

计蒜客与MOOC教学应用

计蒜客布道师 钦恩强

目录

- MOOC 的发展与衍生
- 计蒜客的互动式教学
- 校园推广与合作模式

当MOOC还不叫MOOC时

项目名称	时间	特点	失败原因
<i>NYUonline</i>	2000-2001	营利性『远程学习』企业	<ol style="list-style-type: none"> 1. 缺乏调研 2. 大学官僚主义 3. 销售和产品 沟通
<i>Fathom</i>	2000-2003	高质量，由政府、大学、图书馆、博物馆等各方势力共同组建	缺乏聚焦
<i>AllLearn</i>	2001-2006	更近似于现在的MOOC	注册人数太少
<i>UoPeople</i>	2009	非营利但未经许可的机构 学费免费，考试收费	没有教育部提供的资格认证，难以被认可

第一个 MOOC

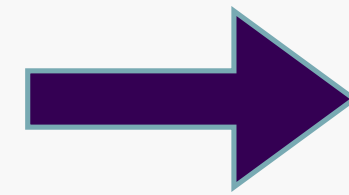
- 曼尼托巴大学 (University of Manitoba) 2008 24 2200
- 2011年 斯坦福 Peter Norvig & Sebastian Thrun 人工智能课 16万 2.3万

三大MOOC平台

	<i>Coursera</i>	<i>edX</i>	<i>Udacity</i>
是否盈利?	是	否	是
合作伙伴	19所高等教育机构, 其中5所美国之外	哈佛、麻省理工学院、加州大学伯克利分校	用大企业来给课程背书; 皮尔逊教育作为考试平台提供方
学生来源	39%来自美国, 还有巴西、印度、中国、加拿大、英国、俄罗斯、德国	美国、印度、英国、哥伦比亚、西班牙、巴基斯坦、加拿大、巴西、希腊	1/3来自美国, 1/3来自巴西、加拿大、欧洲、东亚, 1/3来自其他185个国家
费用	早期: 不收费 现在: 课程+认证 (主)	早期: 不收费 现在: 认证	早期: 皮尔逊教育考试收费 现在: 课程 (主) + 认证

MOOC的教学模式

1. 大规模参与
2. 在线、开放的教学资源
3. 一系列的教学环节与流程
4. 嵌入相关教学资料或测试
5. 机器评估、同学评估和自我评估
6. 网上学习讨论

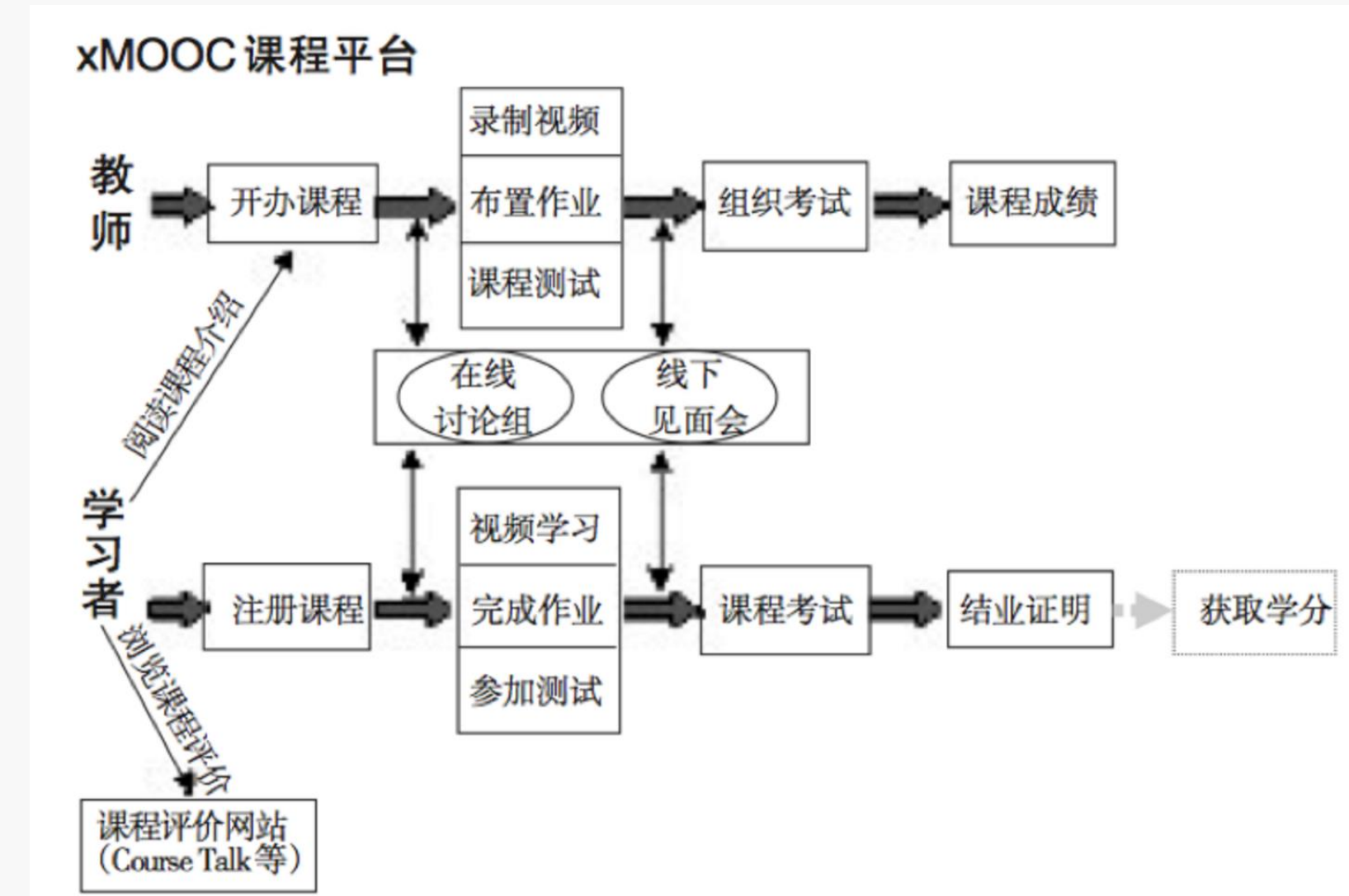
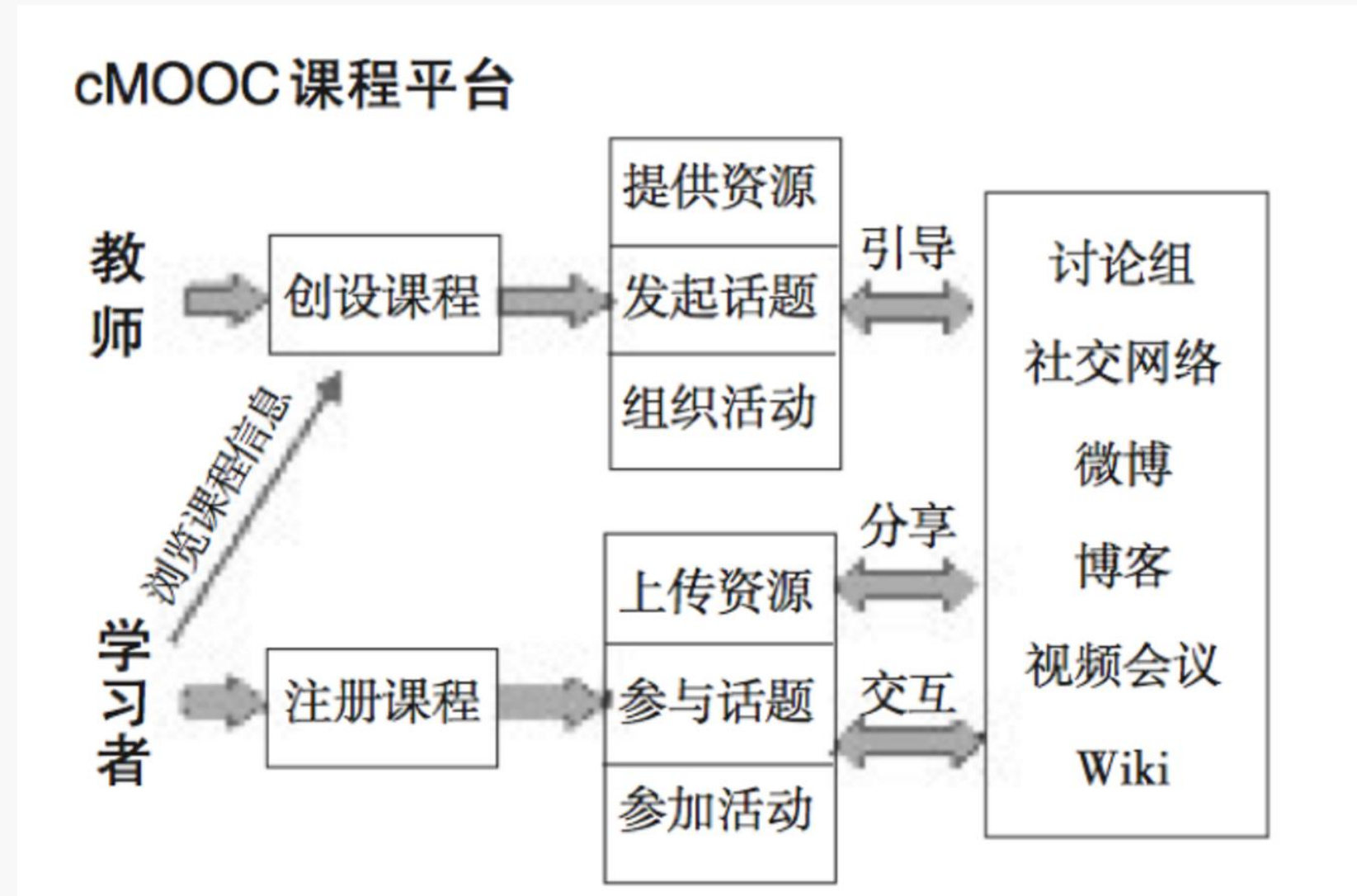


