



软件工程综合实验

静态分析工具——HiSA (Highly intelligent static analyzer)

汇报人：程珂

汇报时间：2020年7月



目录

01

项目任务

02

项目进度与里程碑

03

成员分工

04

辅助工具

05

项目成果

06

项目管理

07

工作量统计

01

项目任务

Project tasks



项目任务：跨平台C++静态分析工具



核心功能

- 核心框架
- 内存检测器 (MemoryChecker)
- 未初始化检测器(UninitObjChecker)
- 除模0检测器(DivModZeroChecker)
- 空指针解引用检测器 (DereferenceChecker)
- 数组越界检测器(ArrayBoundChecker)
- 专家模式



交互式命令行

- 调试模式
 - 打印分析进度
 - 打印ExplodedGraph
- 打印帮助信息
- 指定项目compile_commands.json文件路径
- 指定输出路径
- 指定输出格式
- 列举支持的所有检测器
- 开启专家模式



图形界面

- 囊括命令行的所有功能
- 方便快捷的自定义配置检测相关内容
- 支持语言转换
- 支持历史检测记录
- 支持从项目文件夹中自动检测 compile_commands.json 文件
- 可本地查看选项详细说明
- 进度条随时更新检测进度
- 链接官网，随时查看手册



Windows & Linux , Console & Gui



```
D:\Work\HisaConsole\hisa.exe
Welcome to HiSA REPL
Type 'quit' to stop.
HiSA>>> help
addsrc  add source file to analyze
analyze run analysis
delsrc  delete source file from source list
disable set current state
enable  set current state
help    print help info
info    get current state information
quit    quit REPL
reset   reset state
set     set current state
show    show the analysis result
HiSA>>> set help
help    print help info
debug-output-path  set debug output path
compilation-database  set compilation database path
output-format  set output formats
output-path  set output path
HiSA>>> enable help
help    print help info
gmode  enable/disable god mode
analyzer-stats  enable/disable analyzer stats
debug  enable/disable debug mode
debug-analysis-progress  enable/disable analysis progress
debug-exploded-graph  enable/disable exploded graph
debug-trim-exploded-graph  enable/disable trimmed exploded graph
output-time  enable/disable time output
```

```
hisa@hisa-virtual-machine: ~/Desktop/SEExp2020/test$ hisa
File Edit View Search Terminal Help
hisa@hisa-virtual-machine:~/Desktop/SEExp2020/test$ hisa
Welcome to HiSA REPL
Type 'quit' to stop.
HiSA>>> help
addsrc  add source file to analyze
analyze run analysis
delsrc  delete source file from source list
disable set current state
enable  set current state
help    print help info
info    get current state information
quit    quit REPL
reset   reset state
set     set current state
show    show the analysis result
HiSA>>> set help
help    print help info
debug-output-path  set debug output path
compilation-database  set compilation database path
output-format  set output formats
output-path  set output path
```

Hisa

Help Settings

where is the project:

where to put bug reort:

Check Mode	Output Info	Report Type	Checker
<input checked="" type="checkbox"/> Student	<input checked="" type="checkbox"/> Bug Location	<input checked="" type="checkbox"/> Html	<input checked="" type="checkbox"/> Divide Zero
<input type="checkbox"/> Godmode	<input checked="" type="checkbox"/> Bug Type	<input type="checkbox"/> Json	<input checked="" type="checkbox"/> Memory
	<input type="checkbox"/> Analyze Time	<input type="checkbox"/> Text	<input checked="" type="checkbox"/> Deference
	<input type="checkbox"/> Checker		<input checked="" type="checkbox"/> Uninit Struct
	<input type="checkbox"/> Description		<input checked="" type="checkbox"/> Array Bound
	<input type="checkbox"/> Statistics		

→ GO!
Start to analyze your project



官方网站



产品介绍 在线下载 构建指南 用户手册 开发者手册 联系我们

02

项目进度与里程碑

Project schedule and results



项目进度



- 本项目采用演化模型进行开发
- 项目开发过程由三个周期组成
- 每个周期都有完整的 需求分析→设计→编码→测试 过程

第一周期

- 需求分析
- 框架设计
- 检测器算法设计
- 输出Json模块设计
- 框架与检测器编码实现
- 输出模块编码实现
- 编写测试用例
- 第一周期测试

第二周期

- 需求分析
- 交互式命令行设计
- Gui图形界面设计
- 交互式命令行与Gui实现
- 输出模块功能拓展
- 核心框架与检测器的拓展与维护
- 编写交互式命令行测试用例与Gui测试指南
- 第二周期测试
- 编写用户手册

