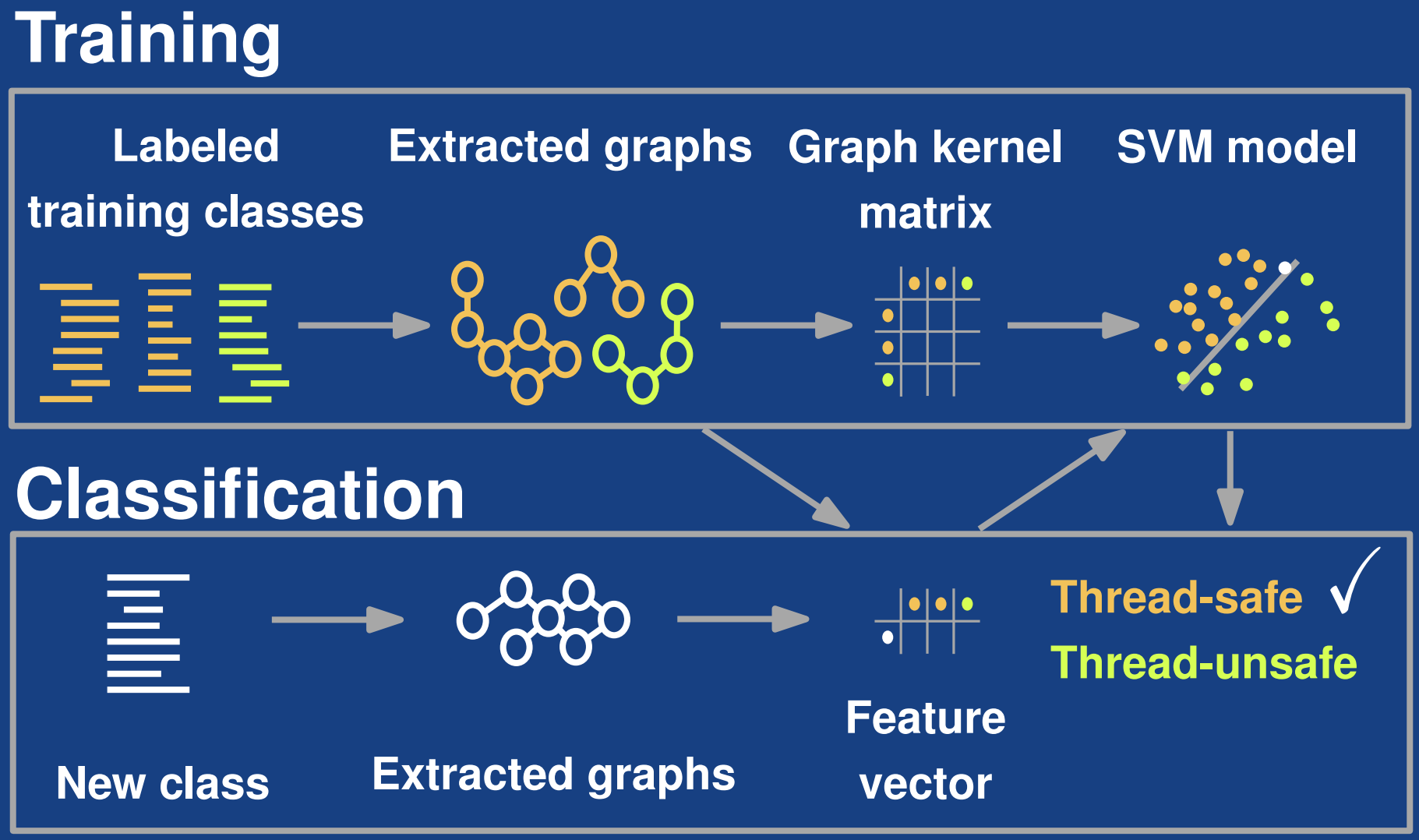
This talk: TSFinder
Machine learning approach
to infer thread-safety documentation

