



Демонстрационный вариант задания заключительного этапа
Всероссийской олимпиады студентов «Я – профессионал»
по направлению «**Программная инженерия**»

Категория участия «Бакалавриат»

Задание 1. Подбор персонала (максимум 30 баллов).

Специалистам по подбору персонала необходима информационная система для автоматизации их работы.

В основе системы должна лежать реляционная база данных, содержащая информацию, необходимую для ввода и хранения следующих данных:

- Сведений о компаниях, включая название компании, название юридического лица, адрес юридического лица, контактные данные (телефон, e-mail), перечень вакансий компании с указанием названия вакансии, описания, требуемого уровня образования, перечня обязательных и желательных навыков (навыки являются справочной информацией), даты выставления вакансии, даты закрытия приема заявок на вакансию, диапазона заработной платы в месяц. Ответственный за ввод информации – пользователь с ролью «Менеджер».
- Сведений о соискателях работы, с указанием ФИО гражданина, телефонного номера, уровня образования, перечня навыков, желаемой заработной платы, перечня желаемых вакансий, текущей вакансии и текущего работодателя. Ответственный за ввод информации – пользователь с ролью «Гражданин».
- Сведений о процессе найма соискателя на работу компанией, включая даты собеседований, дату начала работы (если имеется), текущий статус («собеседование», «рассмотрение», «отказ», «оформление документов», «принят на работу»). Ответственный за ввод информации – пользователь с ролью «Специалист».
- Сведений об истории смены соискателем работодателей с указанием дат начала и окончания работы и наименование работодателя. Ответственный за ввод информации – пользователь с ролью «Гражданин».

Информационная система должна поддерживать следующий функционал для пользователей в зависимости от роли:

- Для пользователя с ролью «Менеджер» система предоставляет возможность ввода сведений о компаниях, ищущих сотрудников, с возможностью редактирования и удаления записи.
- Для пользователя с ролью «Гражданин» система предоставляет возможность ввода сведений о соискателях, а также о компаниях, в которых соискатель работал прежде, с возможностью редактирования и удаления записи.
- Для пользователя с ролью «Специалист» система отображает всех работодателей и соискателей и предлагает инструменты по подбору персонала и установлению соответствия между данными двумя категориями.



Необходимо:

- 1) Используя любую общепринятую нотацию, изобразить схему инфологической модели предметной области, удовлетворяющую третьей нормальной форме, с указанием первичных ключей и направления связей.
- 2) Используя любую общепринятую нотацию, изобразить схему даталогической модели базы данных, с выделением первичных и внешних ключей, типа и направления связей.
- 3) Изобразить прототипы web-страниц работы пользователей с ролями «Менеджер», «Гражданин» и «Специалист» с системой. Прототипы должны иллюстрировать возможности ввода и вывода доступной пользователям информации.
- 4) Описать алгоритм выбора кандидатов на конкретную должность.
- 5) Используя операторы языка SQL, написать запрос для вывода для конкретной компании списка из пяти наиболее популярных вакансий. Наиболее популярной считается вакансия, которую указывают как желаемую наибольшее число соискателей.

Требования к структуре оформления решения с указанием критериев оценивания и максимального количества баллов за каждую часть решения:

1. Введение - до 1 балла.
2. Основная часть:
 - 2.1. Схема решения, описание процессов, описание инфологической модели предметной области – до 5 баллов.
 - 2.2. Структура базы данных, отражающая специфику предметной области. БД должна соответствовать третьей нормальной форме, не быть перегруженной дублированием, учитывать сущности-справочники, не иметь ошибок по связям – до 6 баллов.
 - 2.3. Прототипы визуальных интерфейсов – до 6 баллов.
 - 2.4. Описание алгоритмов (в виде схемы алгоритмов, программного кода или псевдокода) – до 6 баллов.
 - 2.5. Описание SQL-запроса, демонстрация результатов выполнения запроса с примером подготовленных данных – до 5 баллов.
2. Заключение (выводы) – до 1 балла.



Задание 2. Тайный Санта (максимум 30 баллов).

Необходимо реализовать RESTFul API сервис для игры в Тайного Санту.

