课设题目和要求

# 1.现代化教务管理系统

## 1.1 难度系数

|  |  |  |
| --- | --- | --- |
| 项目 | 基本等级 | 最高等级 |
| 工程量 | ☆☆☆☆ | ☆☆☆☆☆ |
| 算法挑战 | ☆ | ☆☆☆ |
| 学习成本 | ☆ | ☆☆☆ |
| 难度系数 | 0.85 | 1.1 |

## 1.2 需求定义

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 需求描述 | 基本要求 | 进阶要求 | 最高要求 | 分数 |
| 1 | 学生能够通过该系统查询自己的课表 | 以周历的形式展示出课表，以及每门课的上课时间、教室、周数、任课教师 | 能够按照所选的周数展示课表，比如（但不限于）前八周、后八周，能够根据课程类别（比如必修课、公选课）展示课表 | 能够根据课表，以邮件、短信、或微信的方式提醒学生每日上课的时间和地点。注意，这种提醒可以随时开启，也可以随时终止 | 5/7/8 |
| 2 | 学生能够在学期开始前通过该系统选课 | 根据自身的年级和专业，列出本学期可选课程、查看课程信息、并提交选课申请 | 系统帮助学生自动选择必修课 | 课程间的先修后修关系以及分数要求，提醒学生不可选择的课程 | 4/5/6 |
| 3 | 学生能够查询自己的成绩并生成成绩单 | 列出所有课程（包含统计信息：如已修学分、应修学分等），显示最终成绩，并可导出成绩单 |  |  | 3/3/3 |
| 4 | 老师能够通过该系统查询自己的课表 | 以周历的形式展示出课表，以及每门课的上课时间、教室、周数 |  | 能够根据课表，以邮件、短信、或微信的方式提醒学生每日上课的时间和地点。注意，这种提醒可以随时开启，也可以随时终止 | 5/5/8 |
| 5 | 老师能够通过该系统查询选课名单 | 列出所有选课学生的基本信息，包括学号、姓名、年级/班级、专业、是否重修等信息 |  |  | 3/3/3 |
| 6 | 老师能够通过该系统填报课程成绩，并生成成绩单 | 课程成绩由平时成绩和考试成绩组成，老师能够分别填写这两个成绩，系统能够根据老师选择的成绩计算公式自动核算出总成绩。成绩提交后能够导出成绩单 | 成绩录入可以通过导入excel表的方式完成 | 成绩录入不需要一次完成，可以暂存和自动保存 | 8/9/10 |
| 7 | 老师能够通过该系统对已经提交的课程成绩进行勘误 | 老师可以修改已经提交的课程成绩，每次修改都必须填写勘误理由。每次勘误行为会被系统记录。这些记录应当可以被任课教师和教务查看 |  |  | 5/5/5 |
| 8 | 教师能够通过该系统设定他所承担的课程的那个讲台的选课要求 | 选课要求包括年级、专业、班级、先修课程及成绩，也可以设定不可选择该课程的年级、专业和班级 |  | 在学生开始选课前，系统能够自动提醒老师设定选课要求 | 5/5/6 |
| 9 | 教师能够通过该系统申请停课、调课 | 停/调课申请包括：a) 课程，b) 申请停/调课的原时间，c) 申请调课的新时间，d) 申请理由。提交停/调课申请后，可以导出停/调课申请单 |  |  | 4/4/4 |
| 10 | 课程的责任教师能够通过该系统在下学期教学准备阶段提交开课申请 |  |  | 能够在线填写开课申请单。开课申请包括课程名称和ID、开课班级、划分的讲台、每个讲台对应的班级及授课教师，以及每个讲台的开课时间、地点和设备约束，具体应参考实际申请表。注意，课程的责任教师是在每次开课前由教务指定的 | 0/0/8 |
| 11 | 老师能够通过该系统填写成绩分析表 |  |  | 成绩分析表需要自动生成一个表头，包含了课程成绩的分数统计。成绩分析表的具体形式应该参考实际分析表。 | 0/0/5 |
| 12 | 学院教务能够通过该系统管理下发开课申请单 |  |  | 每次排课前，需要对每个课程指定一个责任教师，从而使得该老师可以填写申请单 | 0/0/4 |
| 13 | 学院教务能够通过该系统查询该学院里的每门课程的成绩 | 选择查看课程的全部讲台的成绩，也可以选择某个讲台进行查看 |  |  | 3/3/3 |
| 14 | 学院教务能够通过该系统查看每个学生成绩，并生成成绩单 | 对于选定的学生（包括已毕业），列出所有课程，显示最终成绩，并可导出成绩单 |  |  | 3/3/3 |
| 15 | 学院教务设定系所负责人 |  |  | 将系所负责人的权限赋予一位老师 | 0/0/2 |
| 16 | 学院教务可以通过该系统审核课程成绩 | 教师提交课程成绩后，教务可以审核该成绩。注意，如果教师对已提交的成绩进行了勘误，新的成绩依然需要审核 |  | 系统能够提醒教务有新的审核申请 | 4/4/5 |
| 17 | 学院教务可以通过进行排课 | 每个课程的每个讲台安排一个上课时间和教室。注意1，系统必须能够检查和提示上课时间、教室是否有冲突，冲突的情况包括待授课的班级在该时间已经有别的课、该教室在该时间已经被占用、该教室的容量不够、该教室的类型不符合需求等。注意2，课程安排可以是部分的，即可以只安排时间，不安排地点。注意3，学院教务只能排专业课和专选课，不能排公共课 | 系统能够列出无冲突的时间和教室 | 系统能够根据教师填写的开课申请单自动进行排课 | 10/12/20 |
| 18 | 学校教务可以进行学院教务的工作 | 学院教务具备学院教务的权限，可以管理任意学院。注意，学校教务可以进行公共课的排课工作 |  |  | 10/10/10 |
| 19 | 学校教务可以管理每个专业、每个年级的培养方案 | 可以录入新的培养方案、微调已有的培养方案。注意，微调的含义是新增新的课程 |  |  | 5/5/5 |
| 20 | 学校教务可以设定学生（专业）的培养方案 | 只有大一学生需要设定培养方案。培养方案的设定是以年级+专业为单位的。培养方案决定了每个专业每学期能够选修或必修的课程，也决定了教务在计划阶段需要安排的课程 |  | 可以通过excel的方式导入培养方案 | 4/4/5 |
| 21 | 教务能够审批停/调课申请 | 教师提出的申请需由教务审批。注意，学院和学校教务都需要对申请进行审批，先由学院进行审批，再由学校进行审批 |  | 根据审批进度，自动提醒教务需审批事项 | 4/4/5 |