第三周期

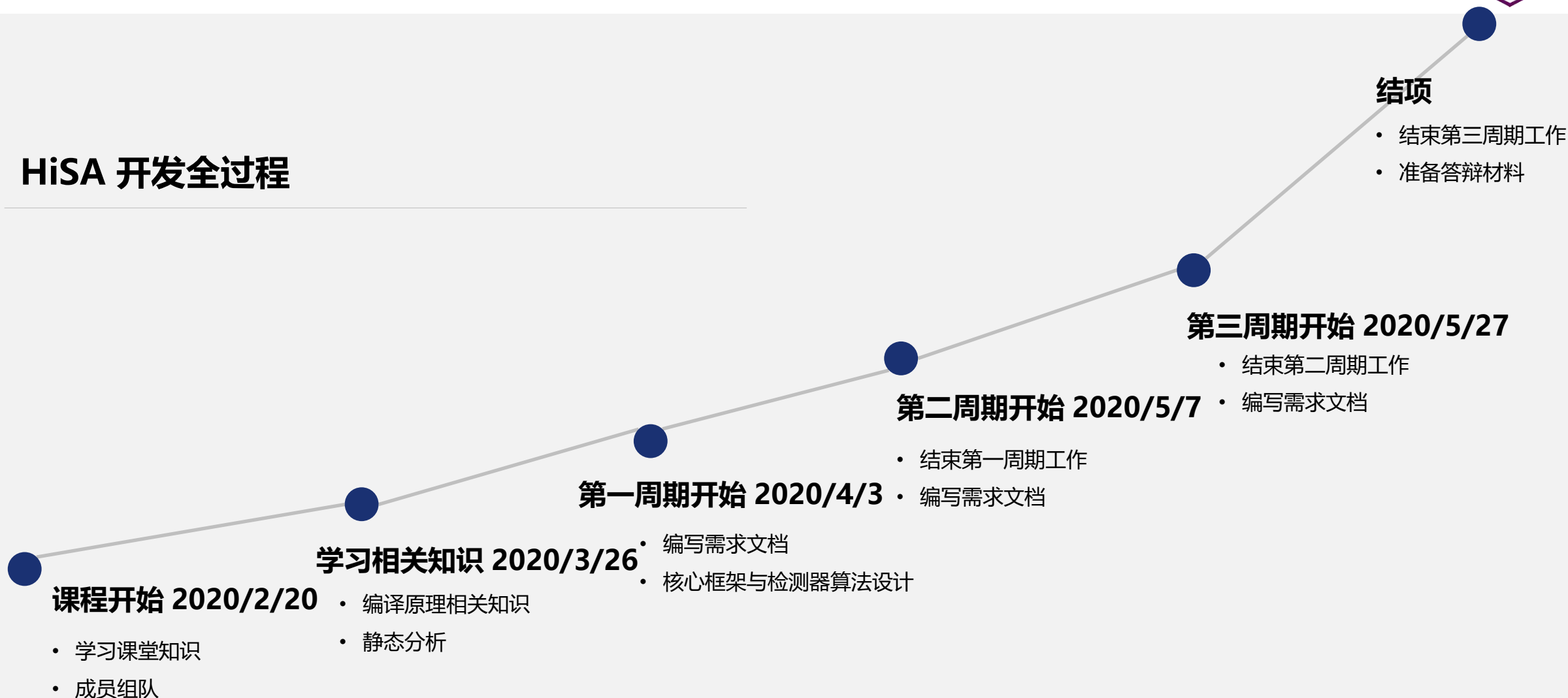
- 需求分析
- 核心框架增强设计与实现
- Gui界面功能增强设计与实现
- 官方网站搭建
- 核心框架维护与拓展
- 大规模C++项目检测
- 第三周期测试
- Bug 修复与性能优化
- 编写开发者手册



里程碑



HiSA 开发全过程





03

成员分工

The tasks assigned to each member



前期探索



助教提供
学习指南



1

- 陈昊东：理解助教框架
- 阅读 llvm, clang 源码，并尝试编写框架和demo

2

- 程珂，施超烜，周若衡，吴晓阳：通过一系列资料学习关于数据流分析，编译原理等静态分析相关知识，并尝试使用llvm和clang提供的API进行算法实现

3

- 组内成员经过编码尝试后与助教讨论后决定：弃用助教框架。自行编写框架并集成clang内置的符号执行引擎进行核心检测算法的开发



第一周期



The first iteration

陈昊东

- MemoryChecker 与项目核心框架的设计
- MemoryChecker与项目核心框架的开发（负责重复释放，释放后使用，内存泄漏缺陷的检测）
- MemoryChecker 与核心框架测试用例的编写