Overview of TSFinder

Training



Overview of TSFinder



Field-Focused Graphs

```
public class Sequence {
  private volatile int seq;
  private int MAX;
  public Sequence(int m) {
    MAX = m;
    reset();
  }
  synchronized
  public int next() {
    if(!isMax())
      return seq++;
    return -1;
  }
  boolean isMax() {
    return seq > MAX;
  }
  void reset() {
    seq = 0;
  }
}
```

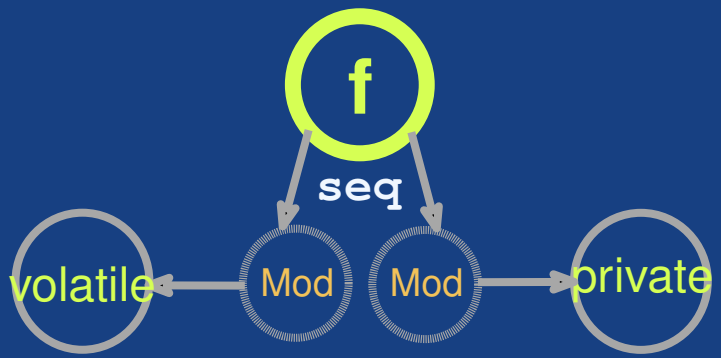
Field-Focused Graphs

```
public class Sequence {
  private volatile int seq;
```


```
}
```

Field-Focused Graphs

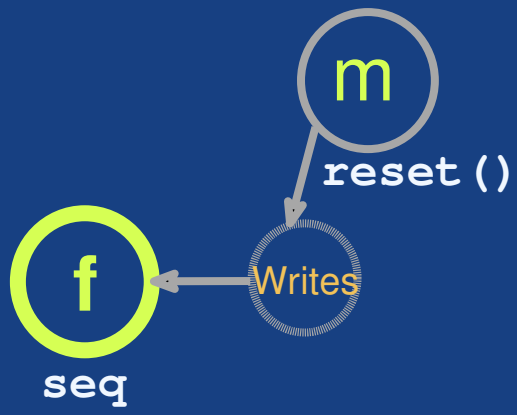
```
public class Sequence {
  private volatile int seq;
}
```



Mod: Modifier

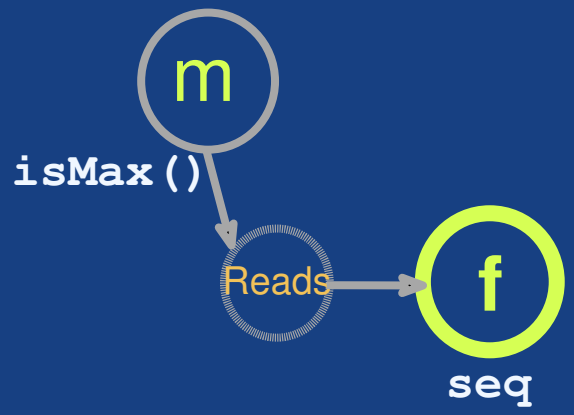
Field-Focused Graphs

```
public class Sequence {
  private volatile int seq;

  void reset() {
    seq = 0;
  }
}
```



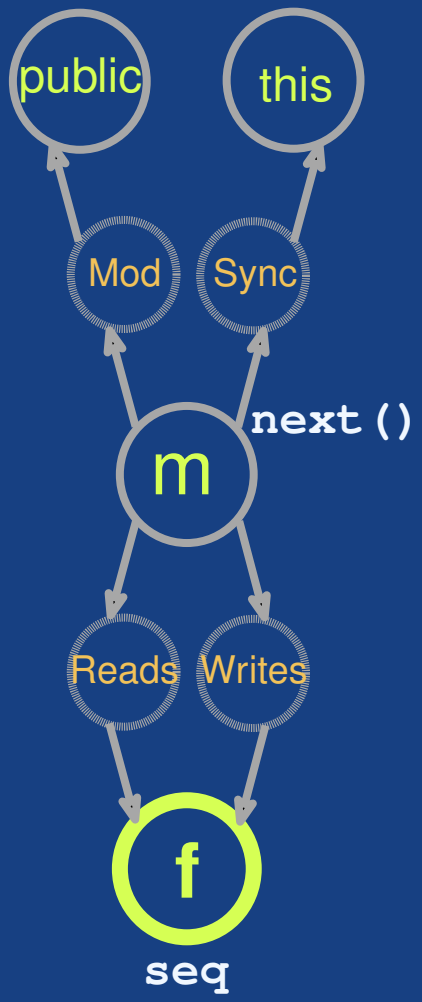
Field-Focused Graphs

```
public class Sequence {
  private volatile int seq;

  boolean isMax() {
    return seq > MAX;
  }

}
```