# 2. MiniJava代码编辑器

## 2.1 难度系数

|  |  |  |
| --- | --- | --- |
| 项目 | 基本等级 | 最高等级 |
| 工程量 | ☆☆☆ | ☆☆☆☆ |
| 算法挑战 | ☆ | ☆☆☆☆ |
| 学习成本 | ☆☆☆ | ☆☆☆☆☆ |
| 难度系数 | 0.85 | 1.25 |

## 2.2 需求定义

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 需求描述 | 基本要求 | 进阶要求 | 最高要求 | 分数 |
| 1 | 基于Eclipse平台实现该编辑器 | 编辑器必须以Eclipse插件的形式发布 |  | 该编辑器利用xtext实现 | 25/25/40 |
| 2 | 语法高亮 | 编辑器能够对MiniJava代码进行语法高亮 |  | 允许用户定义语法高亮的颜色 | 10/10/13 |
| 3 | 语法语义检查 | 编辑器能够对MiniJava代码进行语法检查，并提示错误位置 | 编辑器能够检查语义错误，比如类型错误等 | 编辑器能够给出修正建议（至少能够进行演示） | 10/15/17 |
| 4 | 代码提示 | 编辑器能够进行代码提示，能够根据语法规则推荐用户可能输入的代码 |  | 编辑器能够根据上下文进行提示，比如能够推荐上下文中可以使用的变量、函数、类库等 | 10/10/15 |
| 5 | 项目管理 | 编辑器能够依赖Eclipse进行项目和文件管理 |  | 能够识别和处理MiniJava项目 | 5/5/10 |
| 6 | 代码重构 | 编辑器能够进行两个代码重构功能，重构的操作必须是不同的 |  | 编辑器能够进行四个代码重构功能 | 15/15/20 |
| 7 | 环境融合 | MiniJava能够导入和调用Java代码。注意，MiniJava和Java是不同的语言 |  | Java代码能够调用MiniJava | 10/10/15 |
| 8 | 编译执行 |  |  | 编辑器能够编译执行MiniJava程序。可能的方式是将MiniJava转换为Java程序，然后调用Java编译器完成编译 | 0/0/5 |