程珂

- 第一周期需求分析说明书的编写，DivModZeroChecker的设计
- DivModZeroChecker的开发(负责除0，模0缺陷的检测)
- DivModZeroChecker 测试用例的编写

施超烜

- DeferenceChecker的设计
- DeferenceChecker的开发(负责空指针解引用缺陷的检测)
- DeferenceChecker测试用例的编写

吴晓阳

- JsonProc 与 DiagJsonTransform 模块的设计与实现并编写相应测试用例

周若衡

- UninitObjChecker 的设计
- UninitObjChecker的开发(负责未初始化结构体缺陷的检测)
- UninitObjChecker测试用例的编写



第二周期



The second iteration

陈昊东

- REPL(交互式命令行)模块设计，实现并编写测试用例
- 核心框架的维护与修复
- 命令行版本用户手册的编写

程珂

- 第二周期需求分析说明书的编写
- Gui图形界面的设计，实现并编写测试指南
- Gui版本用户手册的编写

施超烜

- HTML输出报告的设计
- HTMLPrinter模块的设计与实现

吴晓阳

- OutPutManager的设计与实现(负责控制输出格式与类型的模块)
- JsonPrinter 与 TextPrinter 的设计与实现

周若衡

- Statistics 模块的设计与实现
- PreProcessor 模块的设计与实现



第三周期



The third iteration

陈昊东

- 核心框架的维护与性能增强
- 修复大项目测试中出现的有关bug

程珂

- 第三周期需求说明书的编写，维护并拓展 Gui 模块的功能
- 设计，实现了DirTreeDialog模块，ProcessDialog模块，History模块与HistoryDialog模块，集成进Gui模块中并编写了测试指南

施超烜

- 在linux下搭建项目运行环境，并对项目进行全方位的测试。通过分析万行，十万行，百万行级别的C++项目测试HiSA的缺陷检测能力，导出各项性能指标(时间代价，空间代价)，对遇到的bug导出bug报告并通告相关模块编写者进行修复

吴晓阳

- 官方网站的搭建
- 修复大项目测试中出现的有关bug

周若衡

- ArrayBoundChecker (数组越界)的设计，实现与测试用例的编写
- 修复大项目测试中出现的有关bug

04

辅助工具

Helpers



辅助工具

Visual Studio 2019



VisualStudio 2019 & Qt & CMake
& Doxygen



Visual Studio 2019



- llvm + clang 支持使用 visual studio 进行搭建
- Windows 下C++ 开发使用的IDE，使用贯穿HiSA开发的全过程

Qt VS Tools



- C++图形用户界面应用程序开发框架
- 用于开发HiSA的图形界面，在第二，三周期内使用

CMake



- 跨平台的安装(编译)工具，可以用简单的语句来描述所有平台的安装(编译过程)
- 构建HiSA core 和 console 模块，使用贯穿HiSA开发的全过程

Doxygen



- 开源跨平台的，以类似JavaDoc风格描述的文档系统
- 用于通过项目中的规范注释生成开发者手册



辅助工具



OpenCppCoverage Plugin

Visual Studio plugin for OpenCppCoverage to compute code coverage for C++ application.



Compilation Database Generator

build passing pypi v0.10.1 python 2.7 | 3.6 license GPL-3.0



OpenCppCoverage & Git & Compiledb & Understand



Open Cpp Coverage



- 一个用于Windows下c++的开源代码覆盖测试工具
- 用以测试HiSA各个模块的代码覆盖率，使用贯穿所有测试过程

Git



- 一个开源的分布式版本控制系统
- 用于HiSA开发过程中的版本控制，使用贯穿HiSA开发的全过程

Compiledb



- 为基于GNU make 的构建系统生成Clang的编译数据库json文件的工具。
- 辅助用户在本地为待检测项目生成compile_commands.json 文件

Understand



- 一款用以分析软件的工具，支持C++
- 用以生成代码的流程图与序列图等，用以更高效地理解他人编写的代码

05

项目成果

Project Result



benchmark



功能



性能

- 1
 - 除0模0: benchmark中的**全部缺陷**都可以成功检测出
 - 内存重复释放: benchmark中的**全部缺陷**都可以成功检测出
- 2
 - 空指针解引用: benchmark中的**全部缺陷**都可以成功检测出
 - 指针释放后再使用: benchmark中的**全部缺陷**都可以成功检测出
- 3
 - 数组越界: benchmark**仅有一个**缺陷无法检测出
 - 内存泄漏: benchmark中的**全部缺陷**都可以成功检测出

