



Демонстрационный вариант
задания заключительного этапа
по направлению «Программная инженерия»

Категория участия: «Бакалавриат»
(для поступающих в магистратуру)

Задание 1. Анализ текста новостей (максимум 30 баллов)

Дан некоторый текст новости, необходимо выделить основных действующих лиц и организации, а также определить суть события, описываемого в новости.

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

1. Введение (изложить главную идею решения) – до 1 балла.
2. Основная часть:
 - 2.1. Алгоритм выделения активных субъектов (людей и организаций) в тексте, представленный в виде схемы алгоритма или словесного описания алгоритма – до 5 баллов.
 - 2.2. Алгоритм выделения сути событий – до 5 баллов.
 - 2.3. Выбор инструментов (языка, библиотек, моделей) для реализации алгоритмов – до 3 баллов.
 - 2.4. Программная реализация решения – до 8 баллов.
 - 2.5. Демонстрация решения на тестовом примере с использованием программной реализации – до 7 баллов.
3. Заключение (выводы) – до 1 балла.

Задание 2. Расчет оптического кабеля (максимум 30 баллов)

В файле хранится список адресов площадок узлов связи. Требуется рассчитать: сколько метров оптического кабеля потребуется, чтобы соединить каждую площадку с каждой. В качестве расстояния между двумя объектами следует использовать Манхэттенское расстояние. Для прокладки по зданию необходимо заложить удвоенную высоту объекта, где размещается узел связи. Высоту объектов можно определять разными способами (например, 2GIS). Высоту этажа считать равным 3 м.

Пример содержимого файла:

Екатеринбург, ул. Вайнера, д. 16
Екатеринбург, ул. Карла Либкнехта, д. 22
Екатеринбург, ул. Малышева, д. 53
Екатеринбург, ул. Малышева, д. 5
Екатеринбург, ул. Татищева, д. 69
Екатеринбург, ул. Техническая, д. 37



Критерии оценки:

Расстояние вычислено верно, нет дублей, учтен удвоенный запас высоты зданий на каждую площадку.

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

1. Введение (изложить главную идею решения) – до 1 балла.
2. Основная часть:
 - 2.1. Алгоритм для расчета метража кабеля – до 5 баллов.
 - 2.2. Схема инфологической модели предметной области – до 2 баллов.
 - 2.3. Прототипы визуальных интерфейсов – до 2 баллов.
 - 2.4. Описание архитектуры решения – до 2 баллов.
 - 2.5. Выбор инструментов (языка, библиотек, моделей) для реализации алгоритма – до 5 баллов.
 - 2.6. Программная реализация решения – до 7 баллов.
 - 2.7. Демонстрация решения на тестовом примере с использованием программной реализации – до 5 баллов.
3. Заключение (выводы) – до 1 балла.

Задание 3. Промоакции и розыгрыши призов (максимум 40 баллов)

Необходимо реализовать RESTFul API сервис для проведения промоакций и розыгрышей призов.

Типы данных:

Тип данных	Свойства
Промоакция	id – идентификатор промоакции, натуральное число name – название промоакции, строка description – описание промоакции, строка prizes – возможные призы в промоакции, список объектов типа “Приз” participants – участники промоакции, список объектов типа “Участник”
Приз	id – идентификатор приза, натуральное число description – описание приза, строка
Участник	id – идентификатор участника, натуральное число name – имя участника, строка
Результат проведения розыгрыша	winner – объект типа “Участник” prize – объект типа “Приз”



Возможные действия:

Метод	Endpoint	Пояснение	Описание
POST	/promo	В запросе передается body в формате JSON: <pre>{ "name": "string", "description": "string", }</pre> В ответе ожидается идентификатор созданной промоакции, например, 1	Добавление промоакции с возможностью указания названия (name), описания (description) Описание – не обязательный параметр, название – обязательный
GET	/promo	В ответе ожидается полный список промоакций, без указания участников и призов в формате JSON: <pre>[{ "id": number, "name": "string", "description": "string" }, ...]</pre>	Получение краткой информации (без информации об участниках и призах) обо всех промоакциях
GET	/promo/{id}	ID промоакции, полную информацию о которой необходимо получить, передается в виде path-параметра, например: /promo/1 В ответе ожидается полная информация о промоакции, с указанием участников и призов в формате JSON: <pre>{ "id": number, "name": "string", "description": "string", "prizes": [{ "id": number, "description": "string" }, ...], "participants": [{ "id": number, "name": "string" }, ...] }</pre>	Получение полной информации (с информацией об участниках и призах) о промоакции по идентификатору



