

**2018年春季学期  
计算机学院大二软件构造课程**

**Lab 3实验报告**

|  |  |
| --- | --- |
| 姓名 | 康瑞 |
| 学号 | 1160300514 |
| 班号 | 1636101 |
| 电子邮件 | [1160300514@stu.hit.edu.cn](mailto:1160300514@stu.hit.edu.cn) |
| 手机号码 | 18745142051 |

**目录**

[1 实验目标概述 1](#_Toc508910344)

[2 实验环境配置 1](#_Toc508910345)

[3 实验过程 1](#_Toc508910346)

[3.1 待开发的四个应用场景 1](#_Toc508910347)

[3.2 面向复用的设计：Graph<L, E> 1](#_Toc508910348)

[3.3 面向复用的设计：Vertex 1](#_Toc508910349)

[3.4 面向复用的设计：Edge 1](#_Toc508910350)

[3.5 可复用API设计 2](#_Toc508910351)

[3.6 图的可视化：第三方API的复用（选做） 2](#_Toc508910352)

[3.7 设计模式应用 2](#_Toc508910353)

[3.7.1 使用State/Memento模式进行Vertex的状态管理（选做） 2](#_Toc508910354)

[3.7.2 使用factory method模式构造Vertex对象 2](#_Toc508910355)

[3.7.3 使用factory method模式构造Edge对象 2](#_Toc508910356)

[3.7.4 使用abstract factory或builder模式构造Graph对象 2](#_Toc508910357)

[3.7.5 使用Strategy模式调用centrality度量算法 2](#_Toc508910358)

[3.7.6 使用Composite模式设计超边对象（选做） 2](#_Toc508910359)

[3.7.7 使用decorator模式构造不同特征的Edge对象（选做） 2](#_Toc508910360)

[3.7.8 使用其他设计模式（选做） 2](#_Toc508910361)

[3.8 读取基于语法的文件并进行语法解析 2](#_Toc508910362)

[3.9 图操作指令的输入和处理 2](#_Toc508910363)

[3.10 应用设计与开发 3](#_Toc508910364)

[3.10.1 单词网络GraphPoet 3](#_Toc508910365)

[3.10.2 微博社交网络SocialNetwork 3](#_Toc508910366)

[3.10.3 网络拓扑图NetworkTopology 3](#_Toc508910367)

[3.10.4 电影网络MovieGraph 3](#_Toc508910368)

[4 实验进度记录 3](#_Toc508910369)

[5 实验过程中遇到的困难与解决途径 3](#_Toc508910370)

[6 实验过程中收获的经验、教训、感想 3](#_Toc508910371)

# 实验目标概述

根据实验手册简要撰写。

# 实验环境配置

简要陈述你配置本次实验所需环境的过程，必要时可以给出屏幕截图。

特别是要记录配置过程中遇到的问题和困难，以及如何解决的。

None

在这里给出你的GitHub Lab3仓库的URL地址（Lab3-学号）。

JetBrains IntelliJ Java 10

<https://github.com/ComputerScienceHIT/Lab3-160300514>

# 实验过程

请仔细对照实验手册，针对每一项任务，在下面各节中记录你的实验过程、阐述你的设计思路和问题求解思路，可辅之以示意图或关键源代码加以说明（但千万不要把你的源代码全部粘贴过来！）。

## 待开发的四个应用场景

Operating on four Graphs implements from ConcreteGraph

Separated by directed and weighted

## 基于语法的图数据输入

Parsing Patterns:

Operating by Class: ParseInputHelper.java

Separated by class: GraphFactory->(VertexFactory, EdgeFactory)

Checked by class: GraphFactory

VertexMap Held by Class: ParseInputHelper

Class ParseInputHelper: Bridge Pattern.

While you input an file name, the file name will be firstly checked by ParseInputHelper, and for each line of file input, the line will be sent to function: WarpVertex and WarpEdge(both Edge and HyperEdge), and the Function Batch Analysis in Parse Input Helper class is the General Pattern Compiler.

Another condition is use this class as a command middle contact, The class: ParseCommandHelper will send them to Parse Input Helper to make them added to the graph built by ParseInputHelper. In a word, the parse input helper is use for one graph for the add/delete a(n) vertex/edge, and used for one graph only, for the operation on graph.

Interface:

establishGraph(Filename): @return: Graph. input the file name and this function will execute the adder and constructor method.

LabeltoVertex(String label): @return :Vertex .get the store information about label-vertex

getAns():@return Graph.

judgeExist(String label):@return: Boolean, if there exists vertex owning the label return true

ParseInput(String str):private, execute each line

Cmd(Vertex/Edge)(Adder/Deleter): @return: Graph. cmd input interface, public

WrapVertex/Edge(String str):private

==================End Line of Parse input======================

EdgeFactory and VertexFactory use the same Pattern, @return: Vertex/Edge, Builder Pattern

Method: createxxxofCertaintype. Separate the typename keyandd create vertex/edge

## 面向复用的设计：Graph<L, E>

ConcreteGraph<L extends Vertex, E extends Edge> implements Graph.

CG provide the concrete method shared by each class.

Based on the specification, I added the method: clone(), which returns Graph and as the copy of an implementation. This method mostly designed for Memento Pattern.

## 面向复用的设计：Vertex

Constructed by Vertex Factory, return Vertex, Vertex as an abstract class, just implement the method asked in the specifiction.

## 面向复用的设计：Edge

Constructed by Edge factory: builder pattern, Edge is an abstrct class and implemented the method asked in the specification.