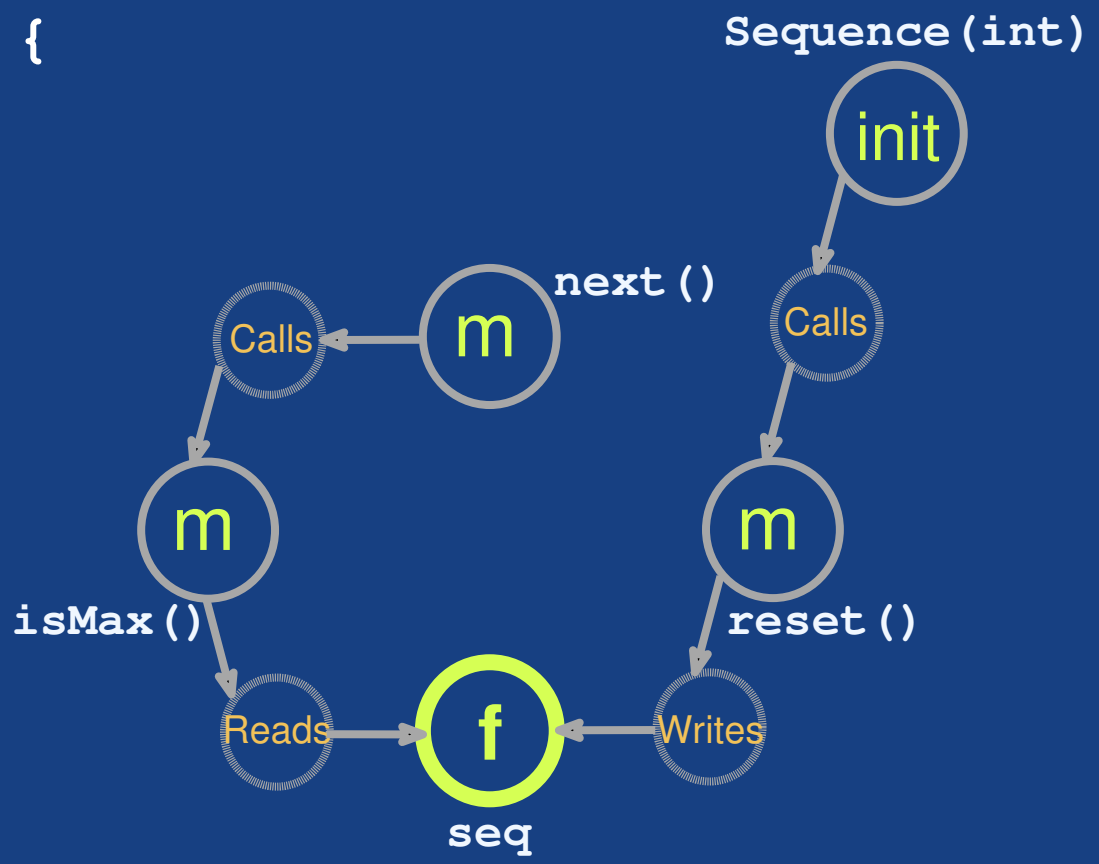
Field-Focused Graphs

```
public class Sequence {
  private volatile int seq;

  synchronized
  public int next() {
    if(!isMax())
      return seq++;
    return -1;
  }

}
```



