|  |  |
| --- | --- |
| Министерство образования и науки Российской Федерации  Федеральное государственное бюджетное образовательное учреждение  высшего профессионального образования  «Ижевский государственный технический университет  имени М. Т. Калашникова» | |
| Кафедра «Программное обеспечение» | |
| Отчёт по лабораторной работе  дисциплины «КПО» | |
| Выполнил:  Студент группы Б08-191-2 | Р. С. Поскребышев |
| Принял: |  |
| Ижевск 2016 | |

Содержание

[Перечень рисунков 3](#_Toc438029265)

[1. ВВЕДЕНИЕ 5](#_Toc438029266)

[1.1. Назначение системы 5](#_Toc438029267)

[1.2. Область применения системы 5](#_Toc438029268)

[1.3. Определения, акронимы, аббревиатуры 5](#_Toc438029269)

[1.4. Обзор системы 5](#_Toc438029270)

[2. ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ 6](#_Toc438029271)

[2.1. Режимы и состояния системы 6](#_Toc438029272)

[2.2. Основные функциональные возможности системы 7](#_Toc438029273)

[2.3. Основные условия системы 7](#_Toc438029274)

[2.4. Основные ограничения системы: 7](#_Toc438029275)

[2.5. Характеристики пользователя 8](#_Toc438029276)

[2.6. Оперативные сценарии 8](#_Toc438029277)

[3. ИНТЕРФЕЙСЫ СИСТЕМЫ 9](#_Toc438029278)

[4. КЛАССЫ КОДИРОВАНИЯ 10](#_Toc438029279)

[4.1. Описание классов 10](#_Toc438029280)

[4.2. Соответствие подсистем и классов 11](#_Toc438029281)

[5. ИСХОДНЫЙ КОД 11](#_Toc438029282)

## Перечень рисунков

Рис 1. Диаграмма модулей

Рис 2. Диаграмма классов

# ВВЕДЕНИЕ

# Назначение системы

Система предназначена для управления задачами, управления доступом, тестирования решений.

# Область применения системы

Система ориентирована на учебные заведения (школы, университеты и др.) и иные учреждения, где есть необходимость автоматизированной проверки решений.

# Определения, акронимы, аббревиатуры

Тестирование, проверка, хранилище задач.

# Обзор системы

Система представляет собой комбинацию из двух основных приложений: программный интерфейс для работы с архивом, программный интерфейс для работы с тестирующей системой, и 2 графических интерфейса. Графический интерфейс содержит спецификацию (описание методов API и моделей). Для выполнения операций необходим токен, выданный CAA, который идентифицирует пользователя.

# ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ

# Режимы и состояния системы

Веб-интерфейс работает все время и не имеет состояний. API также не хранят состояния пользователя.

# Основные функциональные возможности системы



# Основные условия системы

Для работы системы необходим сервер Windows.

# Оперативные сценарии

Система не предполагает дополнительных сценариев, работает в автономном режиме.

# ИНТЕРФЕЙСЫ СИСТЕМЫ

IArchiveManager

public interface IArchiveManager

{

ICollection<Problem> GetCollection(long collectionId, string token);

string GetStatementUrl(long problemId, string token);

}

IArchiveWebSpec

public interface IArchiveWebSpec

{

}

IAuthManager

public interface IAuthManager

{

bool CheckToken(string token);

Token GetData(string token);

}

IBaseRepository

public interface IBaseRepository

{

object Get(long id);

object Add(object entry);

void Update(object entry);

void Delete(object entry);

}

IExecutorWebSpec

public interface IExecutorWebSpec

{

}

ILangRepository

public interface ILangRepository : IBaseRepository

{

}

ILangsController

public interface ILangsController

{

ILangRepository LangRepository { get; }

ITestingManager TestingManager { get; }

ICollection<Language> GetLangs();

}

IProblemRepository

public interface IProblemRepository : IBaseRepository

{

}

IProblemsController

public interface IProblemsController

{

IProblemRepository ProblemRepository { get; }

IAuthManager AuthManager { get; }

IArchiveManager ArchiveManager { get; }

ICollection<Problem> GetCollection(long collectionId, string token);

string GetStatementUrl(long problemId, string token);