## 2.3 MiniJava语法定义

Goal = PackageDecl? ImportDecl\* MainClass ClassDecl\* EOF

PackageDecl = "package" QualifiedName ";"

QualifiedName = Identifier | QualifiedName "." Identifier

QualifiedNameList = QualifiedName | QualifiedNameList "," QualifiedName

ImportDecl = "import" QualifiedName ";"

MainClass = "class" Identifier "{" "public" "static" "void" "main" "(" "String" "[" "]" Identifier ")" "{" Statement "}" "}"

ClassDecl = "class" Identifier ("extends" QualifiedName)? ("implements" QualifiedNameList)? "{" (VarDecl ";" | MethodDecl)\* "}"

VarDecl = Type Identifier

MethodDecl = ("public"|"protected"|"private") Type Identifier "(" VarList? ")" "{" (Statement)\* "}"

VarList = VarDecl | VarList "," VarDecl

Type = "int"

| "boolean"

| "void"

| "double"

| "char"

| QualifiedName

| Type "[" "]"

| "(" VarList ")" % 元组类型，表示一个有类型的，不可更改的n元组对象

;

Statement = "{" (Statement)\* "}"

| "return" Expression? ";"

| "if" "(" Expression ")" Statement ("else" Statement)?

| "while" "(" Expression ")" Statement

| VarDecl ";"

| LValueExp "=" Expression ";"

;

LValueExp = Identifier

| Expression "." Identifier

| Expression "[" Expression "]"

;

Expression = Expression ( "&&" | "<" | "+" | "-" | "\*" ) Expression

| LValueExp

| Expression "." Identifier "(" (Expression ("," Expression)\*)? ")"

| IntegerLiteral

| DoubleLiteral

| "true" | "false"

| "null"

| ""

| "\"" ([^\\\"]|\\[.])\* "\"" % 字符串，中间可包括任意字符和转移字符

| "'" ([^\\\"]|\\[.]) "'"

| Identifier

| "this"

| "super" % super不能单独使用

| "new" Type ("[" Expression "]")?

| "new" QualifiedName "(" ")"

| "!" Expression

| "(" Expression ")"

| "(" Identifier ':' Expression ("," Identifier ':' Expression)+ ")" % n元组实例，如(a:1,b:2)

;

DoubleLiteral = IntegerLiteral "." IntegerLiteral? | IntegerLiteral? "." IntegerLiteral

Identifier is one or more letters, digits, and underscores, starting with a letter

IntegerLiteral is one or more decimal digits

EOF is a distinguished token returned by the scanner at end-of-file

# 3. 赛事在线管理系统

## 3.1 难度系数

|  |  |  |
| --- | --- | --- |
| 项目 | 基本等级 | 最高等级 |
| 工程量 | ☆☆☆ | ☆☆☆☆ |
| 算法挑战 | ☆ | ☆☆ |
| 学习成本 | ☆ | ☆☆ |
| 难度系数 | 0.7 | 0.95 |