1. 隐身者：注册，但未参与
2. 尝鲜者：参与，但「品尝」下就走
3. 脱离者：前中期积极参与，后期退出
4. 旁听者：只看教学内容，不参与测试和评估
5. 完成者：完成了大部分课程及评估

更有意义的MOOC

- 分类:
xMOOCs

cMOOCs



MOOC的衍生

- SPOC: 小班 (Small)、私有 (Private) 在线课程
- MPOC: 大班 (Massive)、私有 (Private) 在线课程

差异一：教师团队的构成

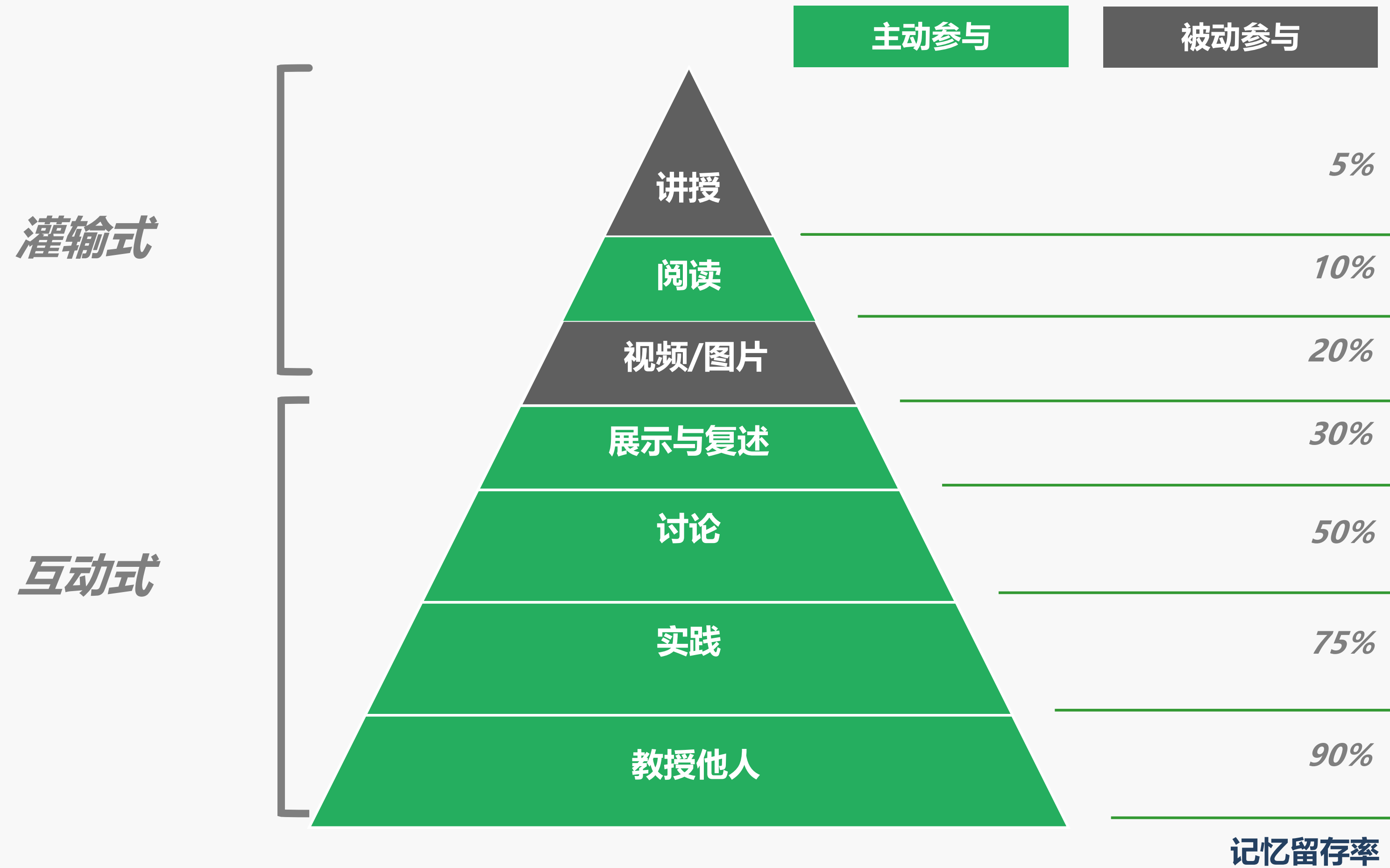
「一名教师 + 一组助教」 VS 「一个网络课程设计者 + 多个负责在线学习的教师」

