



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Software Construction

2019年春季学期《软件构造》 期末考试

Wang Zhongjie
rainy@hit.edu.cn

May 29, 2019

期末考试

- 闭卷，无cheat sheet
- 占总成绩60%
- 18周周二（6月25日）10:00-12:00，正心楼

- 单选题：考核对基本概念的理解，10-20题，30%分数
- 简答与设计题：
 - 给出需求、ADT的基本代码
 - 开展设计和代码：绘图/建模、设计、修改代码、写新代码 (不强调语法)、写注释 (AF/RI/Spec/Testing Strategy/Thread Safety Argument)、设计测试用例、改进/优化各项质量指标等——均在6个实验中训练过
 - 5-6题，70%分数

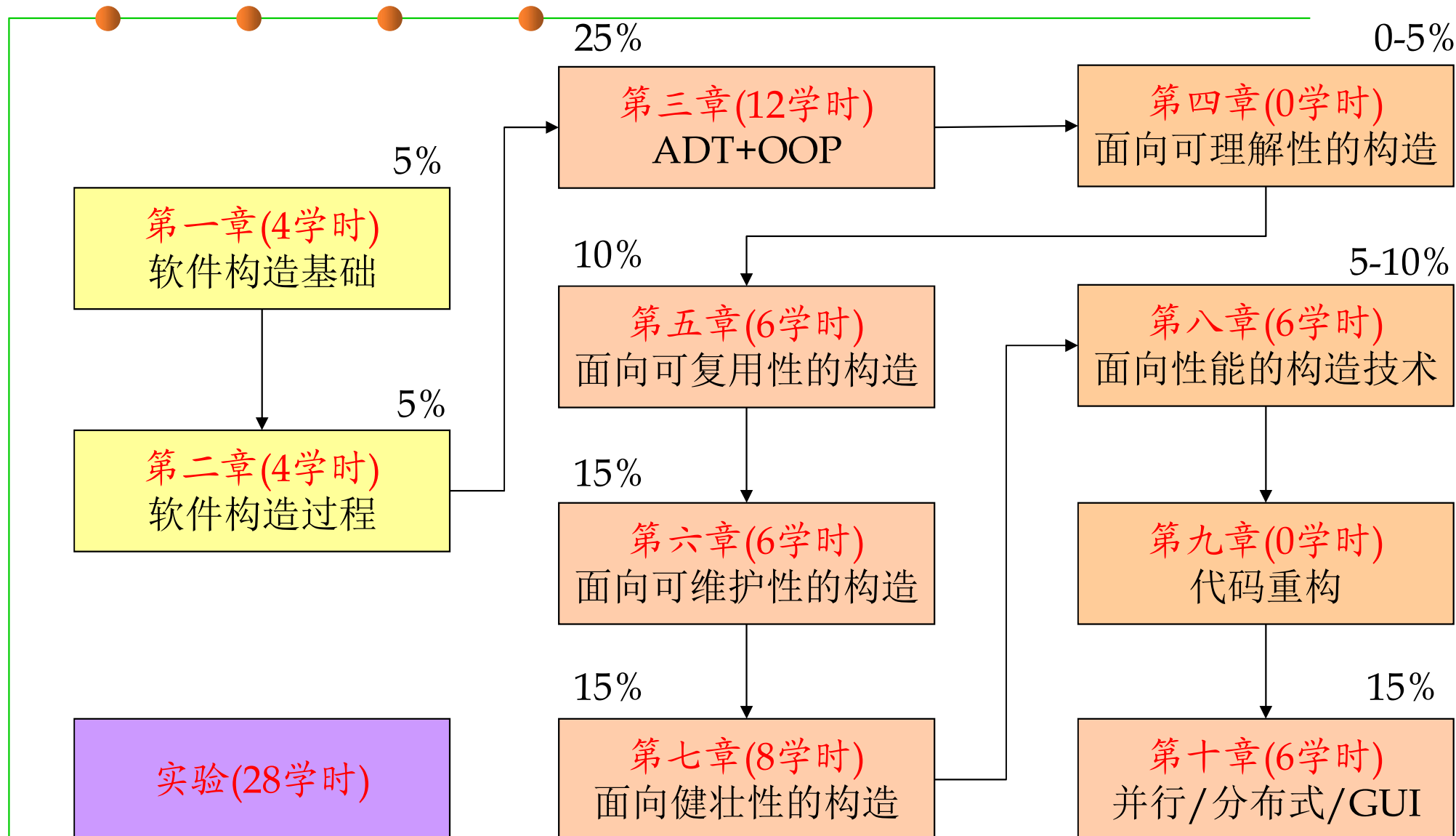
答疑

- 答疑时间：
 - 6月23日（周日） 下午2:00-5:00
 - 6月24日（周一） 下午2:00-5:00
- 地点：
 - 新技术楼510房间
- 微信上消息太多，在微信上提出问题恐无法及时回复
- 鼓励在**Piazza**上提问

参考资料

- MIT历年考试题：
 - <http://web.mit.edu/6.031/www/sp18/quizzes/archive/>
- MIT 2019年考试题：
 - <http://web.mit.edu/6.031/www/sp19/quizzes/archive/quiz1.pdf>
 - <http://web.mit.edu/6.031/www/sp19/quizzes/archive/quiz2.pdf>
- 注意：MIT的授课内容与HIT授课内容并不完全一致
 - HIT第4-9章的大部分内容，MIT并未覆盖
 - MIT某些内容，超过HIT的授课范围

考核重点



第1-2章

- 软件构造的多维度视图
- 软件构造的阶段划分、各阶段的构造活动
- 内部/外部的质量指标
- 软件配置管理SCM与版本控制系统VCS
- **Git**的结构、工作原理、基本指令
- **GitHub**

第3章

- 基本数据类型、对象数据类型
- 静态类型检查、动态类型检查
- **Mutable/Immutable**
- 值的改变、引用的改变
- 防御式拷贝
- **Snapshot diagram**
- **Specification**、前置/后置条件
- 行为等价性
- 规约的强度
- **ADT**操作的四种类型
- 表示独立性
- 表示泄露
- 不变量、表示不变量**RI**
- 表示空间、抽象空间、**AF**
- 以注释的形式撰写**AF**、**RI**
- 接口、抽象类、具体类
- 继承、**override**
- 多态、**overload**
- 泛型
- 等价性**equals()**和**==**
- **equals()**的自反、传递、对称
- **hashCode()**
- 不可变对象的引用等价性、对象等价性
