



# 2018年春季学期《软件构造》课程 期末考试说明

Wang Zhongjie  
[rainy@hit.edu.cn](mailto:rainy@hit.edu.cn)

May 30, 2018

# 期末考试

## ■ 期末考试:

- 闭卷，无cheat sheet
- 占总成绩60%
- 17周周日（6月24日）10:00-12:00，正心楼23、24

更新

试卷满分100分，按60%计入总成绩

## ■ 考试形式:

单选题16个，共32分  
简答题1个，10分  
综合设计题4个，共58分

- 选择题：考核对基本概念的理解，10-20题，30%分数
- 简答题：给出简短代码，根据代码完成题目，1-2题，10-20%分数
- 综合设计题：给出需求、ADT的基本代码，开展设计和代码，包括绘图/建模、设计、修改代码、写新代码、写注释、设计测试用例、优化各项质量指标等，4-5题，50-60%分数

# 答疑

- 答疑时间：

- 6月22日（周五） 下午1:30-5:00
- 6月23日（周六） 下午1:30-5:00

- 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

# 参考资料

- MIT历年考试题：

- <http://web.mit.edu/6.031/www/sp18/quizzes/archive/>

- MIT 2018年考试题：

- <http://web.mit.edu/6.031/www/sp18/quizzes/archive/quiz1.pdf>

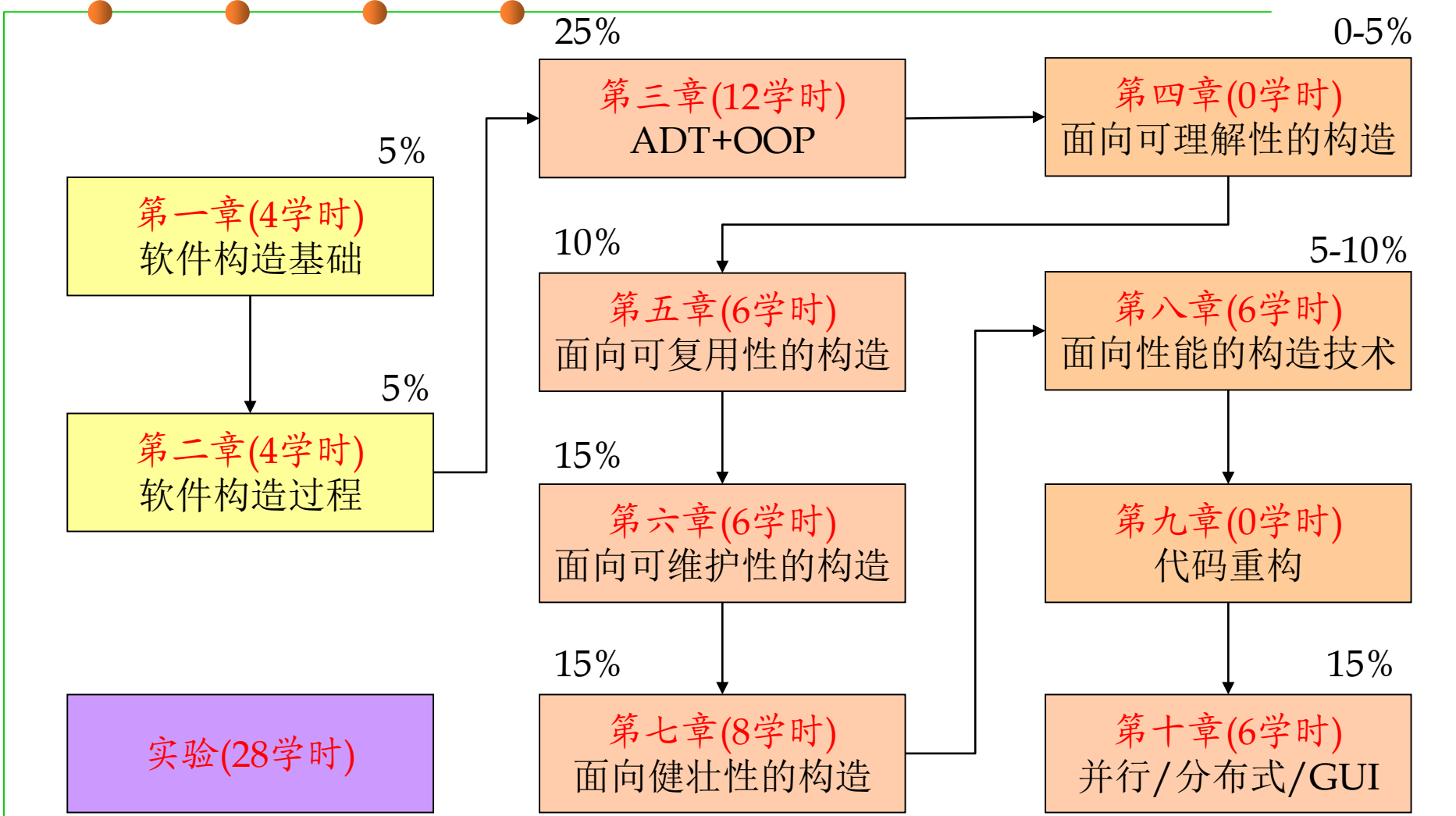
- <http://web.mit.edu/6.031/www/sp18/quizzes/archive/quiz2.pdf>