- 1
 - Libjpeg: 约4w loc
 - 时间: 477s(专家模式), 419s(**学生模式**); 内存峰值: 99M (专家模式), 92M(**学生模式**)
- 2
 - Cpython[**benchmark**]: 约45w loc
 - 时间: 58.1min(专家模式), 49.6min(**学生模式**); 内存峰值: 197M (专家模式), 185M(**学生模式**)
- 3
 - Mysql: 约160w loc
 - 时间: 2.99h(专家模式), 2.69h(**学生模式**); 内存峰值: 267M(专家模式), 244M(**学生模式**)



统计信息

libjpeg

cpython

mysql

Student

Here are the analysis results about files.

In this static analysis, a total of 80 documents were detected.

The time to analyze all files is 419.247s.

The number of files detected to be correct is 44.

The number of files detected to be in error is 36.

Following is the file statistics information.

Here are the analysis results about files.

In this static analysis, a total of 267 documents were detected.

The time to analyze all files is 2978.54s.

The number of files detected to be correct is 144.

The number of files detected to be in error is 123.

Following is the file statistics information.

Here are the analysis results about files.

In this static analysis, a total of 851 documents were detected.

The time to analyze all files is 9679.65s.

The number of files detected to be correct is 627.

The number of files detected to be in error is 224.

Following is the file statistics information.

GodMode

Here are the analysis results about files.

In this static analysis, a total of 80 documents were detected.

The time to analyze all files is 477.042s.

The number of files detected to be correct is 42.

The number of files detected to be in error is 38.

Following is the file statistics information.

Here are the analysis results about files.

In this static analysis, a total of 268 documents were detected.

The time to analyze all files is 3485.18s.

The number of files detected to be correct is 136.

The number of files detected to be in error is 132.

Following is the file statistics information.

Here are the analysis results about files.

In this static analysis, a total of 834 documents were detected.

The time to analyze all files is 10777.3s.

The number of files detected to be correct is 574.

The number of files detected to be in error is 260.

Following is the file statistics information.



06

项目管理

Project management



每周会议



每周会议



总结上周工作&下周计划



制定代码规范，文档规范等



项目管理



01.代码规范

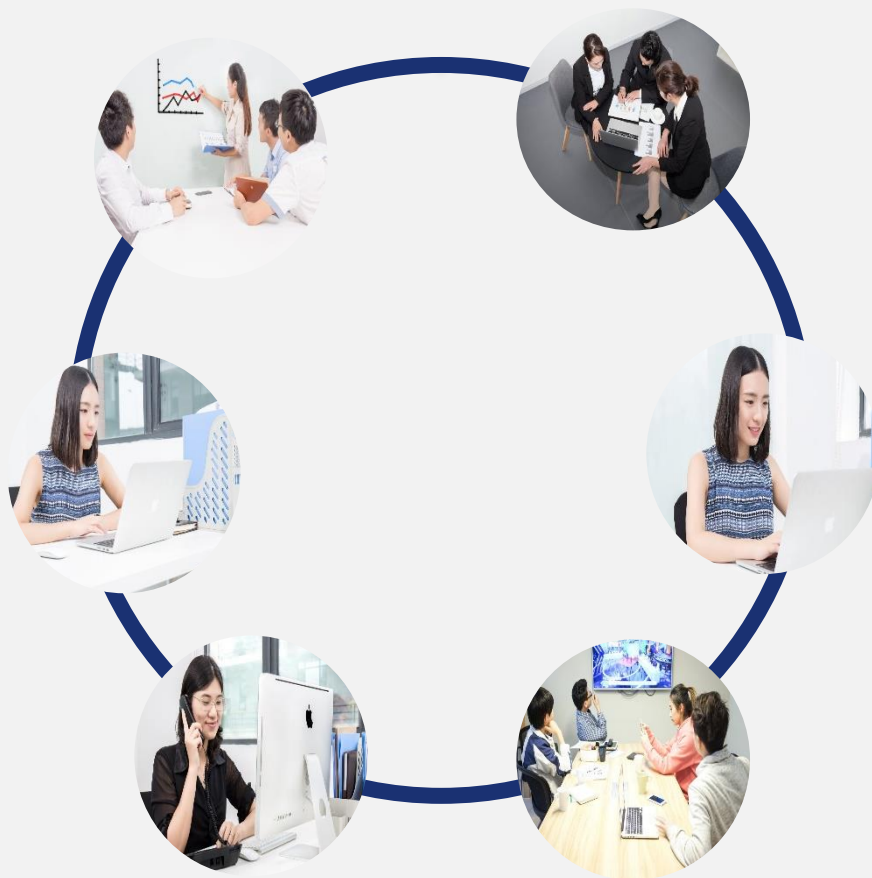
- 变量名和函数等命名规范
- 详见SEExp2020\docs\CodeStyle.md.