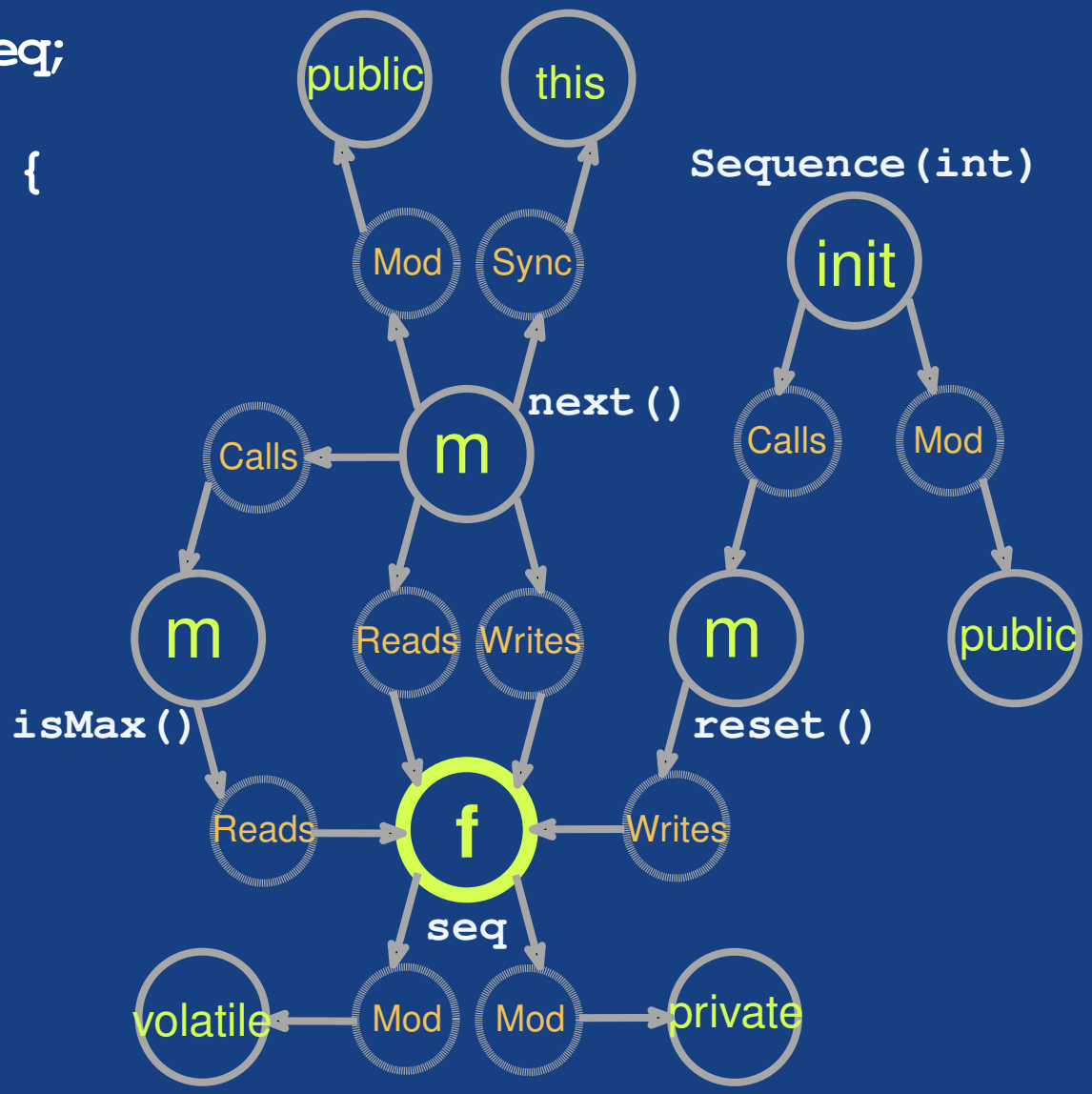
Field-Focused Graphs

```
public class Sequence {
  private volatile int seq;

  public Sequence(int m) {

    reset();
  }
  synchronized
  public int next() {
    if(!isMax())


  }

}
```