		<pre>] }</pre>	
PUT	/promo/{id}	<p>ID редактируемой промоакции, передается как path-параметр, например: /promo/1</p> <p>Редактируемые свойства передаются в body запроса в формате JSON:</p> <pre>{ "name": "string", "description": "string" }</pre>	<p>Редактирование промоакции по идентификатору промоакции</p> <p>Редактировать можно только свойства name, description</p> <p>Удалить имя таким образом нельзя, описание – можно</p>
DELETE	/promo/{id}	<p>ID удаляемой промоакции передается как path-параметр, например: /promo/1</p>	<p>Удаление промоакции по идентификатору</p>
POST	/promo/{id}/participant	<p>ID промоакции, в которую добавляется участник, передается как path-параметр: например: /promo/1/participant</p> <p>В запросе передается body с информацией о добавляемом участнике в формате JSON:</p> <pre>{ "name": "string" }</pre> <p>В ответе ожидается идентификатор добавленного участника, например: 1337</p>	<p>Добавление участника в промоакцию по идентификатору промоакции</p>
DELETE	/promo/{promoId}/participant/{participantId}	<p>ID редактируемой промоакции и ID удаляемого участника передаются как path-параметры, например: /promo/1/participant/1337</p>	<p>Удаление участника из промоакции по идентификаторам промоакции и участника</p>



POST	/promo/{id}/prize	<p>ID промоакции, в которую добавляется приз, передается как path-параметр: например: /promo/1/prize</p> <p>В запросе передается body с информацией о добавляемом призе в формате JSON:</p> <pre>{ "description": "string" }</pre> <p>В ответе ожидается идентификатор добавленного приза, например: 321</p>	Добавление приза в промоакцию по идентификатору промоакции
DELETE	/promo/{promoId}/prize/{prizeId}	<p>ID редактируемой промоакции и ID удаляемого приза передаются как path-параметры, например: /promo/1/participant/1337</p>	Удаление приза из промоакции по идентификаторам промоакции и приза
POST	/promo/{id}/raffle	<p>ID промоакции передается в виде path-параметра, например: /promo/1/raffle</p> <p>В ответе ожидается список объектов типа “Результат проведения розыгрыша” в формате JSON:</p> <pre>[{ "winner": { "id": number, "name": "string" }, "prize": { "id": number, "description": "string" } }, ...]</pre> <p>В случае, если проведение розыгрыша в данный момент невозможно – следует вернуть код ответа 409 (Conflict)</p>	<p>Проведение розыгрыша призов в промоакции по идентификатору промоакции</p> <p>Проведение розыгрыша возможно только в том случае, когда количество участников и призов в промоакции совпадает (т.е., например, если в промоакции в текущий момент 2 участника и 2 приза или 3 участника и 3 приза и т.д.)</p>



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. Результаты проверки автотестом – до 24 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 8 баллов.



Альтернативный вариант заданий 1 или 2

Ведение эталонных реестров данных (максимум 30 баллов)

Необходимо разработать автоматизированную систему ведения эталонных реестров данных городской администрации.

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

- Таблиц, содержащих эталонные реестры данных. Структура таблиц заранее не определена. Ответственный за создание таблиц и ввод информации – пользователь с ролью «Сотрудник» и соответствующими правами доступа к конкретной таблице.
- Сведений о подразделениях городской администрации, отделах подразделений и сотрудниках отделов, включая их паспортные данные, пол, должность, дату приема на работу, контактную информацию (рабочий телефон, корпоративный почтовый адрес, часы приема если есть). Ответственный за ввод информации – пользователь с ролью «Администратор».
- Сведений об уровне доступа каждого сотрудника, отдела в целом и подразделения в целом к каждой таблице, содержащей эталонные реестры данных, а также дате наделения сотрудника таким уровнем доступа и дате прекращения такого уровня доступа. Уровень доступа может быть: 1) создание таблицы, 2) удаление таблицы, 3) просмотр данных таблицы (всех полей или определенного перечня полей), 4) добавление данных в таблицу, 5) редактирование (изменение, добавление и удаление) данных в таблице. Ответственный за ввод информации – пользователь с ролью «Администратор».

Для пользователя «Сотрудник» информационная система должна поддерживать ввод, редактирование и просмотр хранимой информации согласно разграниченному доступу. При просмотре таблицы должна быть возможность сортировки по полям и фильтрации по значениям полей, а также именования каждого доступного для просмотра поля. Система должна позволять экспортировать таблицы из форматов .xls, .xlsx, .csv, а также импортировать данные выбранной таблицы в форматах .xls, .xlsx, .csv.

Для пользователя «Администратор» система должна предоставлять функции ввода новых пользователей в систему и удаления из системы, изменения настроек доступа к таблицам и полям.