Типы данных:

Тип данных	Свойства
Группа	<p>id – идентификатор группы, натуральное число</p> <p>name – название группы, строка</p> <p>description – описание группы, строка</p> <p>participants – участники группы, список объектов типа “Участник”</p>
Участник	<p>id – идентификатор участника, натуральное число</p> <p>name – имя участника, строка</p> <p>wish – пожелание, строка</p> <p>recipient - подопечный, объект типа “Участник”</p>



Возможные действия:

Метод	Endpoint	Пояснение	Описание
POST	/group	<p>В запросе передается body в формате JSON:</p> <pre>{ "name": "string", "description": "string", }</pre> <p>В ответе ожидается идентификатор созданной группы, например 1</p>	<p>Добавление группы с возможностью указания названия (name), описания (description)</p> <p>Описание – не обязательный параметр, название – обязательный</p>
GET	/groups	<p>В ответе ожидается полный список групп, без указания участников в формате JSON:</p> <pre>[{ "id": number, "name": "string", "description": "string" }, ...]</pre>	<p>Получение краткой информации о всех группах (без информации об участниках)</p>



Метод	Endpoint	Пояснение	Описание
GET	/group/{id}	<p>ID группы, полную информацию о которой необходимо получить, передается в виде path-параметра, например: /group/1</p> <p>В ответе ожидается полная информация о группе, с указанием участников в формате JSON:</p> <pre>{ "id": number, "name": "string", "description": "string", "participants": [{ "id": number, "name": "string" "wish": "string", "recipient": { "id": number, "name": "string" "wish": "string" //Здесь не нужно передавать подопечного } }] }</pre>	<p>Получение полной информации (с информацией об участниках) о группе по идентификатору</p> <p>*До проведения жеребьевки <i>recipient</i> у участников не заполнен</p>



Метод	Endpoint	Пояснение	Описание
		<pre> }, ...] } </pre>	
PUT	/group/{id}	<p>ID редактируемой группы, передается как path-параметр, например: /group/1</p> <p>Редактируемые свойства передаются в body запроса в формате JSON:</p> <pre> { "name": "string", "description": "string" } </pre>	<p>Редактирование группы по идентификатору группы</p> <p>Редактировать можно только свойства name, description</p> <p>Удалить название таким образом нельзя, описание – можно</p>
DELETE	/group/{id}	<p>ID удаляемой группы передается как path-параметр, например: /group/1</p>	<p>Удаление группы по идентификатору</p>



Метод	Endpoint	Пояснение	Описание
POST	/group/{id}/participant	<p>ID группы, в которую добавляется участник, передается как path-параметр: например: /group/1/participant</p> <p>В запросе передается body с информацией о добавляемом участнике в формате JSON:</p> <pre>{ "name": "string", "wish": "string" }</pre> <p>В ответе ожидается идентификатор добавленного участника, например: 1223</p>	<p>Добавление участника в группу по идентификатору группы</p> <p>Пожелания – не обязательный параметр, имя – обязательный</p>
DELETE	/group/{groupId}/participant/{participantId}	<p>ID редактируемой группы и ID удаляемого участника передаются как path-параметры, например:</p> <p>/group/1/participant/1223</p>	Удаление участника из группы по идентификаторам группы и участника
POST	/group/{id}/toss	ID группы передается в виде path-параметра, например:	Проведение жеребьевки в группе по идентификатору группы



Метод	Endpoint	Пояснение	Описание
		<p>/group/1/toss</p> <p>В ответе ожидается список объектов типа “Участник” с указанными подопечными в формате JSON:</p> <pre>[{ "id": number, "name": "string", "wish": "string", "recipient": { "id": number, "name": "string", "wish": "string", //Здесь не нужно передавать подопечного } }, ...]</pre> <p>В случае, если проведение жеребьевки в данный момент невозможно – следует</p>	<p>Проведение жеребьевки возможно только в том случае, когда количество участников группы ≥ 3</p> <p>*Участнику в качестве подопечного нельзя выдать самого себя</p> <p>*Участник не может быть подопечным одновременно у двух и более участников</p>



Метод	Endpoint	Пояснение	Описание
		вернуть код ответа 409 (Conflict)	
GET	/group/{groupId}/participant/{participantId}/recipient	<p>ID группы и участника передаются в виде path-параметров, например:</p> <p>/group/1/recipient/2</p> <p>В ответе ожидается объект типа “Участник” – подопечный участника, чей идентификатор передан, в формате JSON:</p> <pre>{ "id": number, "name": "string" "wish": "string", }</pre>	Получение информации для конкретного участника группы, кому он дарит подарок.