- 可变对象的观察等价性、行为等价性

第4-5章

- 代码可理解性/可读性
- 编码规范
- **Programing for/with reuse**
- **LSP**
- 协变、反协变
- 数组的子类型化
- 泛型的子类型化
- 泛型中的通配符(?)
- **Delegation**
- **Comparator和Comparable**
- **CRP原则**
- 接口的组合
- 白盒框架的原理与实现
- 黑盒框架的原理与实现
- 设计模式**adapter**、**decorator**、**façade**、**strategy**、**template**、**iterator/iterable**

第6-7章

- 可维护性的常见度量指标
- 聚合度与耦合度
- SOLID
- 设计模式: factory method、abstract factory、builder、bridge、proxy、composite、observer/observable、visitor、state、memento
- 语法、正则表达式
- 健壮性和正确性
- Throwable
- Error/Runtime异常、其他异常
- Checked异常、Unchecked异常
- Checked异常的处理机制:
 - 声明、抛出、捕获、处理、清理现场、释放资源等
- 自定义异常类
- 断言的作用、应用场合
- 调试的基本过程和方法
- 黑盒测试用例的设计
 - 等价类划分、边界值分析
- 以注释的形式撰写测试策略
- JUnit测试用例写法
- 测试覆盖度

第8章

- 内存管理模型：堆、栈
- GC, root、reachable、unreachable、live、dead
- GC的四种基本算法
- Java/JVM的内存管理模型：各区域、各区域的GC方法
- JVM GC性能调优：参数配置、GC模式选择
- Java性能调优工具：jstat, jmap, jhat, Visual VM, MAT
- Memory dump
- Stack trace
- Java代码调优的设计模式：singleton, prototype/**cloneable**, flyweight, object pool
- String constant pool
- 常见的Java I/O方法

第10章

- 进程和线程
- 线程的创建和启动, **runnable**
- 内存共享模式、消息传递模式
- 时间分片、交错执行、竞争条件
- 线程的休眠、中断
- 线程安全**threadsafe**的四种策略****
 - Confinement、Immutability、ThreadSafe类型
 - Synchronization/Lock
- 死锁
- 以注释的形式撰写线程安全策略 (**ThreadSafe Argument**)



回到本学期的第一堂课





哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Software Construction

星际穿越





学编程如同Interstellar一样，
只有跳进“黑洞”，才能了解
它的秘密

but...