Field-Focused Graphs

```
public class Sequence {
  private volatile int seq;
  private int MAX;
  public Sequence(int m) {
    MAX = m;
    reset();
  }
  synchronized
  public int next() {
    if(!isMax())
      return seq++;
    return -1;
  }
  boolean isMax() {
    return seq > MAX;
  }
  void reset() {
    seq = 0;
  }
}
```



Mod: Modifier

Field-Focused Graphs (2)

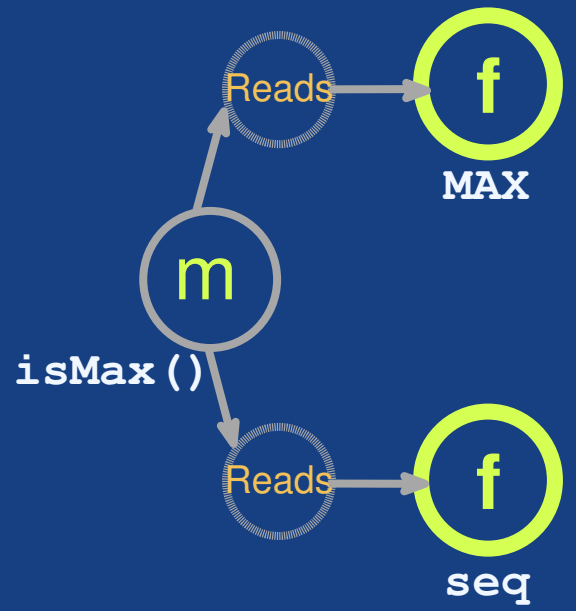
```
public class Sequence {
  private volatile int seq;
  private int MAX;
  public Sequence(int m) {
    MAX = m;
    reset();
  }
  synchronized
  public int next() {
    if(!isMax())
      return seq++;
    return -1;
  }
  boolean isMax() {
    return seq > MAX;
  }
  void reset() {
    seq = 0;
  }
}
```

Field-Focused Graphs (2)

```
public class Sequence {
  private volatile int seq;
  private int MAX;

  boolean isMax() {
    return seq > MAX;
  }

}
```

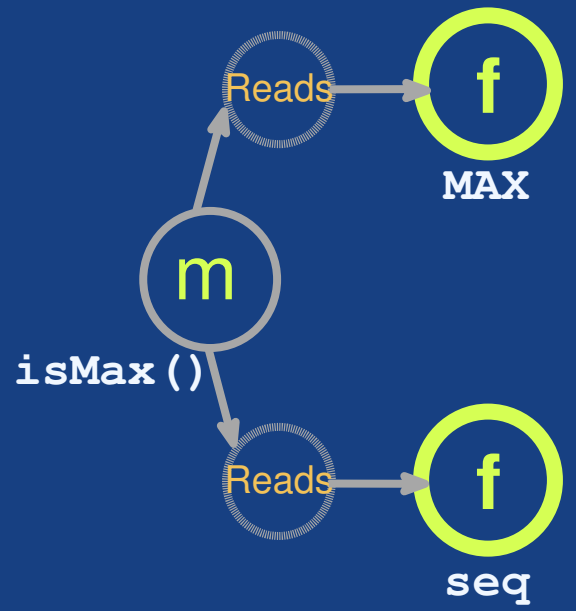
Field-Focused Graphs (2)

```
public class Sequence {
  private volatile int seq;
  private int MAX;

  boolean isMax() {
    return seq > MAX;
  }

}
```



Field-Focused Graphs (2)

```
public class Sequence {
  private volatile int seq;
  private int MAX;

  boolean isMax() {
    return seq > MAX;
  }

}
```



Build the rest of the graph as before

Class to Vector

Known classes

New class C

Class to Vector

Known classes

New class C

Graph kernel: * $K($  ,  $) = k \in [0, 1]$

Similarity score

* We use the *Weisfeiler-Lehman Graph Kernels* [Shervashidze et al., 2011]