Сервис должен запускаться на порту 8080.

При доступных и адекватных запросах код ответа должен быть равен 200, 201, 202 или 204.

При недоступных или неадекватных запросах сервис должен возвращать соответствующие сообщения об ошибках, код ответа должен отличаться от 200, 201, 202 или 204.

Можно использовать любые open source библиотеки.

Проверка будет производиться автоматизированным тестирующим ПО. Необходимо также приложить исходный код в виде архива и выгрузить его в любой git-репозиторий (github, gitlab, bitbucket) с предоставлением публичного всеобщего доступа и приложить ссылку.

Критерии оценки:



1. Проверка автоматизированным тестирующим ПО.
2. Описание технического решения
 - 2.1. Соответствие принципам SOLID
 - 2.2. Описание документации API в одной из общеиспользуемых спецификаций (рекомендуется OpenAPI (Swagger))

Требования к структуре оформления решения с указанием максимального количества баллов за каждую часть решения:

1. Исходный код решения.
 - 1.1. Оценка формируется исходя из результатов проверки API автотестом – до 12 баллов.
 - 1.2. Возможность запуска инфраструктуры приложения в контейнерах (docker-compose) – до 3 баллов.
2. Описание технического решения:
 - 2.1. Соответствие принципам SOLID – до 8 баллов.
 - 2.2. Описание документации API в одной из общеиспользуемых спецификаций (рекомендуется OpenAPI (Swagger)) – до 5 баллов.
 - 2.3. Описание настройки среды выполнения и запуска приложения - до 2 баллов.



Задание 3. Построение прогноза рейтинга телевизионного канала (максимум 40 баллов)

Задача состоит в построении прогноза рейтинга телевизионного канала с использованием признаков, найденных в открытых источниках. Необходимо выделить отдельные элементы программы (фильмы, ТВ-шоу, сериалы), найти по ним оценки, комментарии, список актеров. Также можно использовать дополнительные признаки (прогноз погоды, курс доллара, ...), которые можно получить автоматически из открытых источников в Интернет. Использовать найденные и построенные признаки для формирования прогнозной модели. Проверить полученный результат с данными Mediascope.

Этапы решения задачи:

1. Парсинг телепрограммы за текущий месяц (например, <https://tv.yandex.ru/>), формирование перечня программ
2. Парсинг открытых источников
 - a. <https://dbpedia.org/sparql>
 - b. IMDB
 - c. Youtube API
 - d. ... (список неограничен)
3. Формирование прогнозной модели рейтинга канала на месяц.
4. Сравнение с данными <https://mediascope.net/data/> . (Регион: Россия 100+, Неделя: текущая, Целевая аудитория: население 18-54, Отчет «100 наиболее популярных программ среди россиян в возрасте от 18 до 54 лет»)

Решение требуется оформить в виде **Jupyter**

Notebook <https://docs.jupyter.org/en/latest/start/index.html> .

Литература:

- https://www.researchgate.net/publication/349594007_Forecast_Model_of_TV_Show_Rating_Based_on_Convolutional_Neural_Network
- <https://developers.google.com/youtube/v3/quickstart/python>
- MAP@K метрика <https://medium.com/@misty.mok/how-mean-average-precision-at-k-map-k-can-be-more-useful-than-other-evaluation-metrics-6881e0ee21a9>

Требования к структуре оформления решения с указанием максимального количества баллов за каждую часть решения:

1. Введение (описание решения) – до 1 баллов.
2. Основная часть:
 - 2.1. Решение задачи парсинга телепрограммы – до 3 баллов.



- 2.2. Парсинг открытых источников– до 3 баллов.
- 2.3. Анализ методов машинного обучения и обоснование выбора для решения поставленной задачи. – до 2 баллов.
- 2.4. Описание реализации модели машинного обучения для решения задачи, а также обоснование выбора, используемого ПО и библиотек – до 5 баллов.
- 2.5. Программная реализация прогнозной модели рейтинга канала на месяц – до 9 баллов (приложить файл (ы) с решением)).
- 2.6. Демонстрация решения – до 10 баллов.
- 2.7. Результаты сравнения полученного решения с данными <https://mediascope.net/data/> и их анализ – до 7 баллов.