差异二：内容授权和在线课堂租赁的方式

$MPOC = SPOC * N$

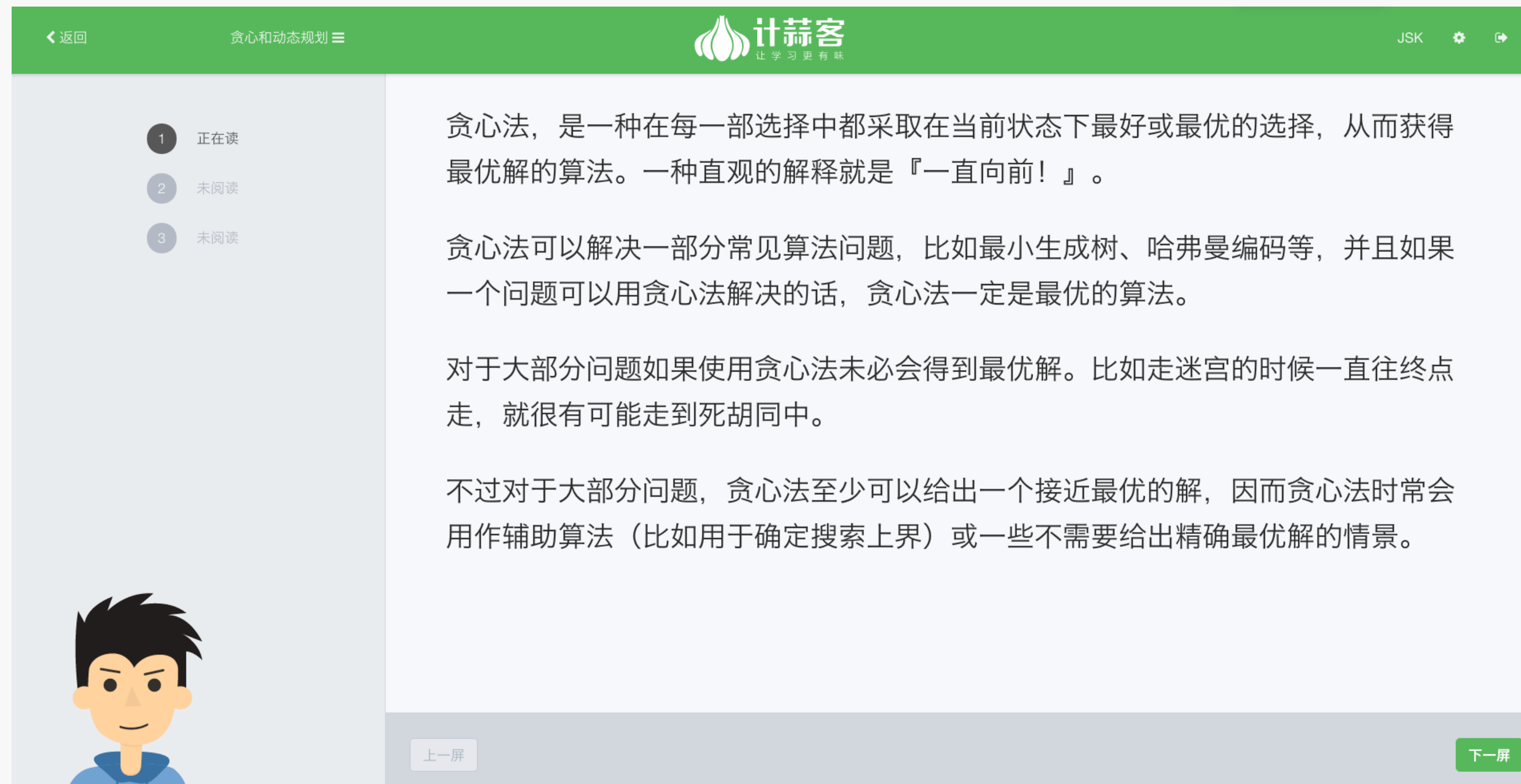
跨越组织共建 MPOC

学习是怎样的？



互动式的教学体验

- 概念阅读课



The screenshot shows a web interface with a green header. On the left is a sidebar with a navigation menu: '1 正在读' (highlighted), '2 未阅读', and '3 未阅读'. Below the menu is a cartoon character of a man with black hair. The main content area has a green top bar with a back arrow, the title '贪心和动态规划', the '计蒜客' logo, and 'JSK' with settings and share icons. The text content is as follows:

贪心法，是一种在每一部选择中都采取在当前状态下最好或最优的选择，从而获得最优解的算法。一种直观的解释就是『一直向前!』。

贪心法可以解决一部分常见算法问题，比如最小生成树、哈弗曼编码等，并且如果一个问题可以用贪心法解决的话，贪心法一定是最优的算法。


对于大部分问题如果使用贪心法未必会得到最优解。比如走迷宫的时候一直往终点走，就很有可能走到死胡同中。

不过对于大部分问题，贪心法至少可以给出一个接近最优的解，因而贪心法时常会用作辅助算法（比如用于确定搜索上界）或一些不需要给出精确最优解的情景。

At the bottom of the content area, there are two buttons: '上一屏' (Previous Screen) and '下一屏' (Next Screen).

互动式的教学体验

- 跟随式代码课



太棒了~! 你可以试试运行这个程序, 如果你觉得还不过瘾, 我们在这里还允许你尝试诸如正弦函数 `sin`、余弦函数 `cos`, 你也可以试试噢。

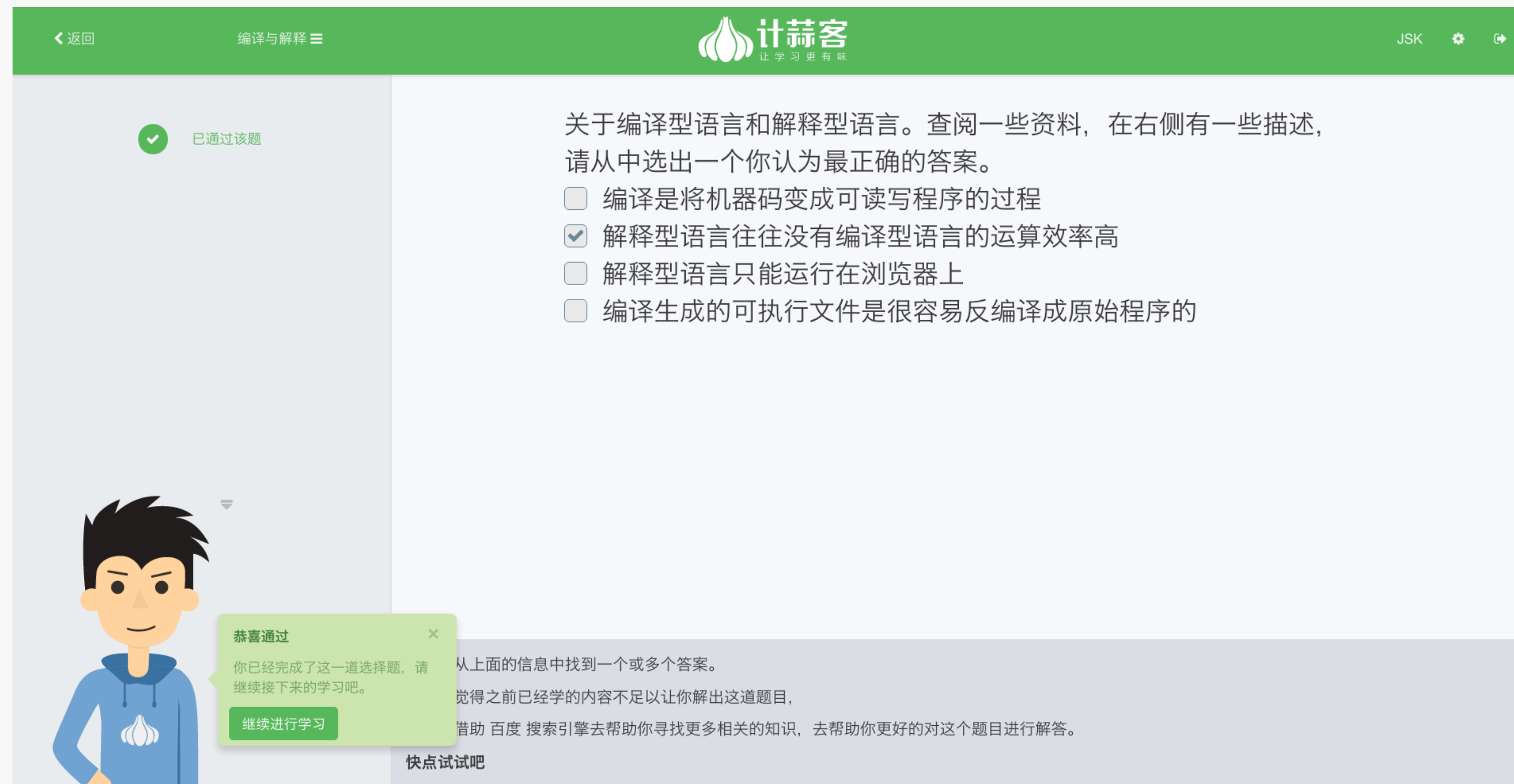
```
main.cpp
1 #include <iostream>
2 #include <cmath>
3
4 using namespace std;
5
6 int main() {
7     //请在下面运用数学函数输出一些结果
8     cout << cos(100) << endl;
9     return 0;
10 }
```