Class to Vector

Known classes

New class C

Graph kernel: * $K($  ,  $) = k \in [0, 1]$

Similarity score

Summary of similarity of C to known classes

* We use the *Weisfeiler-Lehman Graph Kernels* [Shervashidze et al., 2011]

Class to Vector

Known classes

New class C

Graph kernel: * $K($  ,  $) = k \in [0, 1]$

Similarity score

Summary of similarity of C to known classes

Vector representation of class C

* We use the *Weisfeiler-Lehman Graph Kernels* [Shervashidze et al., 2011]

Evaluation: Setup

230 Java classes from the JDK

- Explicit thread safety documentation

		Fields			Methods		
Classes	Count	Min	Max	Avg	Min	Max	Avg
TS	115	1	64	8.7	2	163	34.7
not TS	115	0	55	4.3	1	103	23.8
All	230	0	64	6.4	1	163	29.2

Evaluation: Setup

230 Java classes from the JDK

- Explicit thread safety documentation

		LoC			
Classes	Count	Min	Max	Avg	Graphs
TS	115	13	4,264	430.2	1,989
not TS	115	7	1,931	219.7	2,871
All	230	7	4,264	323.1	4,860

Effectiveness of TSFinder

- Two-class SVM with SGD*
- 10-fold cross-validation
- 230 labeled JDK classes

* Stochastic Gradient Descent

Effectiveness of TSFinder

- Two-class SVM with SGD*
- 10-fold cross-validation
- 230 labeled JDK classes

	Thread-Safe		Not Thread-Safe	
Accuracy	**Prec.**	**Rec.**	**Prec.**	**Rec.**

* Stochastic Gradient Descent

Effectiveness of TSFinder

- Two-class SVM with SGD*
- 10-fold cross-validation
- 230 labeled JDK classes

	Thread-Safe		Not Thread-Safe	
Accuracy	**Prec.**	**Rec.**	**Prec.**	**Rec.**
94.5%	**94.9%**	**94.0%**	**94.2%**	**95.0%**

* Stochastic Gradient Descent

Effectiveness of TSFinder

- Two-class SVM with SGD*
- 10-fold cross-validation
- 230 labeled JDK classes

	Thread-Safe		Not Thread-Safe	
Accuracy	**Prec.**	**Rec.**	**Prec.**	**Rec.**
94.5%	**94.9%**	**94.0%**	**94.2%**	**95.0%**

Most predictions are correct!

* Stochastic Gradient Descent

Comparison with Baseline

Naive classifier using simple class feautres, e.g.:

- % of volatile fields
- % of synchronized methods

Comparison with Baseline

Naive classifier using simple class feautres, e.g.:

- % of volatile fields

	Accuracy	
Classifier	TSFinder	Naive
SVM (SGD* with hinge loss)	94.5%	75.0%
Random forest	94.1%	79.3%
SVM (SMO**)	92.5%	70.6%
SVM (SGD with log loss)	92.0%	74.3%
Additive logistic regression	92.8%	74.5%

* Stochastic Gradient Descent

** Sequential Minimal Optimization

Comparison with Baseline

Naive classifier using simple class feautres, e.g.:

- % of volatile fields

	Accuracy	
Classifier	TSFinder	Naive
SVM (SGD* with hinge loss)	94.5%	75.0%
Random forest	94.1%	79.3%
SVM (SMO**)	92.5%	70.6%
SVM (SGD with log loss)	92.0%	74.3%
Additive logistic regression	92.8%	74.5%

* Stochastic Gradient Descent

** Sequential Minimal Optimization

Efficiency of TSFinder

- **Training**
  - One-time effort
  - All steps: 11.7 minutes
  - Model graphs (230 classes): 0.6 MB
- **Classifying new class**
  - On average over 230 classes: 3 seconds
  - Graphs extraction dominates classification

Conclusion

- State-of-the-art of thread-safety documentation is poor
- TSFinder uses machine learning to infer documentation
- TSFinder infers thread safety documentation with accuracy of 94.5%