

The DeMark Project

Team DeFacto:

Tony Vo

Andrew Tran

Tuan Ma

Mei Zhang

Jeff Xu

Motivation

- Developer's time is Money. ¹
- Debugging takes up a lot of time.
- Developers often debug their code using print statements.
- Want to remove to maintain the quality of their code base.

[1] 15-1132 Software Developers, Applications. (2018, March 30). Retrieved April 15, 2018, from <https://www.bls.gov/oes/current/oes151132.htm>

What is DeMark?

Purpose

- Promote Clean Code
- Increase programmer productivity
- Save on precious developing time

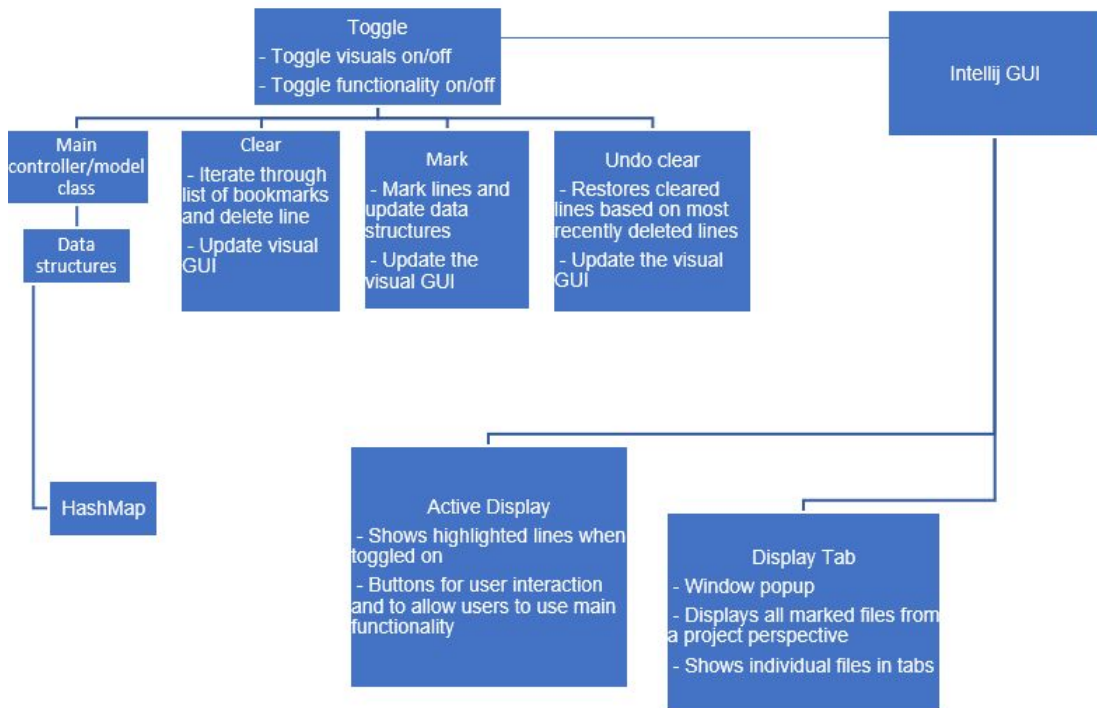
Specifics

- Keep track of temporary code
 - Print statements
 - Extra verification code
- Mark, unmark, and clear
- IDE Plugin
 - Not language specific
 - Integrate with programmer's environment

Approach

- Two major components:
 1. A model/controller.
 2. An interface that connects with IntelliJ.
- Major functions:
 - Toggle, mark, clear, unclear
- Uses a Map to keep track of marked lines.
- Uses the IntelliJ Platform SDK and extends the IntelliJ GUI as a plugin.

Architecture Diagram



DeMark Disabled

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

dslabs-tonyvo labs lab0-pingpong src dslabs pingpong PingClient

tst in Distributed Systems Labs

Project

dslabs-tonyvo [Distributed Systems Labs] ~\Classes\CSE4
├── .idea
├── .idea-out
├── doc
├── img
└── labs
 ├── lab0-pingpong
 │ ├── src
 │ │ ├── dslabs.pingpong
 │ │ │ ├── Messages.java
 │ │ │ ├── PingApplication
 │ │ │ ├── PingClient
 │ │ │ ├── PingServer
 │ │ │ └── Timeouts.java
 │ └── tst
 │ ├── README.md
 │ ├── lab1-clientserver
 │ ├── lab2-primarybackup
 │ ├── lab3-paxos
 │ ├── lab4-shardedstore
 │ ├── .gitignore
 │ ├── dviz.jar
 │ ├── framework.jar
 │ ├── framework-src.jar
 │ ├── lombok.config
 │ ├── Makefile
 │ ├── README.md
 │ └── run-tests.py
 └── External Libraries
 └── Scratches and Consoles

```
16 class PingClient extends Node implements Client {
17     private final Address serverAddress;
18
19     private Ping ping;
20     private Pong pong;
21
22     /*
23      * Construction and Initialization
24      */
25     public PingClient(Address address, Address serverAddress) {
26         super(address);
27         System.out.printf("Constructing client with client address %d\n", address);
28         this.serverAddress = serverAddress;
29     }
30
31     @Override
32     public void init() {
33         // No initialization necessary
34     }
35
36     /*
37      * Client Methods
38      */
39     @Override
40     public synchronized void sendCommand(Command command) {
41         if (!(command instanceof Ping)) {
42             System.out.println("Entering here");
43             throw new IllegalArgumentException();
44         }
45
46         Ping p = (Ping) command;
47
48         ping = p;
49         pong = null;
50
51         System.out.println(serverAddress);
52         assert (p != null);
53
54         send(new PingRequest(p), serverAddress);
55         set(new PingTimeout());
56     }
57
58     @Override
59     public synchronized boolean hasResult() {
60         if (pong == null) {
61             System.out.println("pong is null");
62         }
63         return pong != null;
64     }
65
66     @Override
67     public synchronized Result getResult() throws InterruptedException {
68         while (pong == null) {
69             wait();
70         }
71     }
72 }
```