恭喜通过
你已经成功通过了这个题目的考验, 你可以继续学习了。
继续进行学习

重做 运行

互动式的教学体验

- 选择题



The screenshot shows a web interface for a multiple-choice question. At the top, there is a green navigation bar with a back arrow, the text "编译与解释", the "计蒜客" logo, and the tagline "让学习更有味". On the right side of the bar are the letters "JSK", a settings gear icon, and a share icon. The main content area has a light gray background. On the left, a green checkmark icon is followed by the text "已通过该题". Below this, a cartoon character of a man with black hair and a blue hoodie is shown. A green notification box with a white border and a close button (X) is positioned over the character. The notification text reads: "恭喜通过", "你已经完成了这一道选择题, 请继续接下来的学习吧。", and a green button labeled "继续进行学习". To the right of the character, the question text is: "关于编译型语言和解释型语言。查阅一些资料, 在右侧有一些描述, 请从中选出一个你认为最正确的答案。". Below the question are four radio button options: "编译是将机器码变成可读写程序的过程", "解释型语言往往没有编译型语言的运算效率高", "解释型语言只能运行在浏览器上", and "编译生成的可执行文件是很容易反编译成原始程序的". The second option is selected with a blue checkmark. At the bottom of the interface, there is a light gray footer area with the text "快点试试吧".

互动式的教学体验

- 配对题



工程师的能力评价

admin

我们对于工程师的能力评价有了一定的了解后，请在下面的选项中进行点击，对两个属于同一评价体系的选项进行配对。

写码的代码行数 单一项目的时间预估 产品的功能点数 修改的代码行数

代码的提交次数 开发 100 行代码用时 用户的不可用反馈

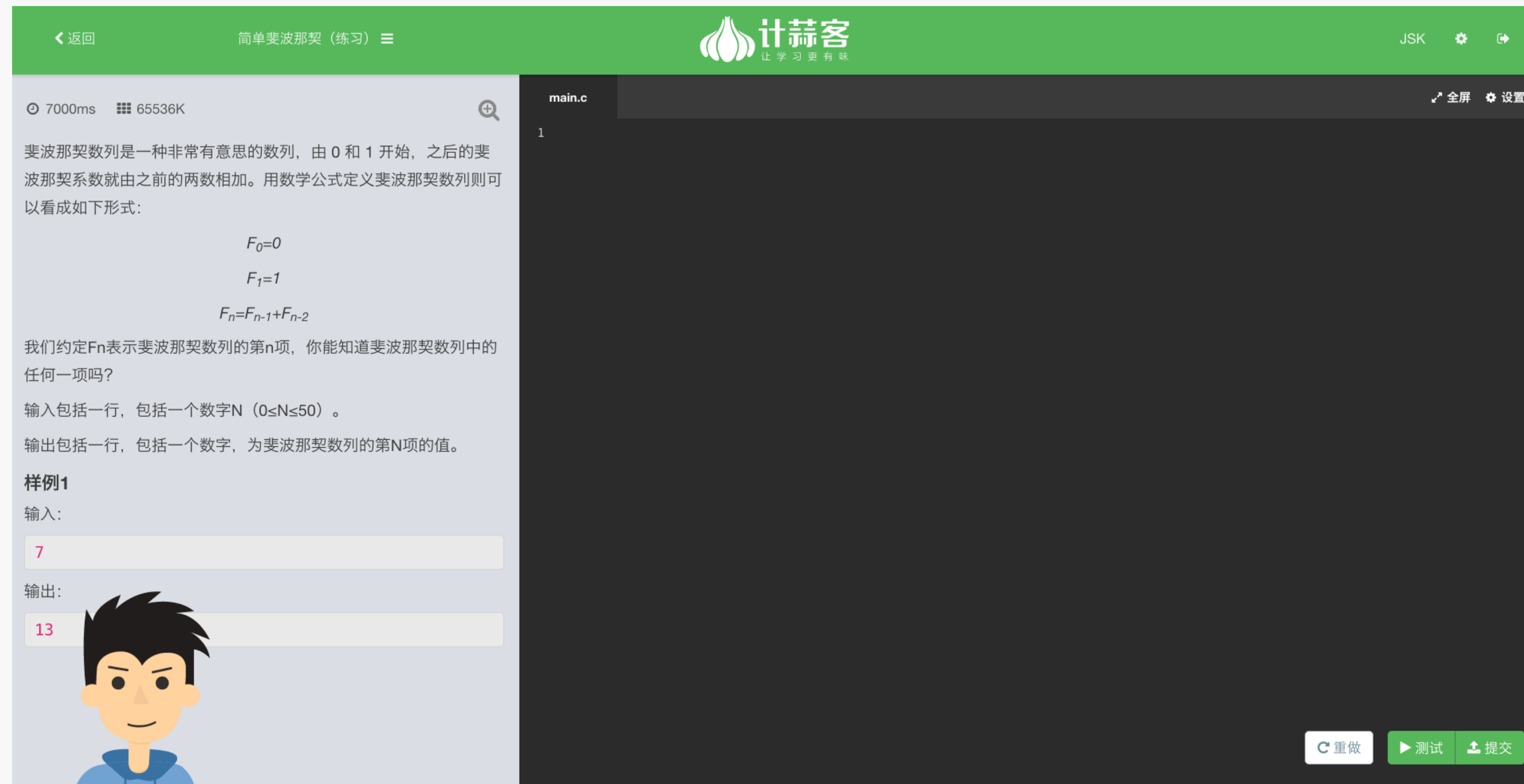
一个功能模块开发用时 一贯的时间预估准确性 写工作日记的行数

提示信息：
亲，好遗憾哦。配对不正确，再试试！
关闭

快点试试吧

互动式的教学体验

- 挑战难题



简单斐波那契 (练习) JSK 设置

7000ms 65536K

斐波那契数列是一种非常有意思的数列，由 0 和 1 开始，之后的斐波那契系数就由之前的两数相加。用数学公式定义斐波那契数列则可以看出如下形式：

$$F_0=0$$

$$F_1=1$$

$$F_n=F_{n-1}+F_{n-2}$$

我们约定 F_n 表示斐波那契数列的第 n 项，你能知道斐波那契数列中的任何一项吗？

输入包括一行，包括一个数字 N ($0 \leq N \leq 50$)。

输出包括一行，包括一个数字，为斐波那契数列的第 N 项的值。

样例1

输入：
7

输出：
13

main.c

```
1
```

重做 测试 提交

互动式的教学体验

- 约20%的课程完结率
- 知识点拆分到细节 确保学生每一点都掌握
- 全程得到记录 每一次敲击都带来内容价值
- 实践是工科学科掌握知识的唯一途径

课程体系分级

CS 100 编程初学入门常识 CS 111 C语言 CS 112 C++ CS 114 Ruby
 CS 113 Python CS 115 Javascript CS 116 Java CS 161 离散结构
 CS 141 数据形式与基础数据处理 CS 151 计算机导论 CS 171 网页制作设计与方法

针对中国学生基础教育阶段
 计算学科教育缺失的现状
 设置第一层级课程用于补足
 计算学科基础知识

CS 200 程序设计方法与习惯 CS 211 面向对象的程序设计 CS 221 操作系统入门与系统编程
 CS 241 数据库引论与关系数据库 CS 251 计算机组成原理 CS 261 数据结构
 CS 271 数字版权导论 CS 272 计算、设备与健康