## 3.2 需求定义

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 需求描述 | 基本要求 | 进阶要求 | 最高要求 | 分数 |
| 1 | 用户作为赛事组织者可以通过该系统提出赛事申请 | 在线填写赛事申请表，包括赛事名称、比赛类型（公开比赛——任何人都可申请、私有比赛——只有邀请的人可以参赛）、赛制（例如，但不限于，联赛、杯赛等，应支持至少四种真实存在的类型）、比赛时间（段）、赛事说明、是否为团体赛等 |  | 允许申请者配置多阶段比赛，每个阶段可以有不同的赛制，从而实现更加复杂的比赛计划。 | 8/8/12 |
| 2 | 赛事组织者可以通过该系统组织比赛 | 根据赛事阶段和赛制，组织者可以进行赛事组织工作，包括安排赛程和对阵表、填写比赛结果、管理参赛队员（伍），还可以添加其他用户作为赛事管理人员 | 系统应当以有好的形式展示和编辑赛事组织方式。系统可以自动生成报名二维码或报名链接 | 系统能够自动生成赛程和对阵表，并允许组织者随时修改和维护为进行的赛程和对阵计划 | 20/22/25 |
| 3 | 组织者可以通过该系统进行赛事网站管理 | 赛事申请一旦通过审批，系统应当为该赛事建立一个赛事网站，其中展示了赛事的信息，方便其他组织者和参与者查看。基本赛事信息包括申请表中的信息，赛事组织和管理者的信息。 | 网站能够实时展示赛事信息还包括赛程和赛果，参赛队伍列表等。如果是私有比赛，那么只有参赛人员才能查看赛事的相关信息 | 该网站也是所有用户访问该赛事并进行业务操作的界面。 | 5/8/15 |
| 4 | 组织者可以进行赛事生命周期管理 | 赛事生命周期包括准备、报名、比赛（可能又分为不同子阶段）、完赛。在不同的状态下，赛事组织者和参与者可执行不同的操作 |  |  | 5/5/5 |
| 5 | 用户可以通过该系统查看比赛列表 | 系统应列出所有比赛，按时间倒序排序，并支持赛事检索 |  | 系统可以按照不同的筛选条件显示赛事列表，例如按照时间、比赛状态等条件 | 6/6/7 |
| 6 | 用户可以通过该系统申请参赛 | 用户可以通过赛事网站、报名二维码/链接申请报名，报名时应填写报名表，包括姓名/队名（取决于比赛设定）、参赛队员（如果是团体赛）、联系方式，及其他必要信息。注意，如果报名人数超过比赛所需人数，那么赛事组织者应当能够选择实际参赛人（在需求2中完成） |  | 系统能够通过短信/邮件/微信的形式通知报名结果（及相应的变化） | 8/8/10 |
| 7 | 参赛队员可以通过该系统获得比赛信息 | （已被需求3覆盖） |  | 系统能能够通过短信/邮件/微信的形式提醒用户即将发生的比赛和刚刚结束的赛果 | 0/0/2 |
| 8 | 参赛队员可以通过该系统申请退赛 | 可以填写退赛申请 |  |  | 4/4/4 |
| 9 | 组织者可以通过该系统审批退赛申请 | 可以审批退赛申请并通知申请人员 |  |  | 4/4/4 |
| 10 | 系统管理员可以通过该系统审批赛事申请 | 系统管理可以批准或驳回赛事申请。驳回申请需要填写驳回理由 |  | 被驳回的申请应当可以被申请者修改后再次提交，重新提交的申请应当保留之前的驳回记录 | 5/5/10 |
| 11 | 系统管理员可以管理任意赛事的声明周期 | 系统管理员能够执行需求4 |  |  | 5/5/5 |

# 4. 评价指标

**说明：下面的评价指标对应课程的六个教学目标**

* 软件系统（40%）：需求达成度+完成质量，具体评分点见前三节
  + 代码质量将占有分数
* 开发环境（10%）：使用现代开发环境和工具开发，包括但不限于
  + 能够熟练使用git系统进行代码管理，包括分支、提交、pull request等
  + 能够使用wiki、bug issue等功能管理项目开发
  + 能够使用代码分析、自动化编译和测试功能改进开发
  + 使用Docker等工具规范开发环境
* 团队合作（10%）：有合理分工并且能够相互协作
  + 分工明确性
  + 协作程度（各队员尽量均衡贡献或取长补短）
* 文档和报告（30%）：最终文档和演示的质量
* 过程管理（10%）：有合理的进度安排并能充分利用时间、把握进度
  + 有开发计划和明确的里程碑（里程碑应对应需求）
  + 开发应充分利用时间
  + 能够基本按照计划完成开发

# 5. 华为云使用要求

1. 班长加入企业（需企业账号和充值），并获得管理员权限，将成员加入企业。
2. 每个小组建立一个项目，并管理自己的成员。
3. 小组利用“工作项”界面进行分工、计划和进度管理。**教师将追踪小组的工作项。**
   1. 在“需求”分组中建立和追踪软件需求（见上）。
   2. 在“开发”分组中建立组内工作项。
4. 利用“代码托管”功能管理项目代码。**特别注意，教师将通过统计每个成员提交的代码次数、代码量和代码质量评价每个成员的参与度、贡献度、以及小组的团队合作程度、进度控制程度**。因此必须充分使用“代码托管”功能才能获得最恰当的评价。
5. 利用“代码检查”功能管理代码质量。**教师将利用代码检查功能评价项目质量（只检查master分支）**。
6. 利用“测试计划”、“测试设计”、“测试执行”管理软件的测试过程。**教师将根据小组实际使用的情况评价软件的质量和工具使用情况**。
7. 利用“Wiki”撰写项目的过程和开发文档。**教师将根据这部分内容评价过程文档**。
8. 越充分利用现代项目管理工具，将获得越高的得分。