Необходимо:

- 1) Используя любую общепринятую нотацию, изобразить схему инфологической модели предметной области.
- 2) Используя любую общепринятую нотацию, изобразить схему даталогической модели базы данных, удовлетворяющую третьей нормальной форме, с выделением первичных и внешних ключей, типа и направления связей.
- 3) Изобразить прототипы web-страниц работы пользователей с ролями «Администратор» и «Сотрудник» с системой. Прототипы должны иллюстрировать возможности ввода и вывода доступной пользователям информации.



- 4) Описать алгоритм расчета процента сотрудников заданного подразделения и заданного отдела, имеющих право редактирования данных в указанной таблице.
- 5) Используя операторы языка SQL, написать запрос для вывода по каждому подразделению количества сотрудников с правом создания таблиц, действующим в последние полгода.

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

1. Введение – до 2 баллов.
2. Основная часть:
 - 2.1. Схема решения, описание процессов, описание инфологической модели предметной области – до 3 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 5 баллов.
 - 2.3. Прототипы визуальных интерфейсов – до 6 баллов.
 - 2.4. Описание алгоритма (в виде схемы алгоритма, программного кода или псевдокода) – до 6 баллов.
 - 2.5. Описание SQL-запроса – до 6 баллов.
3. Заключение (выводы) – до 2 баллов.



Требования к аппаратному и программному обеспечению персонального компьютера, используемого для выполнения заданий заключительного этапа Всероссийской олимпиады «Я – профессионал» по направлению «Программная инженерия»

Системные требования

Операционная система: Windows 10 (Допустимо Windows 7, Ubuntu 14 и выше)
Оперативная память: 8 ГБ (минимум 4 ГБ)

Требования к программному обеспечению

Общие требования:

Офисное приложение MS Office Word 2007 или выше
OpenOffice (<https://www.openoffice.org/ru/>) LibreOffice (<https://ru.libreoffice.org/>).
Браузер (Google Chrome, Yandex, Opera, Microsoft Edge, Mozilla Firefox, Яндекс.Браузер) с включенной технологией JavaScript и cookie и с доступом к внешним ресурсам.

Модели и алгоритмы:

Рекомендуется: MS Visio 2010
Допустимо: Браузер с доступом к внешним ресурсам (<https://app.diagrams.net/>, <https://draw.io>)

Базы данных:

Рекомендуется: MySQL (<https://www.mysql.com/downloads/>) версия 5.1 и старше
PostgreSQL (<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>) версия 8 и старше
dbeaver (<https://dbeaver.io/download/>).
Допустимо: Microsoft SQL Server (<https://www.microsoft.com/ru-ru/sql-server/sql-server-downloads>), HeidiSQL (<https://www.heidisql.com/download.php>), JetBrains DataGrip (<https://www.jetbrains.com/ru-ru/datagrip/download>)

Программный код:

- 1) MS Visual Studio 2019 (<https://visualstudio.microsoft.com/ru/downloads/>) допустимы версии с VS2012
- 2) Anaconda + Jupyter Notebook (<https://www.anaconda.com/products/individual>) рекомендуется версия под python 3.8. Допустимы версии от 2.5. (альтернативно может быть установлен интерпретатор python).
- 3) OpenServer (<https://ospanel.io/>) включает в себя MySQL, MariaDB, PostgreSQL) рекомендуемая версия 5.3.7, или XAMP (<https://www.apachefriends.org/ru/index.html>) или Denwer (<http://www.denwer.ru/>)
- 4) IntelliJ IDE A Community (<https://www.jetbrains.com/ru-ru/idea/download/#section=windows>) – любая версия (опционально). Обязательно наличие любой актуальной версии java.
- 5) NodeJS и npm (<https://nodejs.org/en/>) рекомендуемая версия 12.19.0 и более старые. Использование более новых версий, включая 14.13.1 также допустимо.



6) VisualStudioCode(<https://code.visualstudio.com/>) или SublimeText 3 (<https://www.sublimetext.com/3>) любые версии.

UI дизайн:

Рекомендуется: Браузер (GoogleChrome, Yandex, Opera, MicrosoftEdge, MozillaFirefox) с доступом к внешним ресурсам (<https://www.figma.com/>)
Допустимо: CorelDraw (<https://www.coreldraw.com/ru/free-trials/?topNav=ru>)

UI дизайн, работа с видео:

Рекомендуется: онлайн-ресурсы, в том числе figma, Windows Movie Maker (<https://www.microsoft.com/ru-ru/p/movie-maker-10-free/9mvfq4lmz6c9?activetab=pivot:overviewtab>), VSDC Free Video Editor (<http://www.videosoftdev.com/free-video-editor/download>), Adobe After Effects Trial (<https://www.adobe.com/ru/products/aftereffects.html>), Adobe Premier Trial (<https://www.adobe.com/ru/products/premiere/free-trial-download.html>).
Допустимо: прочие видеоредакторы.