第二层级的课程
 对计算学科
 后续核心课程的开展
 做好导论准备

CS 321 操作系统设计 CS 340 数值方法
 CS 341 数据库进阶与非关系数据库 CS 361 计算理论
 CS 362 算法引论 CS 371 计算发展史 CS 381 工程与通信中的安全机制
 CS 411 软件工程 CS 421 嵌入式系统 CS 422 实时系统 CS 423 并行计算 CS 431 计算机图形学 CS 432 科学数据可视化

精简第三层级课程
 强化这一部分核心课程
 对于逻辑与设计等方面的要求

CS 433 社会数据可视化 CS 434 计算摄影学 CS 435 计算生物学 CS 440 数值分析 CS 441 复杂数据系统与数据挖掘
 CS 442 信息检索与自然语言处理引论 CS 443 计算广告学 CS 451 计算机体系结构 CS 461 算法 CS 462 人工智能导论
 CS 463 机器学习导论 CS 464 信息论 CS 465 程序语言与编译器 CS 466 形式计算建模 CS 467 形式验证 CS 492 信号系统
 CS 471 计算机研究方法与伦理 CS 472 界面与交互设计 CS 473 城市计算导论 CS 481 系统安全导论 CS 482 网络安全导论

根据计算学科的子方向
 全面的对接工业界需求
 在扎实的核心课程的基础上
 进行垂直方向课程教学
 鼓励在某一两个方向学深入

CS 483 密码学导论 CS 491 数字信号处理 CS 493 多媒体与多媒体信息 CS 494 网络与通信导论 CS 495 无线网络通信导论

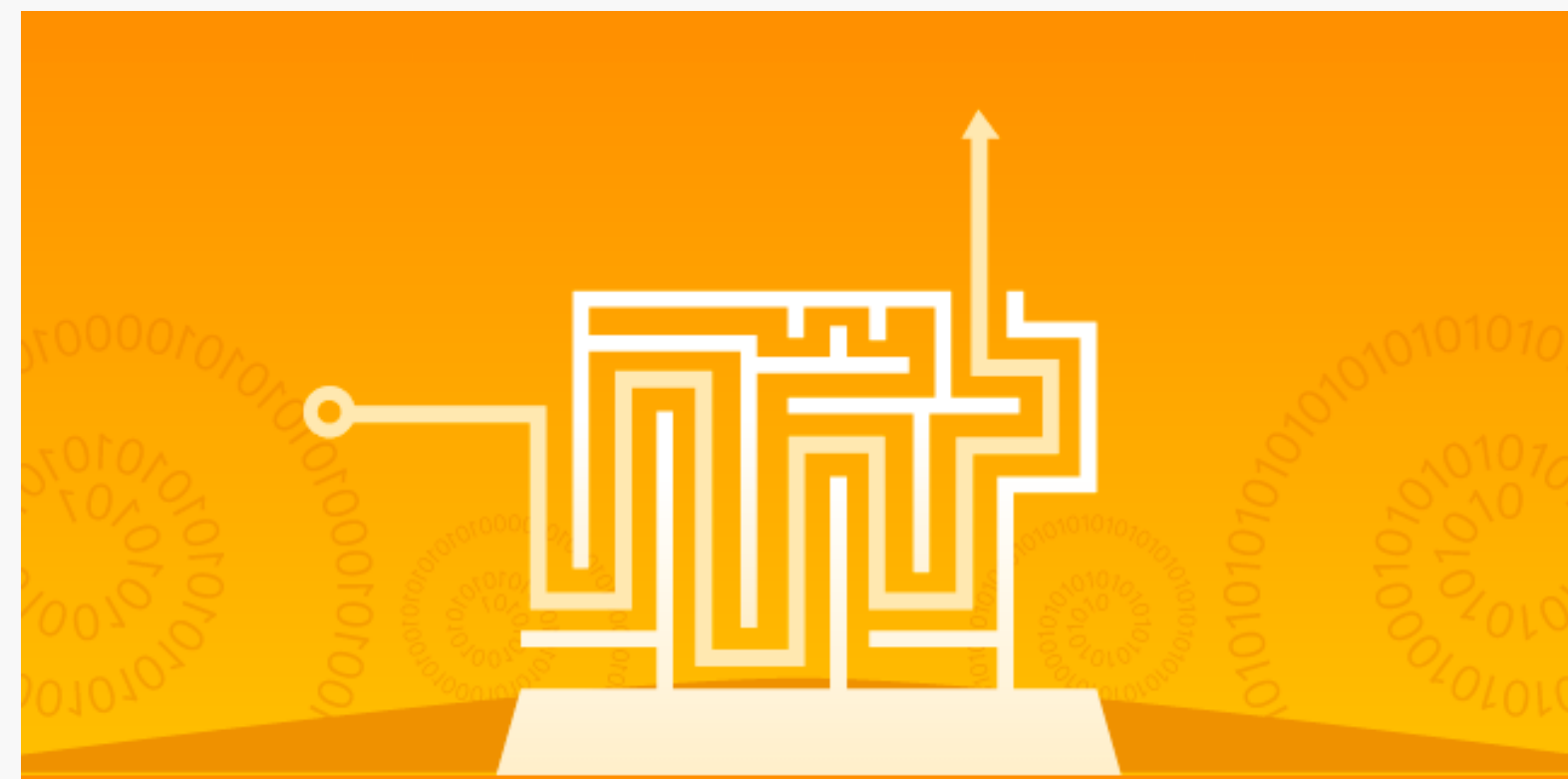
课程体系分级



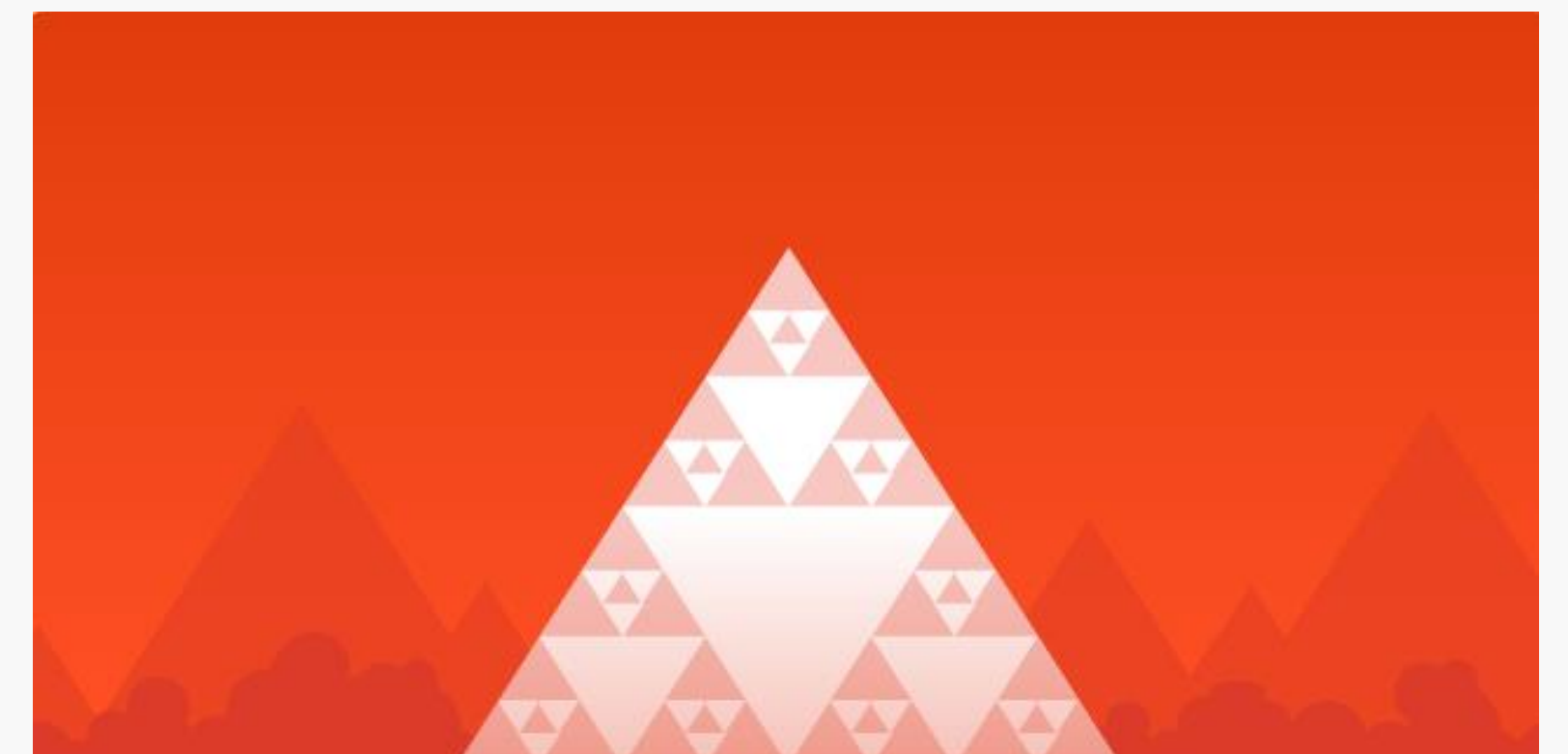
CS 113: Python 程序设计



CS 261: 数据结构



CS 362: 算法导论



CS 461: 算法

按照方向划分

基础课	程序开发与软件工程	系统	图形图形与交叉应用	数据与数值	始源/体系结构	计算理论/算法	人机交互 普适计算应用	安全	信号/通信/网络
CS 100 编程初学入门常识	CS 111 C语言 CS 112 C++ CS 113 Python CS 114 Ruby CS 115 Javascript CS 116 Java			CS 141 数据形式与基础数据处理	CS 151 计算机导论	CS 161 离散结构	CS 171 网页制作设计与方法		
CS 200 程序设计方法与习惯	CS 211 面向对象的程序设计	CS 221 操作系统入门与系统编程		CS 241 数据库引论与关系数据库	CS 251 计算机组成原理	CS 261 数据结构	CS 271 数字版权导论 CS 272 计算、设备与健康		
	CS 311 设计模式与工程	CS 321 操作系统设计		CS 340 数值方法 CS 341 数据库进阶与非关系数据库		CS 361 计算理论 CS 362 算法引论	CS 371 计算发展史	CS 381 工程与通信中的安全机制	
CS 411 软件工程	CS 421 嵌入式系统 CS 422 实时系统 CS 423 并行计算	CS 431 计算机图形学 CS 432 科学数据可视化 CS 433 社会数据可视化 CS 434 计算摄影学 CS 435 计算生物学	CS 440 数值分析 CS 441 复杂数据系统与数据挖掘 CS 442 信息检索与自然语言处理引论 CS 443 计算广告学	CS 451 计算机体系结构	CS 461 算法 CS 462 人工智能导论 CS 463 机器学习导论 CS 464 信息论 CS 465 程序语言与编译器 CS 466 形式计算建模 CS 467 形式验证	CS 471 计算机研究方法与伦理 CS 472 界面与交互设计 CS 473 城市计算导论	CS 481 系统安全导论 CS 482 网络安全导论 CS 483 密码学导论	CS 491 数字信号处理 CS 492 信号系统 CS 493 多媒体与多媒体信息 CS 494 网络与通信导论 CS 495 无线网络通信导论	