02.文档规范

- 会议记录规范, 需求分析文档规范等
- 详见SEExp2020\docs\meeting\会议记录模板.md.

03.协调合作

- 使用github进行协同开发
- 每周统计上周工作量并做出本周计划.
- 每位成员分工明确



04.会议记录

- 共8次会议记录
- 详见SEExp2020\docs\meeting

05.工作量统计

- 统计了每位成员的文档量, 代码量, 测试量等一系列信息
- 详见SEExp2020\docs\POW.md

06.项目研发报告

- 记录了项目研发过程中的里程碑和需求变更等信息
- 详见SEExp2020\docs\ProjectDevelopDoc.md

07

工作量统计

Proof of work



整体工作量统计



12810/1881

注释率: 14.68%

Console: 594 (C++)

Gui: 3573 (C++)

Core: 6268 (C++)

Mainpage: 2375 (HTML)



约127页

需求分析说明书: 48页

设计文档: 26页

其余文档: 共计1588行

(按每页30行计算约53页)



654/约9900行

共计检测并修复了56个bug

测试用例编写行数: 654行

测试C++文件总数: 48

平均覆盖率: 95%+





代码覆盖率

Gui



Coverage	Total lines	Items
 Uncover 5% Cover 95%	1728	Hisa_GUI.exe
 Uncover 5% Cover 95%	1728	D:\Work\SEExp2020\src\gui\debug\Hisa_GUI.exe

Console+Core

Coverage	Total lines	Items
 Uncover 4% Cover 96%	3745	hisa.exe
 Uncover 4% Cover 96%	3745	D:\github\SEExp2020\src\out\build\x64-Debug\console\hisa.exe



代码量



Console

Language	files	blank	comment	code
C++	3	8	6	566
CMake	2	6	6	29
C/C++ Header	2	5	7	28
SUM:	7	19	19	623

Core

Language	files	blank	comment	code
C++	20	460	407	4800
C/C++ Header	21	292	759	1468
CMake	2	12	12	65
SUM:	43	764	1178	6333

Gui

Language	files	blank	comment	code
C++	14	321	290	2823
C/C++ Header	10	132	260	750
Qt Project	1	8	5	25
CMake	1	5	7	9
JSON	1	0	0	1
SUM:	27	466	562	3608

Web

Language	files	blank	comment	code
HTML	4	236	0	2375
SUM:	4	236	0	2375



个人工作量统计



01.陈昊东

- 代码量: 2147行
 - 文档量: 469行(约16页)
 - 测试量: 编写测试用例246行, 覆盖率 95%+, 修复了16个bug
-

02.程珂

- 代码量: 2460行
 - 文档量: 48页+563行(约19页)
 - 测试量: 编写测试用例109行, 覆盖率 95%+, 修复了11个bug
-

03.施超烜

- 代码量: 2004
- 文档量: 352行, 约12页
- 测试量: 编写测试用例91行, 覆盖率 95%+, 修复了6个bug

04.吴晓阳

- 代码量: 2332
 - 文档量: 26页
 - 测试量: 编写测试用例25行, 覆盖率 92%+, 修复了5个bug
-

05.周若衡

- 代码量: 1882
 - 文档量: 204行(约7页)
 - 测试量: 编写测试用例183行, 覆盖率 95%, 修复了18个bug
-

06.何杰煊(助教)

- 学习指导
- 开发指导
- 进度监督



结项总结



Hisa

在软件工程综合实验课程中，Hisa全体开发人员使用演化模型进行开发，体验了软件开发从需求分析，系统设计，编码实现，软件测试和后期维护的软件开发全过程。并且在课程中学习了有很多有关软件开发过程中最前沿的知识，同时积累下了软件分析的知识基础并进行了开发实践，收获颇丰。右侧展示了开发的过程中的需求变更。





计算机科学与技术系



非常感谢您的观看

汇报人：程珂

汇报时间：2020年7月