完成实验所需的编程技能，只有30%
来自于课堂，其他的需要你自学
so ...?





完成第 x 周的实验所需的知识，可能在
第 $x+n$ 周才能在课堂上出现

so ...?





只能用Java，但我不会/不喜欢Java
so...?





无教材...so?

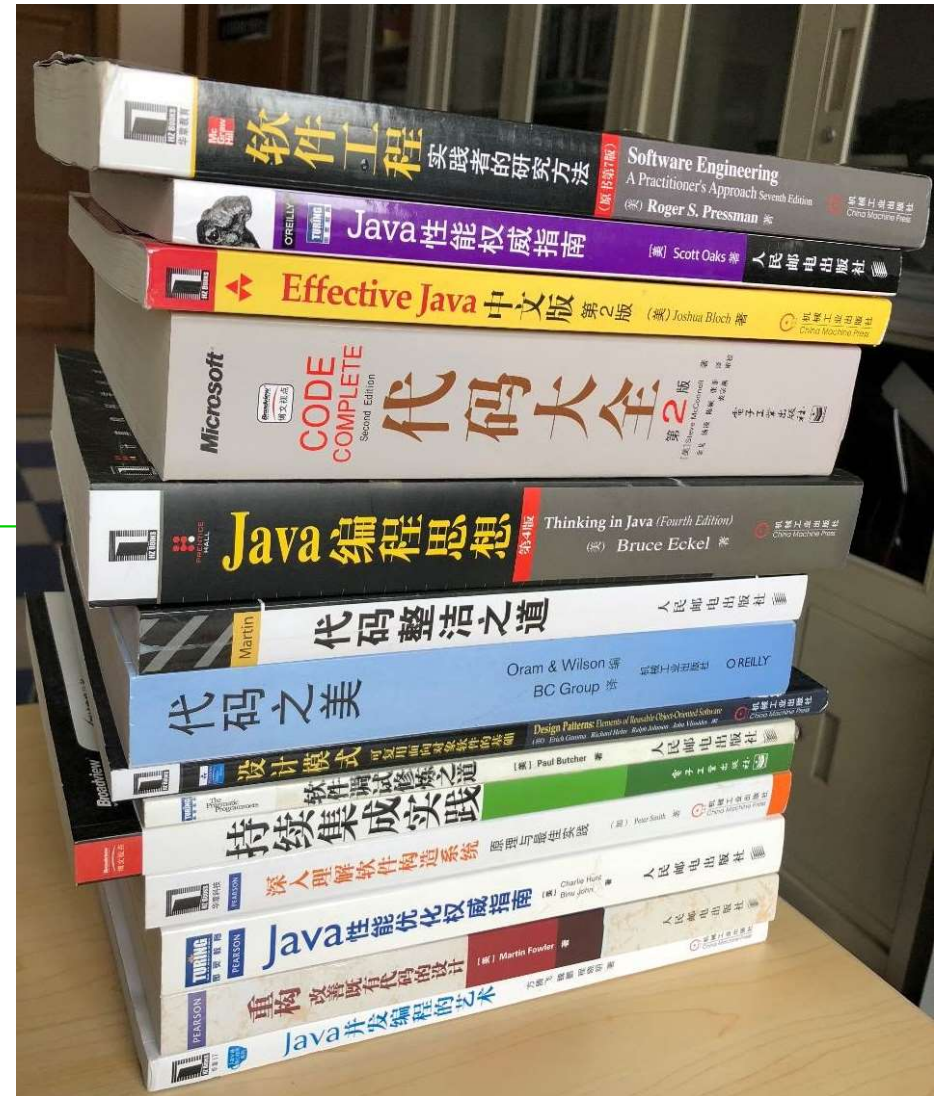




哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Software Construction

但有10+本参考书
...so?