校园推广与合作

直接向老师推荐

由学生反向推荐

- 优：速度快、途径多
- 劣：缺乏合作基础、转化率低

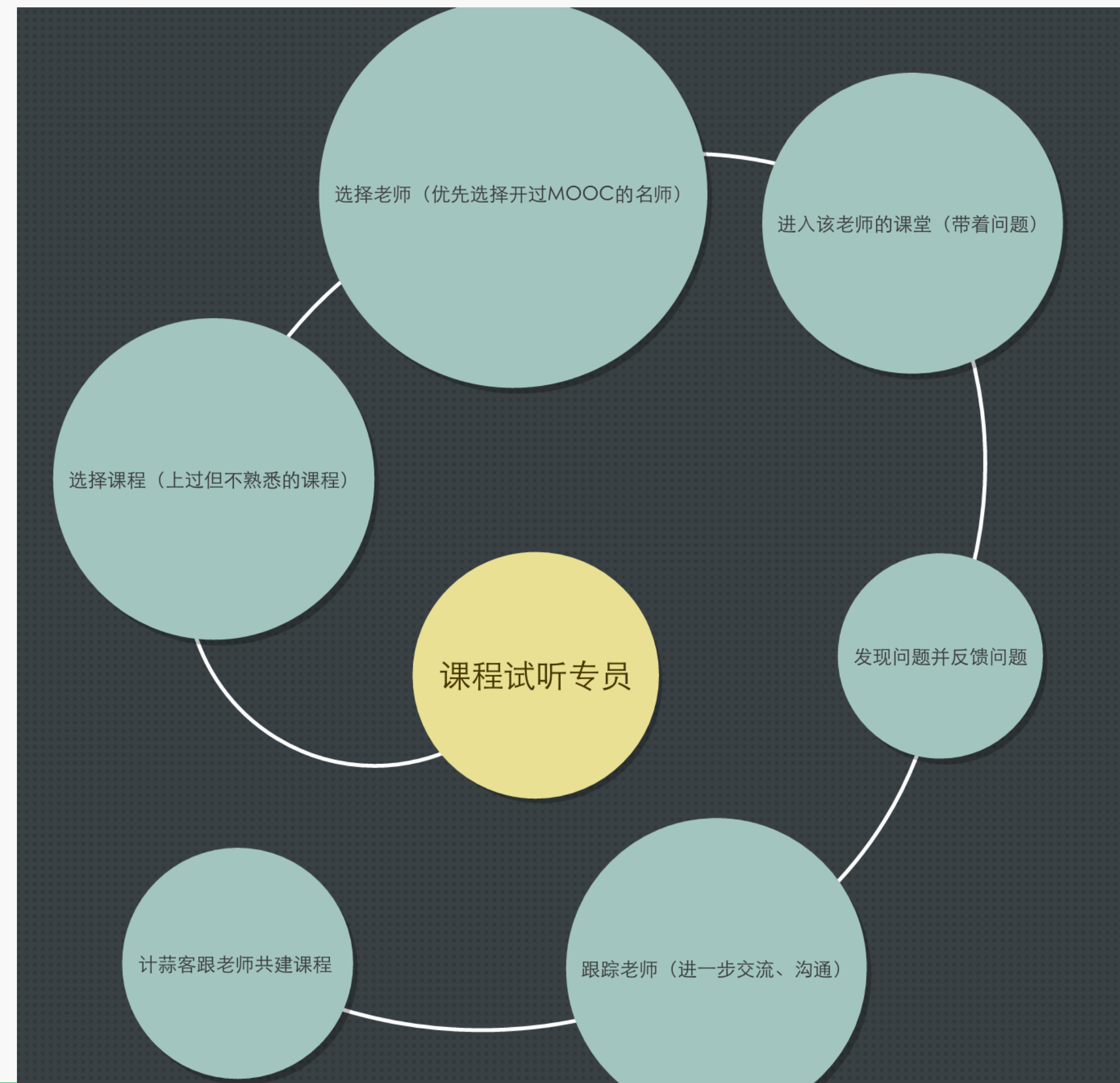
- 优：合作深入、效果好
- 劣：周期长、难把控

校园推广与合作

- 课程试听专员计划

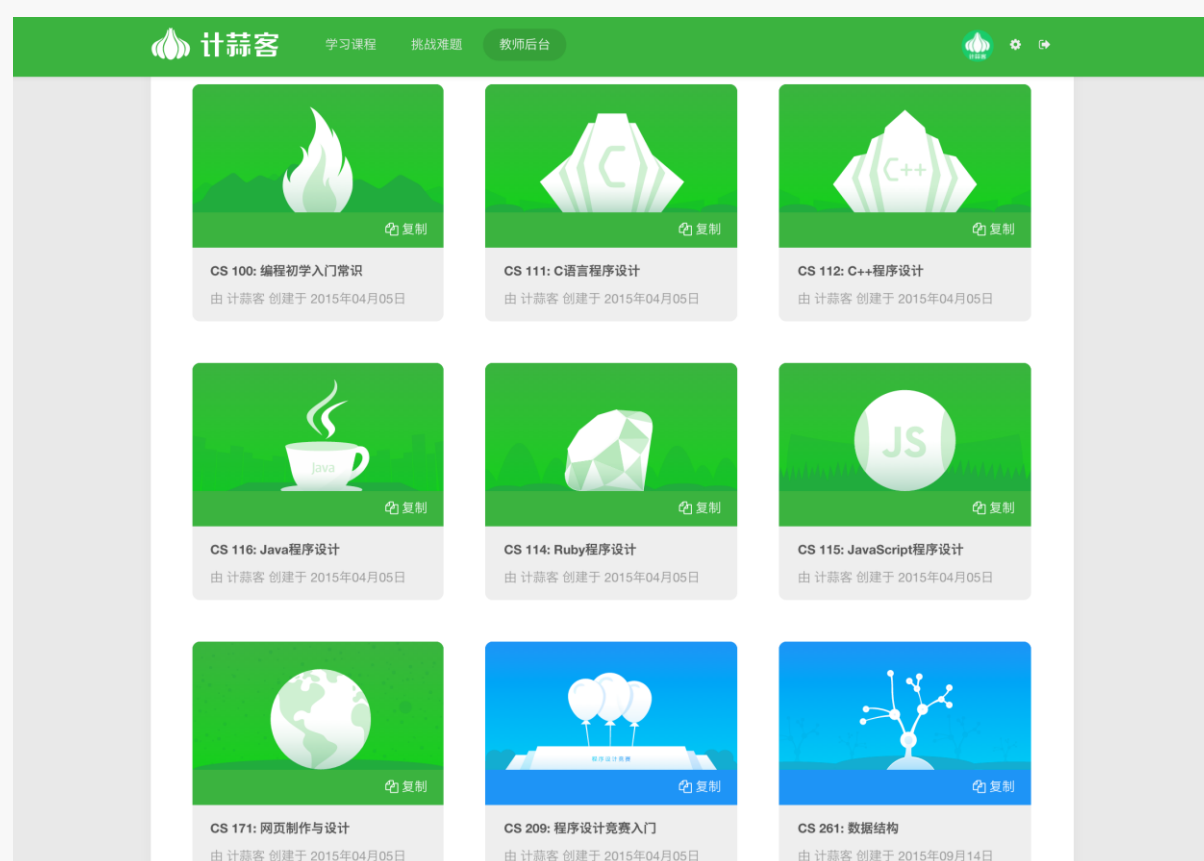
少而精

单点突破



教学辅助工具：教师后台

- 自由创建，设计一门津津有味的课程
- 各式各样的课程类型，让不同知识以更适合的形式教学
- 复用所有公开课程



教学辅助工具：教师后台

▼ C语言入门尝试
最后编辑于10秒前

- 什么是C语言
- ⟨> Hello World (旧)
- ? 错误的转义字符
- 常见变量
- ⟨> 变量声明 (旧)
- 代码风格之换行
- ? 正确的换行
- ⟨> 新的代码课
- > C语言的基本控制
最后编辑于10秒前
- > C语言的分支结构
最后编辑于9秒前
- > C语言的逻辑与循环结构
最后编辑于9秒前
- > C语言的函数
最后编辑于9秒前
- > C语言的数组
最后编辑于9秒前
- > C语言的结构
最后编辑于8秒前
- > C语言中的位运算
最后编辑于8秒前
- > 内存与C语言中的指针

标题

课程内容
代码框中的代码是需要学生在当前步骤匹配的所有代码，匹配成功时才会显示这一步的引导信息。

main.c x + C

```

1 #include <stdio.h>
2 int main(){
3     printf("Hello");
4     return 0;
5 }
```

B I U S |≡≡ “ ‹› %

恭喜你完成啦~!

字数: 7/400

上一步 1 2 3 x

B I U S |≡≡ “ ‹› %

请在这里输入卡顿提示信息

字数: 0/50

添加新步骤

我的课程 / Java

课程管理 学生管理

- > Java入门尝试
最后编辑于14小时前
- ▼ Java的基本控制
最后编辑于14小时前
- ⟨> 五则运算 (旧)
- ⟨> 类型转换 (旧)
- 标识符
- ? 哪一个不是保留字?
- ⟨> 从控制台读取输入 (旧)
- 📄 A+B+C问题 (练习)
- 类和对象
- 赋值语句和赋值表达式
- ⟨> String类型 (旧)