- 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()
- 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

- 7.1 ■ 健壮性和正确性

- Throwable

- 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

- 7.3 ■ 断言的作用、应用场合

- 7.4 ■ 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

- 7.5 ■ 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

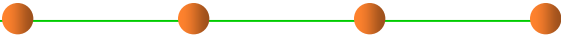
- 内存管理模型：堆、栈
- GC, root、reachable、unreachable、live、dead
- GC的四种基本算法
- Java/JVM的内存管理模型：各区域、各区域的GC方法
- JVM GC性能调优：参数配置、GC模式选择
- Java性能调优工具：jstat, jmap, jhat, Visual VM, MAT
- Memory dump
- Stack trace
- Java代码调优的设计模式：singleton, prototype/cloneable, flyweight, object pool
- String constant pool
- 常见的Java 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
- 多线程之间基于消息传递的实现机制、如何保证`threadsafe`



# 回到本学期的第一堂课





# 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 ...?

现在遇到新技术，你是否已经毫无恐惧感了？  
查技术手册、查网络，希望你以后的学习习惯





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 ...?



“原理”和“技术”，搞懂前者，后者学起来就很容易了





只能用Java，但我不会/不喜欢Java  
so...?

触类旁通，其他语言也是同样的学习路径



# 无教材...so?

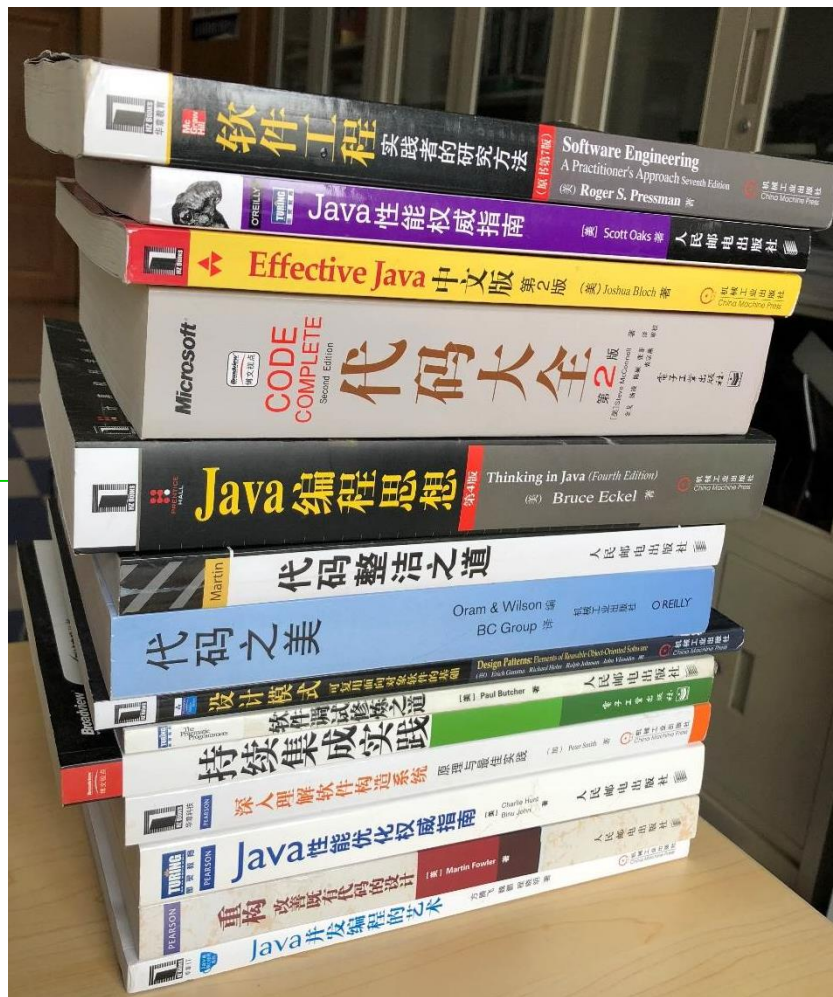
真的无所谓，有那么多参考书汗牛充栋



# 无教材 但有10+本参考书 ...so?

我们的课程内容，是这些参考书的“拼盘”。

真正要深入理解，以后的编程中还是要系统性的读书。





# MIT 6.031 + CMU 15-214 so?

名校，值得学习  
不管是你们还是我们



## 英文的讲义 so?

你应该已经没有了  
对英文讲义和阅读材料的恐惧  
掌握更多的专业词汇，你  
会在以后感受到巨大的益处





哈爾濱工業大學  
HARBIN INSTITUTE OF TECHNOLOGY

5学分  
80学时  
延续17周  
闭卷考试  
no cheat sheet...  
so?

