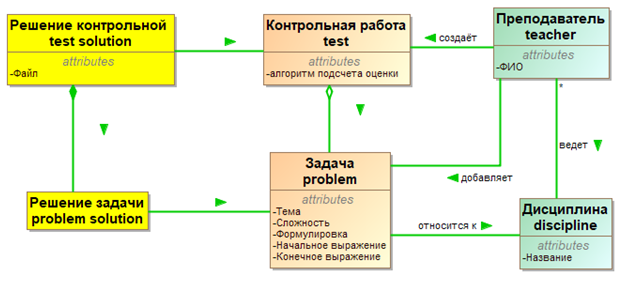
Система автоматизированного обучения прикладной математики

**Подсистема автоматической проверки правильности решения задач**

**Описание подсистемы**

Подсистема позволяет автоматически генерировать варианты контрольных работ и проверять их решения.

**Описание предметной области**



На данной диаграмме представлены сущности (задача и контрольная работа) и отношения, принадлежащие нашей части подсистемы, а также взаимосвязанные с ними части других подсистем.

Глоссарий:

|  |  |  |
| --- | --- | --- |
| **Термин** | | **Значение** |
| Задача | problem | Задача для студентов |
| Контрольная работа | test | Набор задач для студентов (в системе это список задач и соответствующих им баллов) |
| Решение задачи | problem solution | Текстовое изложение решения задачи по шагам |
| Решение контрольной работы | test solution | Файл с решениями задач |
| Преподаватель | teacher | Преподаватель |

* Контрольная работа состоит из задач и соответствующих им баллов.
* Решение контрольной работы состоит из решений задач.
* Контрольная работа создаётся преподавателем.
* Задача имеет тему, сложность, формулировку и относится к дисциплине, также для неё хранятся начальное и конечное выражения решения.
* Задача добавляется в базу преподавателем.

**Описание бизнес-процессов**

|  |  |  |  |
| --- | --- | --- | --- |
| Сценарий | Пояснение | Тип интерфейса | Приоритет |
| Добавление задачи | Преподаватель загружает задачу в базу данных. | API | 1 |
| Генерация контрольной работы | Преподаватель запрашивает создание контрольной работы по параметрам, указанным в разделе «Требования». Преподаватель указывает сколько баллов можно максимально получить за каждую задачу в контрольной. Сгенерированная контрольная работа сохраняется в базе - запоминаются номера выбранных системой задач, соответствующие им баллы. | API | 1 |
| Проверка решения контрольной работы | Подсистема Core запрашивает проверку решения контрольной работы. Осуществляется пошаговая автопроверка всех решений задач в контрольной работе. После чего подсистема АППРЗ выставляет оценку и отправляет её подсистеме Core. За любую контрольную работу минимальная оценка 2, максимальная 5. | API | 1 |
| Выдача формулировок задач | По запросу от подсистемы Core осуществляется выдача формулировок задач контрольной работы. | API | 2 |

**Действующие лица**

|  |  |
| --- | --- |
| Действующее лицо | Действия |
| Подсистема Core | * Запросить проверку решения контрольной работы. * Запросить выдачу формулировок задач контрольной работы |
| Преподаватель через подсистему GUI | * Запросить у системы генерацию контрольной работы. * Добавить задачу в базу. |

**Требования**

Общим предусловием для всех сценариев является существование базы данных для задач и сгенерированных контрольных работ (далее БД).

Текстовый файл с решением контрольной должен быть составлен студентом по правилам, решения задач должны быть пронумерованы в соответствии с нумерацией задач в контрольной.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сценарий | Доп. предусловие | Входные данные | Алгоритм | Выходные данные |
| Добавление задачи | - | Дисциплина, тема, сложность, формулировка задачи, начальное и конечное выражения. | Получение введенной преподавателем информации, добавление в базу данных соответствующей записи. | Булево значение: добавилась\не добавилась. |
| Генерация контрольной работы | - | * 1. Дисциплина, темы, количество задач каждой сложности.   2. Баллы для каждой задачи в сгенерированном наборе задач. | Генерация осуществляется в 2 этапа:   * 1. В базе имеется Ni задач сложности Zi (i=1,..k). Преподаватель задает темы задач, точное количество задач каждой доступной сложности (указывает, что будет Li задач сложности Zi).   2. Подсистема АППРЗ генерирует случайным образом Li из доступных Ni задач сложности Zi. Преподавателю выводится список задач контрольной работы.   3. Преподаватель назначает по своему усмотрению баллы каждой задачи из выведенного списка. Баллы задач указываются абстрактно, т.к. подсистема проведет автоматическое масштабирование.   4. Сгенерированной контрольной работе присваивается номер, в базу данных сохраняется список номеров задач и баллов. | Id контрольной работы в БД. |
| Проверка решения контрольной работы | Существование в БД контрольной работы с указанным номером | Текстовый файл с решением контрольной работы, номер контрольной работы. | Проверка решений задач по шагам, вердикт - верно/неверно - для решения каждой задачи  Оценка решений производится из следующих соображений дальнейшего оценивания:   * >=90% - оценка 5 * >=70% - оценка 4 * >=50% - оценка 3. | Оценка решения контрольной работы. |
| Выдача формулировок задач | Существование в БД контрольной работы с указанным номером | Номер контрольной работы | Поиск в базе списка задач, относящихся к данной контрольной работе, добавление их формулировок в массив. | Массив формулировок задач контрольной работы. |