- > Java的分支结构
最后编辑于14小时前
- > Java的逻辑与循环结构
最后编辑于14小时前
- > Java的数组
最后编辑于14小时前
- +

标题

阅读内容

B I U S |≡≡ “ ‹› %

我们在之前的代码中看到的，类似于i1, i2, input这些变量名，都有一个统一的名字叫做标识符。Java中的标识符有以下这些命名规则：

标识符由字母、数字、下划线 (_) 和美元符号 (\$) 所构成
标识符必须以字母、下划线或美元符号开头，不能以数字开头
标识符不能是保留字
标识符不能是true、false或null
标识符可以有任意长度

上一页 1 x 2 3 下一页

关键字

保存 删除

教学辅助工具：教师后台

我的课程 / Java

课程管理
学生管理

- > Java入门尝试
最后编辑于14小时前
- ▼ Java的基本控制
最后编辑于14小时前
- </> 五则运算 (旧)
- </> 类型转换 (旧)
- 标识符
- ? 哪一个不是保留字?
- </> 从控制台读取输入 (旧)
- 📄 A+B+C问题 (练习)
- 类和对象
- 赋值语句和赋值表达式
- </> String类型 (旧)
- > Java的分支结构
最后编辑于14小时前
- > Java的逻辑与循环结构
最后编辑于14小时前
- > Java的数组
最后编辑于14小时前

标题

问题描述

B I U S 列表 列表 列表 列表 列表 列表 列表

上节课我们认识了java中的一些标识符，那么在右边的选项中，哪一个不是保留字呢？

选项

<input type="checkbox"/> 您可以在这里撰写新选项	添加
[✓] foreach	✖
final	✖
finally	✖
for	✖

关键字

- </> 快速排序的实现 (旧)
- 贪心和动态规划
- </> 数塔问题 (旧)
- </> 最长上升子序列 (旧)
- ? 字典序最小问题
- 📄 蒜头爬楼梯
- ▼ 数据结构入门
最后编辑于14小时前
- 线性数据结构
- ? 数据结构的选择
- 📄 最近通话记录
- 树形数据结构
- 📄 冗余关系
- </> 堆排序 (旧)
- ? 二叉搜索树
- ▼ 图论入门
最后编辑于14小时前
- 图的存储
- 图论经典问题
- ? 最短路问题
- 📄 蒜国高速路
- ? 生成树问题