}

IRabbitQueue

public interface IRabbitQueue

{

void Add(Submit submit);

SubmitResult GetSubmitResult(long submitId);

}

IResultRepository

public interface IResultRepository : IBaseRepository

{

}

IResultsController

public interface IResultsController

{

IResultRepository ResultRepository { get; }

IAuthManager AuthManager { get; }

ITestingManager TestingManager { get; }

ICollection<SubmitResult> GetSubmitResults(ICollection<long> ids, string token);

}

IRoleManager

public interface IRoleManager

{

void SetAccess(string methodId, string token);

}

ISubmitRepository

public interface ISubmitRepository : IBaseRepository

{

}

ISubmitsController

public interface ISubmitsController

{

IRabbitQueue RabbitQueue { get; }

ISubmitRepository SubmitRepository { get; }

IAuthManager AuthManager { get; }

ITestingManager TestingManager { get; }

ICollection<long> Send(ICollection<Submit> submits, string token);

void Rejudge(ICollection<long> submitIds, string token);

}

ITestingManager

public interface ITestingManager

{

ICollection<long> Send(ICollection<Submit> submits);

void Rejudge(ICollection<long> submitIds);

ICollection<SubmitResult> GetSubmitResults(ICollection<long> ids);

ICollection<Language> GetLangs();