**Программный интерфейс (API documentation)**

1. Добавление задачи

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Field** | **Description** | **Req** |
| long | disciplineId | id дисциплины | true |
| long | topicId | id темы | true |
| int | difficulty | Сложность задачи | true |
| string | statement | Формулировка задачи | true |
| string | startExpression | Начальное выражение решения | true |
| string | finalExpression | Конечное выражение решения | true |

|  |  |
| --- | --- |
| Title | addProblem |
| URL | http://localhost:7777//task/problems |
| Method | POST |
| Data Parameters | Example:  {  "disciplineId": 1,  "topicId": 1,  "difficulty ": 1,  "statement": "Given: 5x+5x^2=0. Find x",  "startExpression": "5x+5x^2=0",  "finalExpression": "x=0;x=-1"  } |
| Response | Without response body |

Генерация контрольной работы разбивается на 2 этапа.

1. Генерация набора задач по дисциплине, темам и указанному количеству задач каждого уровня сложности

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Field** | **Description** | **Req** |
| long | disciplineId | id дисциплины | true |
| long[] | topicIds | Список id тем | true |
| int[] | problemsByDifficulties | Cложности задач | True |
| int[] | amountByDifficulties | Количество задач по сложностям | True |
| long | testId | id контрольной работы | true |
| string[] | problemsDescriptions | Список формулировок задач | true |

|  |  |
| --- | --- |
| Title | generateProblemsSet |
| URL | http://localhost:7777/task/tests |
| Method | POST |
| Data Parameters | Example:  {  "disciplineId": 1,  "topicIds": [1],  "problemsDifficulties": [1, 3],  "amountByDifficulties": [1, 2]  } |
| Response | Example:  {  "problemsDescriptions" : [“Prove that B(m) equals S(n,0,m,S1(m,n)/P(n))”, "Given: 5x+5x^2=0. Find x"],  "testId": 0  } |

1. Генерация теста на основе набора задач по указанным баллам за каждую задачу

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Field** | **Description** | **Req** |
| int[] | scores | Список баллов | true |
| long | testId | id контрольной работы | true |

|  |  |
| --- | --- |
| Title | createTest |
| URL | http://localhost:7777/task/tests/8 |
| Method | PUT |
| Data Parameters | Example:  [ 10, 30, 30 ] |
| Response | Without response body |

Проверка решения контрольной работы:

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Field** | **Description** | **Req** |
| long | testId | id контрольной работы | true |
| string | testSolution | Решение задач контрольной работы | true |
| long | mark | Оценка | true |

|  |  |
| --- | --- |
| Title | evaluateTestSolution |
| URL | /task/marks/1 |
| Method | GET |
| Data Parameters | Example:  {  "testSolution" : " Problem №1 Solution: V(m,n)/A(k,k-m+1) = (m + n - 1)!/n!/(m-1)!/A(k,k-m+1) = (m + n - 1)!/n!/(m-1)!/k!\*(m-1)! = (m + n - 1)!/n!/k! = (m + n - 1)!/P(n)/k! = A(m+n-1,m+n-1-k)/P(n) Problem №2 Solution: 2 = 2 Problem №3 Solution: 3 = 3 Problem №4 Solution: V(m,n)/A(k,k-m+1) + 1 = (m + n - 1)!/n!/(m-1)!/A(k,k-m+1) + 1 = (m + n - 1)!/n!/(m-1)!/k!\*(m-1)! + 1 = (m + n - 1)!/n!/k! + 1 = (m + n - 1)!/P(n)/k! + 1 = A(m+n-1,m+n-1-k)/P(n) + 1"  } |
| Response | Example:  {  5  } |

Выдача формулировок задач:

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Field** | **Description** | **Req** |
| long | testId | id контрольной работы | true |
| string[] | problems | Список формулировок задач с указанием их локальных id | true |

|  |  |
| --- | --- |
| Title | getProblems |
| URL | http://localhost:7777/task/problems/8 |
| Method | GET |
| Data Parameters | Without request body |
| Response | Example:  {  "problems" : [  “Prove that A(m,n) equals A(m-1, n) + n\*A(m-1,n-1)”,  “Prove that F(n+3) – F(n-3) equals 2 \* F(n+1)”,  “Prove that B(m) equals S(n,0,m,S1(m,n)/P(n))”  ]  } |