IdeaVim: Using the Ctrl+S shortcut for Vim emulation. // You can redefine it as an IDE shortcut or configure its handler in Vim Emulation settings. (3 minutes ago)

27:9 LF UTF-8 Git: master

DeMark Enabled

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help DeMark

The screenshot shows the IntelliJ IDEA interface with the DeMark menu open. The menu options are: Clear Marked Lines, Mark Line, Toggle, Disable, Undo, and Display. The background shows the project structure on the left, including 'dsllabs-pongyo', 'labs', 'lab0-pingpong', 'src', 'dsllabs', 'pingpong', 'Messages.java', 'PingApplication', 'PingClient', 'PingServer', 'Timeouts.java', 'tst', 'README.md', 'lab1-clientserver', 'lab2-primarybackup', 'lab3-paxos', 'lab4-shardedstore', '.gitignore', 'dviz.jar', 'framework.jar', 'framework-src.jar', 'lombok.config', 'Makefile', 'README.md', 'run-tests.py', and 'External Libraries'. The main editor displays the 'PingClient.java' file with the following code:

```
class PingClient implements Client {
    private Address address;
    private Address serverAddress;
    private Pong pong;

    /*
     * Construction and Initialization
     */
    public PingClient(Address address, Address serverAddress) {
        super(address);
        System.out.printf("Constructing client with client address %d\n", address);
        this.serverAddress = serverAddress;
    }

    @Override
    public void init() {
        // No initialization necessary
    }

    /*
     * Client Methods
     */
    @Override
    public synchronized void sendCommand(Command command) {
        if (!(command instanceof Ping)) {
            System.out.println("Entering here");
            throw new IllegalArgumentException();
        }

        Ping p = (Ping) command;

        ping = p;
        pong = null;

        System.out.println(serverAddress);
        assert (p != null);

        send(new PingRequest(p), serverAddress);
        set(new PingTimeout());
    }

    @Override
    public synchronized boolean hasResult() {
        if (pong == null) {
            System.out.println("pong is null");
        }
        return pong != null;
    }

    @Override
    public synchronized Result getResult() throws InterruptedException {
        while (pong == null) {
            wait();
        }
    }
}
```

The status bar at the bottom shows: IdeaVim: Using the Ctrl+S shortcut for Vim emulation. // You can redefine it as an IDE shortcut or configure its handler in Vim Emulation settings. (3 minutes ago) 27:9 LF UTF-8 Git: master

Display Function

Build Run Tools VCS Window Help DeMark

src dslabs pingpong PingClient

View.java Messages.java PingApplication.java PingClient.java DeMark - PingClient.java



01 10 01

tst in Distributed Systems Labs



```
Classes/CSE4
25 public PingClient(Address address, Address serverAddress) {
26     super(address);
27     System.out.printf(s: "Constructing client with client address %d\n", address);
28     this.serverAddress = serverAddress;
29 }
40
41 public synchronized void sendCommand(Command command) {
42     if (!(command instanceof Ping)) {
43         System.out.println("Entering here");
44         throw new IllegalArgumentException();
45     }
46
47     pong = null;
48
49     System.out.println(serverAddress);
50     assert (p != null);
51
52     send(new PingRequest(p), serverAddress);
53
54
55
56
57
58 @Override
59 public synchronized boolean hasResult() {
60     if (pong == null) {
61         System.out.println("pong is null");
62     }
63     return pong != null;
64 }
65
```

Research and Experiments

Goal:

- Show that DeMark improves the development experience.
- Refine product based on feedback.

Challenges:

- Willingness of participation.
- Data is opinion based.

Relevant metrics:

- Time saved (at least 2 min/hour/person).
- Usefulness.
- Frequency and ease of use.

Methodology:

- Two phase testing.

Two Phase Testing

Phase 1:

- Closed group of developers.
- Alpha Version → Core functions only.

Phase 1 specific focus:

- Fix any bugs that might appear.
- Completely refine core functions.

Phase 2:

- Distributed to public to test run.
- Beta Version → Core functions refined, new features added.

Phase 2 specific focus:

- Collecting data on frequency of use.
- Most of data for research will be from phase 2.

Preliminary results

What have we learned so far:

- Coordinate meetings
- Communication

What was unusually difficult:

- Setting up work environment
- Reading documentation

Specific challenges that we are worried about:

- Finding users who are willing to test our tool
- Keep track of user reports and issues
- Keeping persistent state

What do we know today that we wish we had known a month ago?

- How to use build tools, such as Gradle

Today's World

Logging levels and loggers

Advantages:

- Multiple logging levels
- Customized output

Disadvantages:

- Language specific
- Requires external libraries and integration into the source code
- Excessive logging
- Impact performance

Manually add and delete

Advantages:

- No extra tool required
- Flexibility

Disadvantages:

- Repetitive and tedious work
- Easily make mistakes

Demark

- Not language specific
- Do not impact performance
- Mark/Unmark and clear debug lines with one click/shortcut
- Have multiple profiles
- Toggle display, separated from clear