}

Executor::IStartup

public interface IStartup

{

IExecutorWebSpec WebSpec { get; }

ILangsController LangController { get; }

IResultsController ResultsController { get; }

ISubmitsController SubmitsController { get; }

}

Archive::IStartup

public interface IStartup

{

IArchiveWebSpec WebSpec { get; }

IProblemsController ProblemsController { get; }

}

# КЛАССЫ КОДИРОВАНИЯ

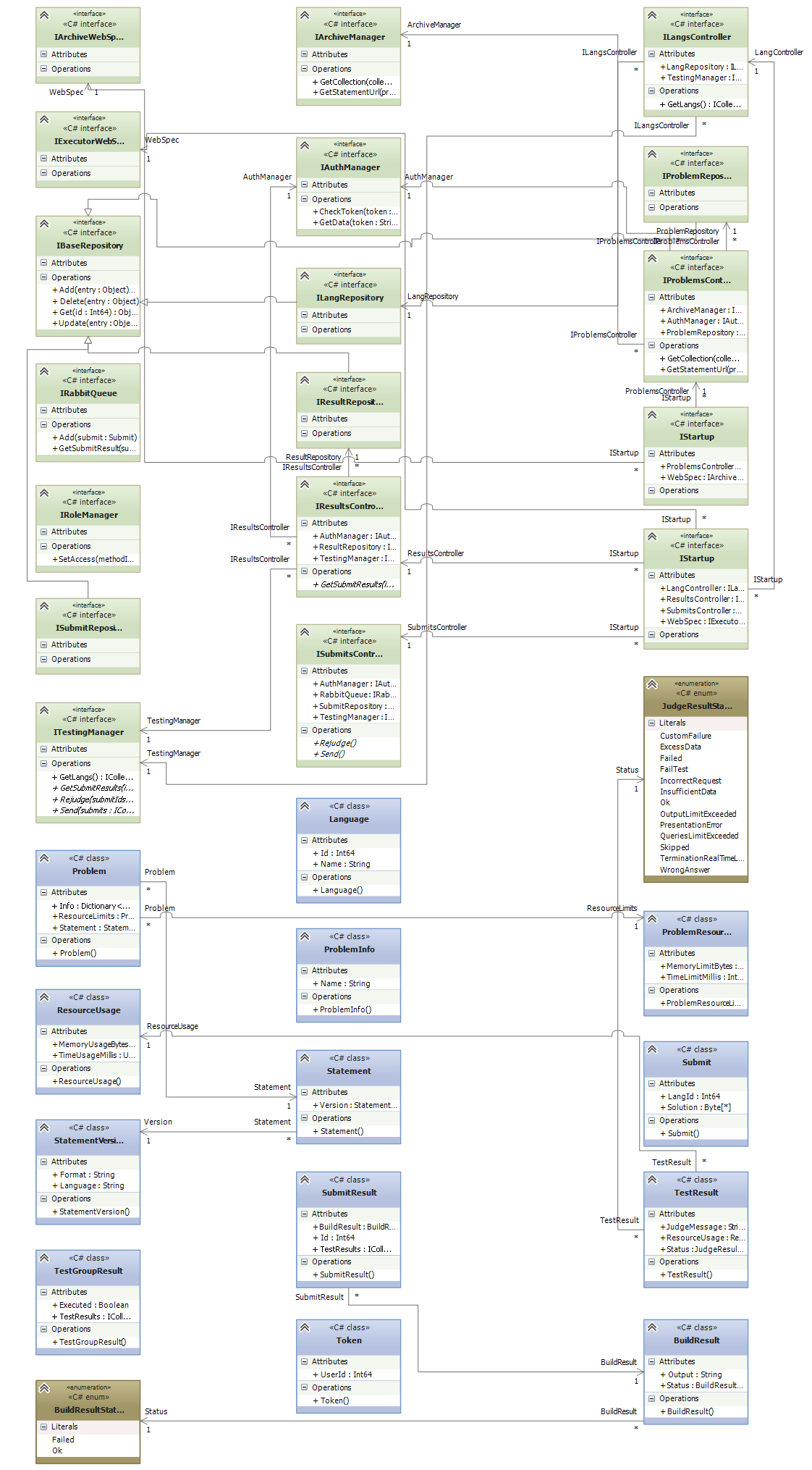


Рис 2. Диаграмма классов

# Описание классов

1. IArchiveManager - управляет архивом;

2. IArchiveWebSpec – представляет интерфейс спецификации архива;

3. IAuthManager – управляет авторизацией пользователей;

4. IBaseRepository – базовый репозиторий;

5. IExecutorWebSpec – представляет интерфейс спецификации проверяющей системы;

6. ILangRepository – репозиторий языков программирования;

7. ILangsController – контроллер языков программирования;

8. IProblemRepository – репозиторий задач;

9. IProblemsController – контроллер задач;

10. IRabbitQueue – очередь решений;

11. IResultRepository – репозиторий результатов решений;

12. IResultsController – контроллер результатов решений;

13. IRoleManager – управляет ролями;

14. ISubmitRepository – репозиторий решений;

15. ISubmitsController – контроллер решений;

16. ITestingManager – управляет тестирующей системой;

17. Language – язык программирования;

18. SubmitResult – результат решения;

19. BuildResult – результат построения решения;

20. TestGroupResult – результат тестирования на группе тестов;

21. TestResult – результат тестирования на одном тесте;

22. ResourceUsage – используемые ресурсы;

23. Problem – задача;

24. Statement – условие;

25. StatementVersion – версия условия;

26. ProblemInfo – информация о задаче;

27. ProblemResourceLimits – ограничение по решению задачи;

28. Submit – решение;

29. Archive::IStartup – запуск Archive API;

30. Executor::IStratup – запуск Executor API.

# Соответствие подсистем и классов

1. Archive API (Backend) – Archive::IStratup;
2. Archive API Spec (Frontend) - IArchiveWebSpec;
3. Executor API (Backend) – Executor::IStartup;
4. Executor API Spec (Frontend) - IExecutorWebSpec;
5. Подсистема репозиториев Archive – IProblemRespoditory, IBaseRepository;
6. Подсистема репозиториев Executor - ILangRepository, ISubmitRepository, IResultRepository, IBaseRepository;
7. Подсистема ролей – IRoleManager;
8. Очередь решений - IRabbitQueue;
9. Подсистема авторизации - IAuthManager;
10. Управление архивом – IArchiveManager;
11. Тестирование задач – ITestingManager.

# ИСХОДНЫЙ КОД

Исходный код системы располагается в репозитории по адресу https://github.com/poma-prs/CODIS