那就启动复习呗!  
还有三周多时间, 足够了

## 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

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

### COMMENTS

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

### ELEMENTS

<b>Declaration</b>	
<%! declaration %>	Creates a global variable or method.
<b>Expression</b>	
<%= expression %>	Statements evaluated on the server before the page is outputted to the client.
<b>Page Directive</b>	
<%@ directive %>	Attributes that apply to the entire page.
<b>Scriptlet</b>	
<% code fragment of one or more lines %>	Contains a block of scripting code which is executed when the page is generated.
<b>Taglib Directive</b>	
<%@ taglib uri="URI To TagLibrary" 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
<b>(Integers)</b>		
byte	Byte-length integer	8
short	Short integer	16
int	Integer	32
long	Long integer	64
<b>(Real Numbers)</b>		
float	Single-precision floating point	32
double	Double-precision floating point	64
<b>(Other)</b>		
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





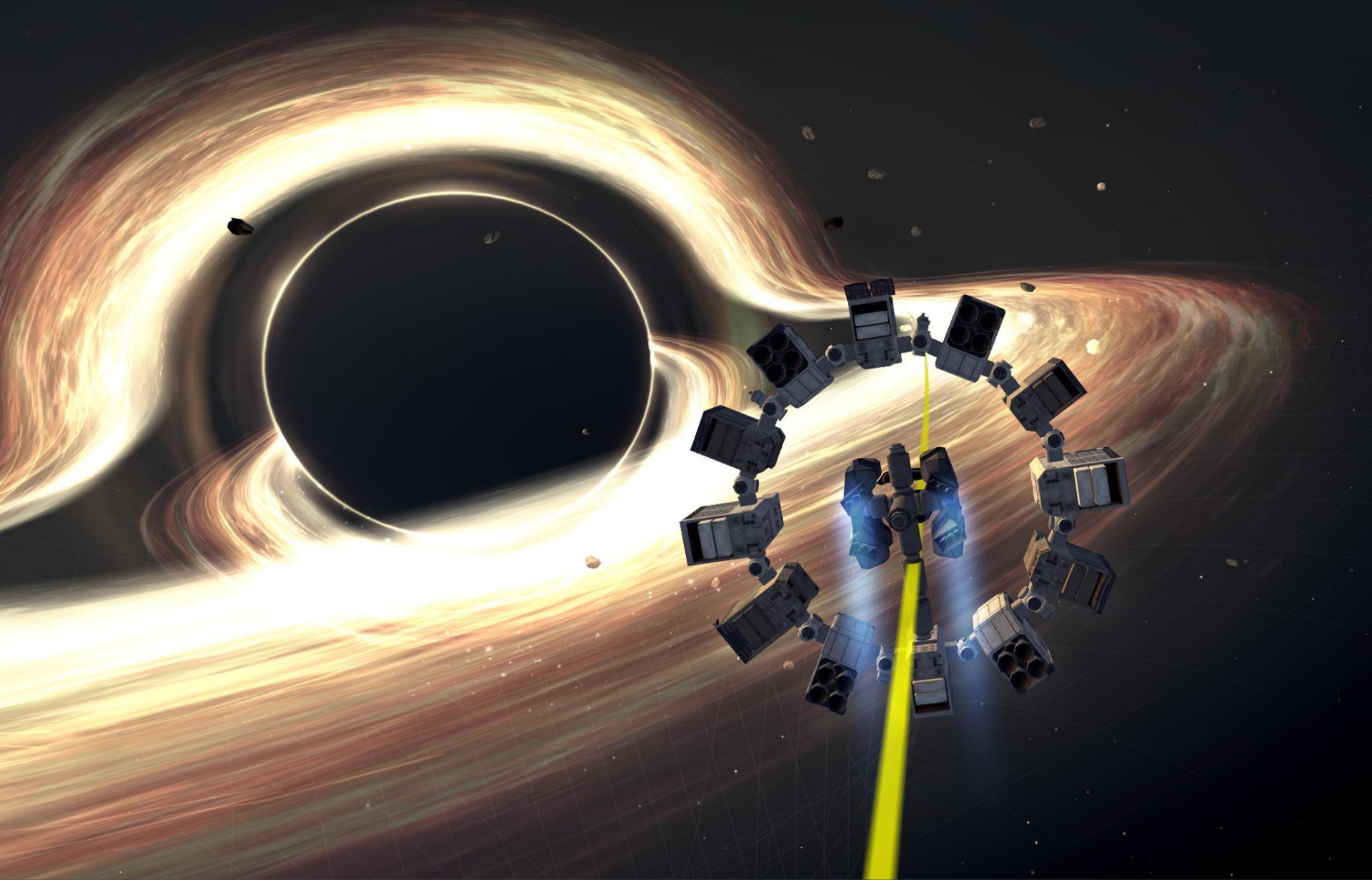
CALCULATING TRAJECTORY

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  
多与教师/TA交流  
多总结多反思/写博客

# 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](mailto:rainy@hit.edu.cn)  
[rainy.wang@gmail.com](mailto: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 30, 2018