放"+", 再放"-", 这么放的原因是因为"+"的ASCII码要比"-"小。

样例输入

7

样例输出

1+2-3+4-5-6+7
1+2-3-4+5+6-7
1-2+3+4-5+6-7
1-2-3-4-5+6+7

提示信息

初始代码

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 int n; // 保存读入的整数N
4 bool opr[10]; // 可以用一个bool数组保存所有n-1个符号，我们用true表示加号"+", 用false表
5 bool found = false; // 表示是否找到一组解，如果没找到的话需要在最后输出"None"
6 // dfs函数，保存两个状态：deep表示递归深度，就是枚举到第几个数字左边的符号；sum表示之前部分
7 // 如果枚举完成后sum的值刚好为0，则输出这组方案。我们需要在搜索的过程中用opr数组保存状态。
8 void dfs(int deep, int sum) {
9     // 请在下面的条件表达式中填上对应的值。
10    if (deep == ) {
11        if (sum == 0) {
12            found = true;
13            // 请在下面将最终方案输出，方案格式参照题目描述。
14            // 最后记得换行哦。
15        }
16    }
17    return ;
18 }
19 // 请在下面的语句中填入正确的逻辑。
20 opr[deep] = ;
    
```

测试数据

注意事项:

1. 请将输入数据文件命名为"inx.in", 输出数据文件命名为"stdX.out" (X为从1开始的整数, 用来标识测试数据的组数)
2. 请一次性上传所有测试数据文件, 重新上传将覆盖之前的文件
3. 上传完成后, 请点击保存按钮保存所有信息

关键字

合作案例

- 预习 + 复习 + 上机练习



软件工程

已完成 79 % 课程内容的学习

清华：软件工程

建议 1 周完成学习



Python

151 位同学正在学习

清华：Python

建议 4 周完成学习



Javascript

150 位同学正在学习

清华：JavaScript

建议 3 周完成学习

合作案例

- 考试



北京大学
PEKING UNIVERSITY

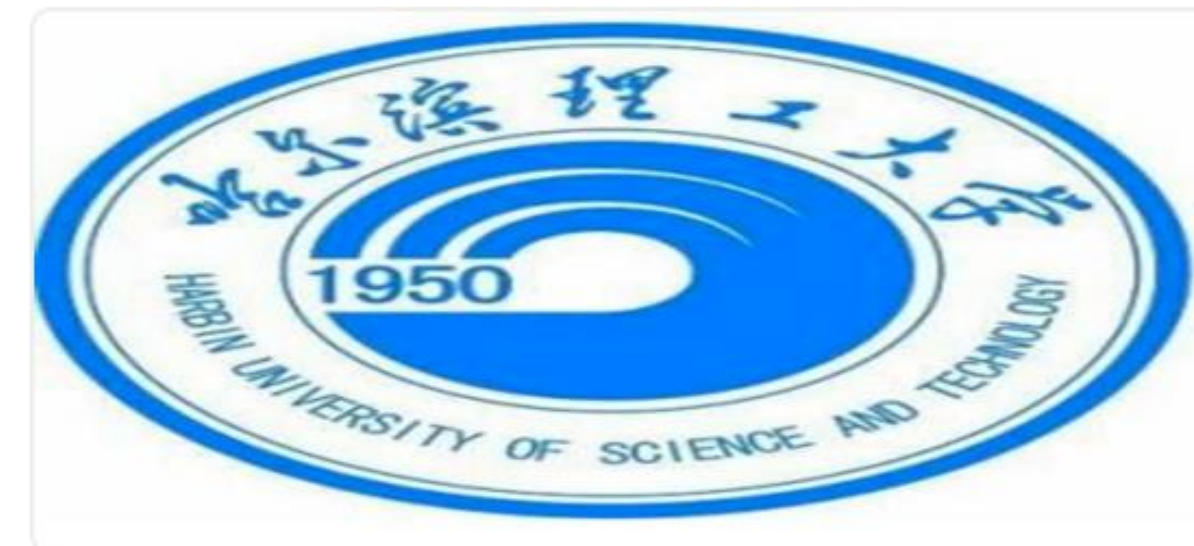
北京大学软件与微电子学院

已有 64 人报名

北京大学软件学院 2015 新生编程技能测验

比赛时间: 2015年9月10日 10:15 - 11:45

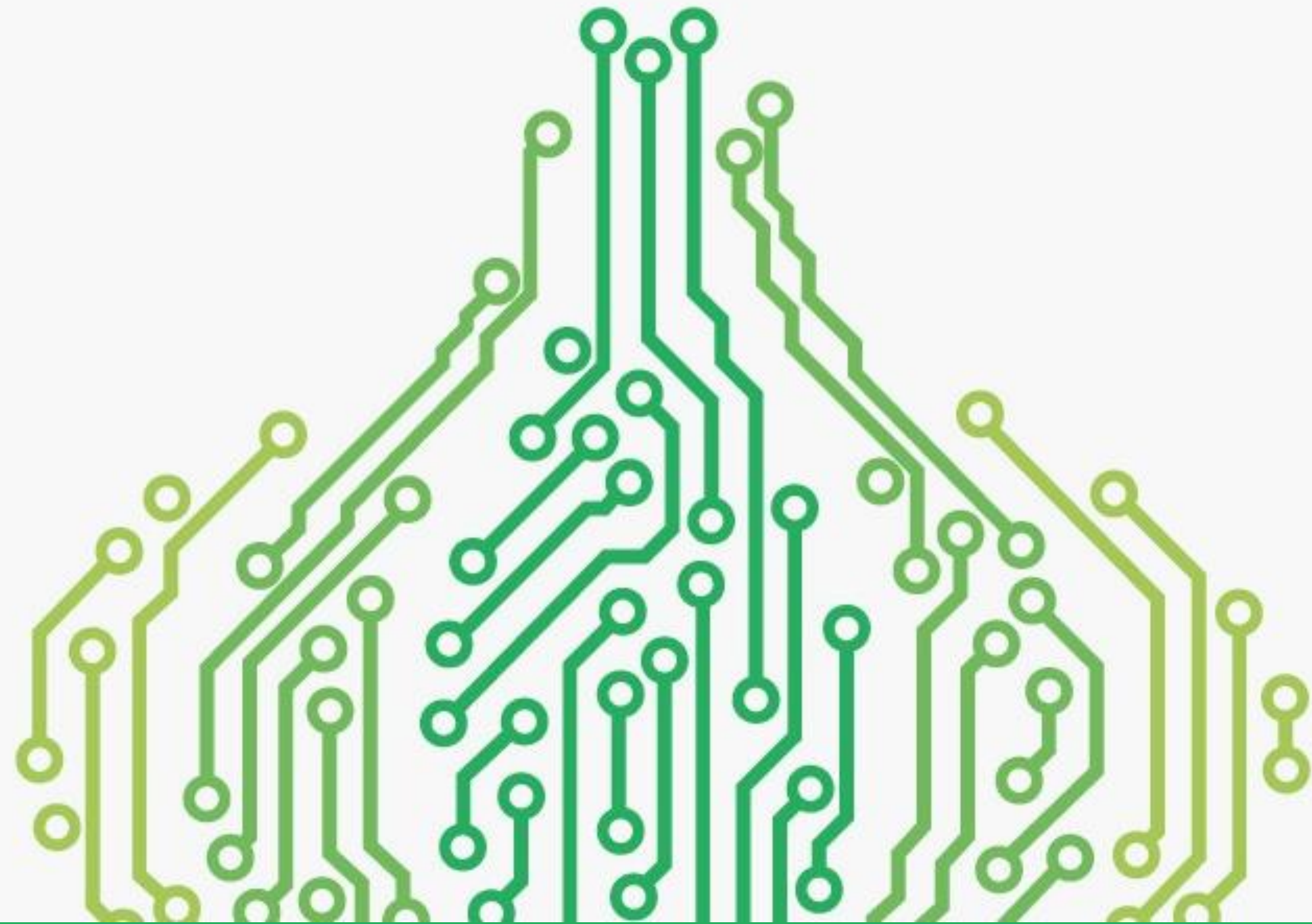
比赛信息



已经结束: 2015年11月22日 16:30

参加人数: 36

“计蒜客杯”哈理工软件学院第五届程序设计大赛 高年级组



谢谢大家！