MIT 6.031 + CMU 17-214 so?

Rank	School name	Score
#1 Tie	Carnegie Mellon University Pittsburgh, PA	5.0
#1 Tie	Massachusetts Institute of Technology Cambridge, MA	5.0
#1 Tie	Stanford University Stanford, CA	5.0
#1 Tie	University of California—Berkeley Berkeley, CA	5.0



英文的讲义 so?

“希望课件不要用英文，看不懂又没用...”



讲义上有n页ppt，老师只讲其中1/3
so?

这门课是完全开放的课程，知识点不局限在课堂上所讲，任何与ADT和OOP相关的编程知识，只要实战有用，都是学习目标。



5学分
80学时
延续16周
闭卷考试
no cheat sheet...
so?

JAVA SYNTAX CHEAT SHEET

Control Flow

Selection

If

Else

Switch

Case

Loop

While

Do

For

Exception

Throw

Try

Catch

Finally

Branch

Return

Break

Continue

Label

BASIC DEFINITIONS

Class	Describes a particular kind of object. It can contain related methods and variables.
Method	A function defined in a class. Methods implement the behavior for objects.
Object	The principal building blocks of Java. Objects consist of variables (data) and methods (functionality).

COMMENTS

HTML Comments	
<!-- comment -->	Sent to the client in the viewable page source.
JSP Comments (Not Sent to Client)	
<%-- comment --%>	Comments in JSP file.
// comment	Comment in scriptlet part of JSP file.

ELEMENTS

Declaration	
<%! declaration %>	Creates a global variable or method.
Expression	
<%= expression %>	Statements evaluated on the server before the page is outputted to the client.
Page Directive	
<%@ directive %>	Attributes that apply to the entire page.
Scriptlet	
<% code fragment of one or more lines %>	Contains a block of scripting code which is executed when the page is generated.
Taglib Directive	
<%@ taglib uri="URIToTagLibrary" prefix="tagPrefix" %>	Defines a tag library and prefix for tags used in a JSP page.

SEPARATORS

()	Used to surround parameters
{ }	Defines a block of code for a class or method or to contain the values of automatically initialized arrays
[]	Declares arrays or references array values
:	Denotes the end of a statement
.	Separates variables
.	Separates package names from subpackages/ classes or a variable/method from a reference variable

PRIMITIVE DATA TYPES

Type	Description	Bits
(Integers)		
byte	Byte-length integer	8
short	Short integer	16
int	Integer	32
long	Long integer	64
(Real Numbers)		
float	Single-precision floating point	32
double	Double-precision floating point	64
(Other)		
char	A single character	16
boolean	A boolean value (true or false)	1

OPERATORS

+	addition of numbers, concatenation of Strings
+=	add and assign numbers, concatenate and assign Strings
-	subtraction
-=	subtract and assign
*	multiplication
*=	multiply and assign
/	division
/=	divide and assign
%	take remainder
%=	take remainder and assign
++	increment by one
--	decrement by one
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to
!	boolean NOT
!=	not equal to
&&	boolean AND
	boolean OR
==	boolean equals
=	assignment

Key Words

abstract
assert
boolean
break
byte
case
catch
char
class
const
continue
default
do
double
else
enum
extends
false
final
finally
float
for
goto
if
implements
import
instanceof
int
interface
long
native
new
null
package
private
protected
public
return
short
static
strictfp
super
switch
synchronized
this
throw
throws
transient
true
try
void
volatile
while