## 可复用API设计

Class GraphMetrics: Strategy

Static Method:

degreeCentrality: for a specified point the ans is the sum of in/out degree, and for the graph: the point shares the largest in and out degree, and the return value is equal to the sum of the value of the largest point minored by each point and normalized before return.

shortestPath(Graph, vertex): as an private method, return Pair

using the algorithm of dijkstra

closenessCentrality: return one count of centrality, caluated by shortestPath

betweennessCentrality: the shortest path between two point v1 and v2, and if this path passes v, the count added 1, and the counter is divided by the whole number of path.

In/out degree: the size of source/target points set.

Distance(v1, v2): using Floyd algorithm. Return a double value of the distance between v1 and v2.

Eccentricity(v): First introduced in math to describe ellipse. Graph Theory describe the value by the shortest path between v1 and v2.

Radius: of Graph g: calculated by eccentricity of every graph and divide this value by 2.

Diameter: radius\*2.

## 图的可视化：第三方API的复用（选做）

None

## 设计模式应用

Bridge, Factory, Memento, State, Builder, Strategy, decorator, Composite.

### 使用State/Memento模式进行Vertex的状态管理（选做）

Memento and State is @helper-package

Using the command to save the graph, the class: memento: keeps the timescale and the Graph saved and keep theee label of saving, when it is asked to get back the stored graph, the memento return Graph with the label at the first place and if there is no label attached, you can use the return value of save graph.

For details: String Save(Graph) with no label, and the string returned is the label, you can use this value to get Graph stored, and the method void Save(Graph, Label): attach a label to the graph to store and get graph stored, which is strongly recommended.

Method:save,restore,constructor.

### 使用factory method模式构造Vertex对象

Vertex Factory Class is @Vertex-package

Using the factory pattern, separate the input (String[], typename, label), String[] is the input (either cmd or file) separated by regex. Label, typename is gotten by the regex and can also be found in String[].

createVertexOfCertainType: judge the input legal or not, and construct the vertex.

### 使用factory method模式构造Edge对象

Using the factory pattern, separate the input (String[], typename, label), String[] is the input (either cmd or file) separated by regex. Label, typename is gotten by the regex and can also be found in String[].

createEdgeOfCertainType: judge the input legal or not, and construct the edge.

### 使用abstract factory或builder模式构造Graph对象

Using the build pattern, judge graph input legal, and the vertex and edge construct is also put in this class, keep the addVertex and addEdge method the input is Vertex and Edge constructed by Vertex Factory and Edge Factory., and judge this parameter legal or not.

### 使用Strategy模式调用centrality度量算法

Position: @help-package: GraphMetrics.class

Inherited from the Wiki and using two algorithm: Dijkstra and Floyd to calculate the value.

### 使用Composite模式设计超边对象（选做）

Hyperedge: composited as edge but judge in a different way, the input command is made of labels.

### 使用decorator模式构造不同特征的Edge对象（选做）

EdgeFactory:@Edge-package

judge the input legal or not, and construct the edge.

### 使用其他设计模式（选做）

Façade: parsecommandhelper

## 图操作指令的输入和处理（选做）

使用façade设计模式，完善ParseCommandHelper类

Parse the input string to the class: parse input helper, and judge by the following classes. The test case is recommend not contains \”.

## 应用设计与开发

利用上述设计和实现的ADT，实现手册里要求的各项功能。

### 单词网络GraphPoet

No loops directed, weighted graph and with no hyperedge. Calculated in the GraphMetrics. And constructed by the Graph Factory.

### 微博社交网络SocialNetwork

Directed, edges on one vertex is allowed, weighted graph, constrcuted by the Graphfactory。

### 网络拓扑图NetworkTopology

No loops directed, weighted graph and with no hyperedge. Calculated in the GraphMetrics. And constructed by the Graph Factory. and has some conditions and they are judged in the Graph factory.

### 电影网络MovieGraph

With loops directed, weighted graph and with hyperedge. Calculated in the GraphMetrics, hyperedge is not calc. And constructed by the Graph Factory.

## 应对四个应用面临的新变化（任选两个）None

### 单词网络GraphPoet

### 微博社交网络SocialNetwork

### 网络拓扑图NetworkTopology

### 电影网络MovieGraph

# 实验进度记录

请尽可能详细的记录你的进度情况。

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 计划任务 | 实际完成情况 |
| 5.6 | 5.6 | all | all |
|  |  |  |  |
|  |  |  |  |

# 实验过程中遇到的困难与解决途径

Most needs is not specified, and some has conflicts. The problem giver should config these problems, but actually not.

# 实验过程中收获的经验、教训、感想

本节除了总结你在实验过程中收获的经验和教训，也可就以下方面谈谈你的感受（非必须）：

1. 重新思考Lab2中的问题：面向ADT的编程和直接面向应用场景编程，你体会到二者有何差异？本实验设计的ADT在四个图应用场景下使用，你是否体会到复用的好处？
2. 重新思考Lab2中的问题：为ADT撰写复杂的specification, invariants, RI, AF，时刻注意ADT是否有rep exposure，这些工作的意义是什么？你是否愿意在以后的编程中坚持这么做？
3. 之前你将别人提供的API用于自己的程序开发中，本次实验你尝试着开发给别人使用的API，是否能够体会到其中的难处和乐趣？
4. 在编程中使用设计模式，增加了很多类，但在复用和可维护性方面带来了收益。你如何看待设计模式？
5. 你之前在使用其他软件时，应该体会过输入各种命令向系统发出指令。本次实验你开发了一系列命令行指令，使用语法和正则表达式去解析它们并映射到对后台程序的调用。你对语法驱动编程有何感受？
6. 关于本实验的工作量、难度、deadline。
7. 到目前为止你对《软件构造》课程的评价。