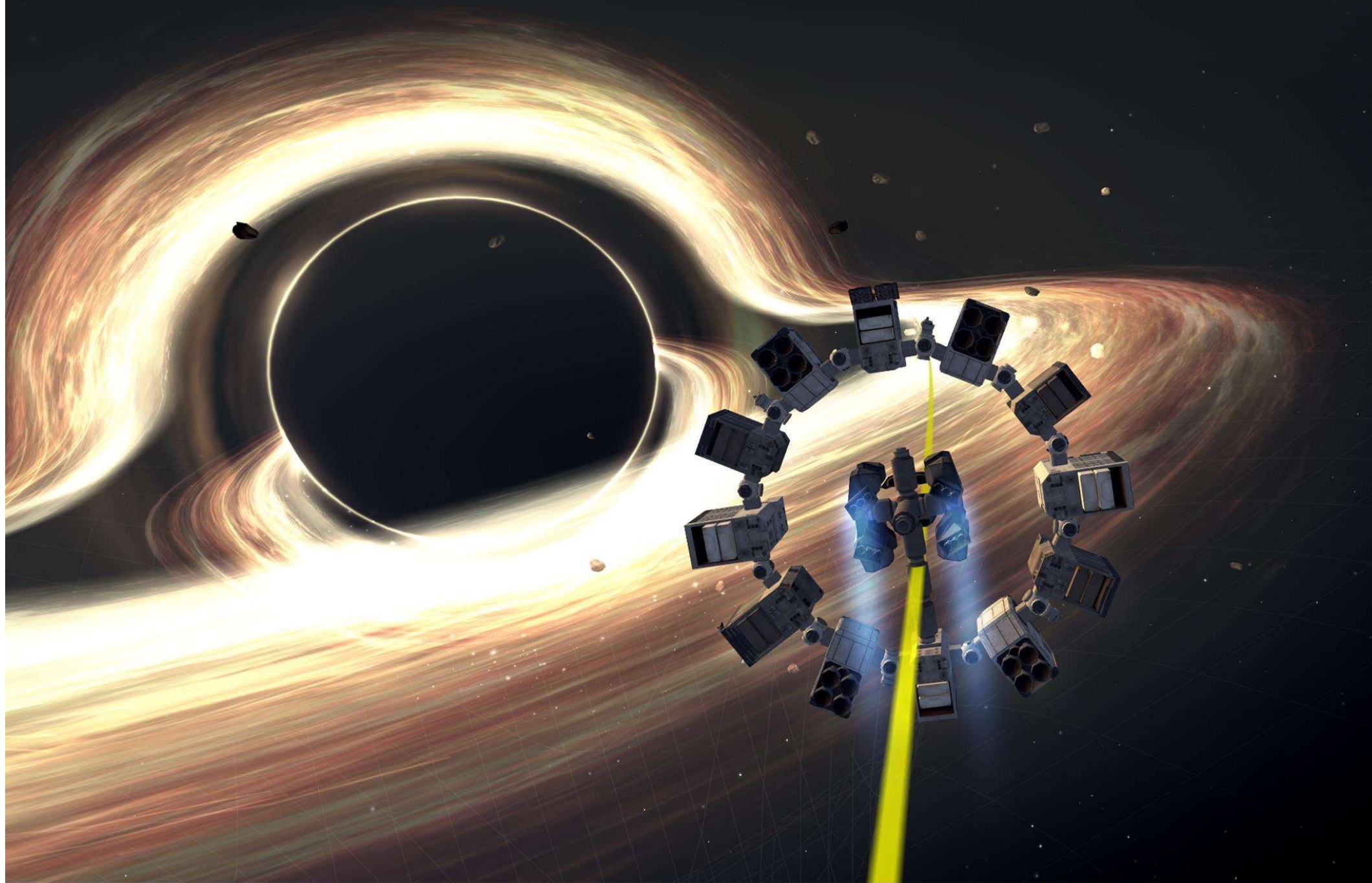
CALCULATING TRAJECTORY

MISSION TIME

EARTH TIME

00 00:08:56

00 14:08:42





三个月以来，这些是否已经变成了你的习惯？

Tips:

写更多的代码

把使用工具变成习惯

独立完成实验

遇到问题，尽可能独立解决

学会用StackOverflow解决问题

多与教师/TA交流

多总结多反思/写博客

其他部分的成绩

- 6个实验：35%，等待TA反馈
- 平时表现：5%
 - 雨课堂：自动统计计分
 - 个人博客：截止至6月24日晚

欢迎后续的互动

- 王忠杰 副院长、教授，博士生导师
 - 计算机科学与技术学院
 - 企业与服务智能计算研究中心 (ICES)
 - 电子邮件: rainy@hit.edu.cn
rainy.wang@gmail.com
 - 联系电话: 18604507162
 - 地点: 新技术楼510房间
 - 研究方向:
 - “云+端”融合的服务计算
 - 软件体系结构
 - 实证软件工程



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY





预祝各位期末考试取得好成绩！

May 29, 2019