|  |  |  |
| --- | --- | --- |
|  |  | |
|  |
|
|

Конкурсное задание JS 10+

по компетенции:

**Интернет вещей (Умный город)**

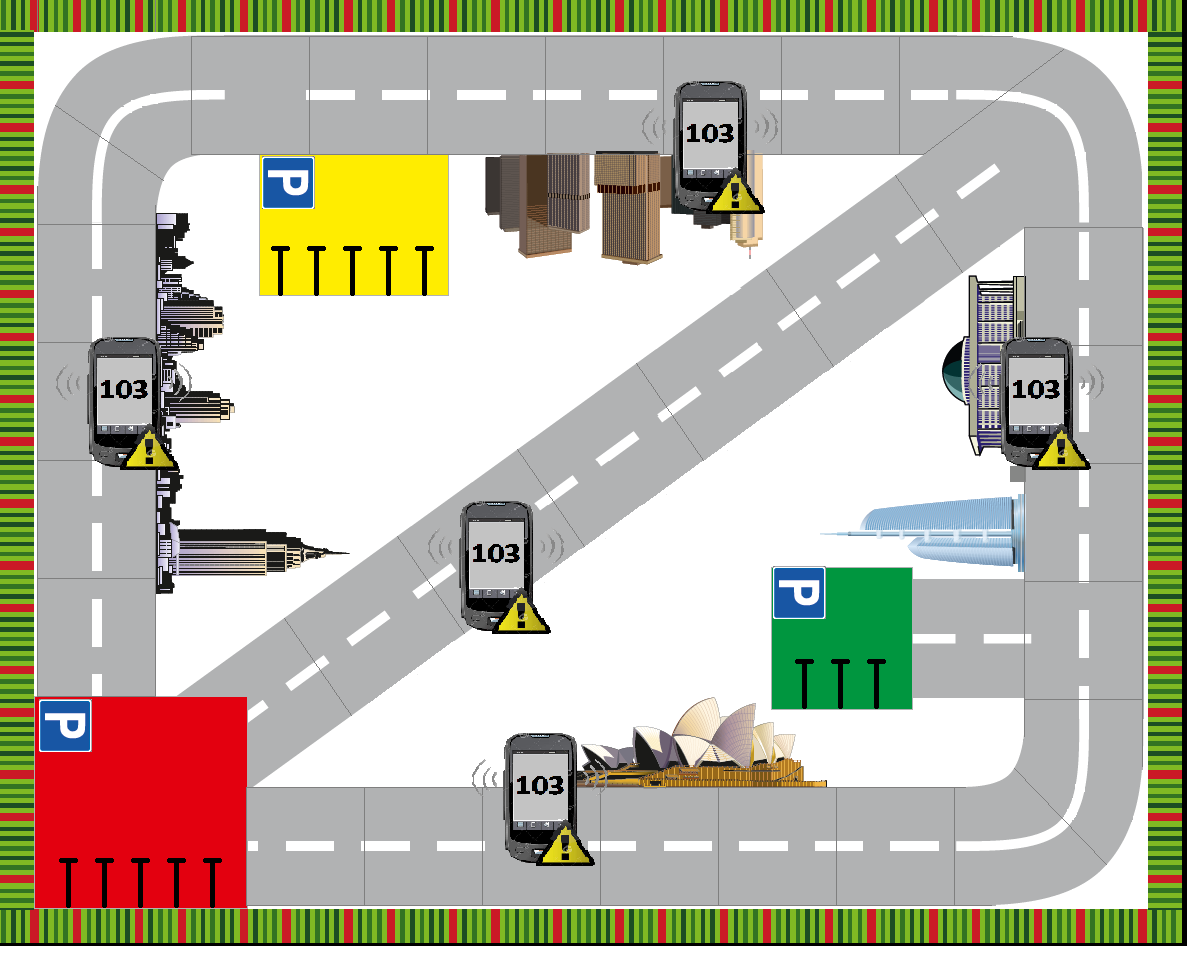
**Тема:**

Диспетчерская служба "Скорой помощи" (мониторинг обслуживания вызовов и выдача рекомендаций диспетчеру)

**Общие исходные данные для всех заданий**

Дан план города с дорожной сетью и местами, отведенными для парковки машин "Скорой помощи" у подстанций. На плане города отмечены места, откуда может быть произведен вызов. В месте вызова возможна остановка, но не парковка, поэтому после вызова машина должна вернуться на одну из парковок.

Примерный план города



Движение в городе одностороннее, допускаются только повороты направо, повороты налево и развороты запрещены.

Парковка осуществляется на отведенных местах, машины занимают последовательные места справа налево; пропуск парковочных мест не допускается; на вызов выезжает крайняя слева машина.

Число машин во время выполнения задания остается неизменным.

Парковки оснащены датчиками, передающими в диспетчерский центр информацию о количестве машин на парковке и количестве свободных мест.

Дорога маркирована на зоны, условно обозначающих 1км пути; таким образом по числу пройденных зон можно определить пройденное расстояние, время в пути (1 мин на км) и расход топлива (10л на км). Для отдельных зон могут действовать повышающие или понижающие коэффициенты.

Участникам соревнований предлагается выполнить три последовательных задания, каждое из которых включает в себя период подготовки к выполнению задания и выполнение самого задания в регламентированные сроки.

Участникам предоставляется преднастроенная платформа по Интернету вещей с созданными в ней объектами, атрибутами и взаимосвязями, а также методические рекомендации для настройки необходимых веб страниц.

**Задание 1.**

**Обеспечение мониторинга состояния парковок.**

Время выполнения задания – 3 минуты

Время подготовки к заданию – 4 часа

**Описание задачи**

Для всех подстанции вывести количество машин на площадках для парковки. Проверить правильность парковки машин; при неправильной парковке указать номера соответствующих подстанций.

**Порядок подготовки к заданию**

1. В Composer ThingWorx создать mashup для каждой подстанции.
2. На каждый mashup добавить дашборд, связанный со значением параметра «Количество машин на подстанции»
3. На каждом mashup создать схему парковки с индикаторами, цвет индикатора установить зеленым, если показание с датчика парковочного места «Место свободно», цвет индикатора установить красным, если показание датчика парковочного места «Место занято»
4. Создать объекта «Правильность парковки» с параметром, который будет принемать значения «Правильно» или «Не правильно» по придуманному алгоритму. Придумать и реализовать данный алгоритм
5. На каждом mashup добавить индикатор, и связать его со значением параметра для объекта «Правильность парковки».
6. На каждый mashup добавить список, который будет выводить название парковок, в которых допущена ошибка. При нажатии на парковку из данного списка должен открываться mashup соответсвующей парковки.
7. Запустить просмотр mashup в действии для проверки правильности отображения данных.

**Порядок выполнения задания**

Судья расставляет машины на макете города и объявляет старт задания.

На выполнение задания отводится 3 мин.

В течение указанного времени командам следует вывести на веб-страницу своего приложения указанную в задании информацию.

**Задание считается завершенным когда:**  
1. На экране компьютера в режиме реального времени отображается правильное количество машин для каждой паркови

2. На экране компьютера в режиме реального времени выводится информация о неправильной парковке с указанием соответсвующей подстанции.

**Задание 2.**

**Обеспечение мониторинга движения машины**

Время выполнения задания –3 минуты

Время подготовки к заданию – 4 часа

**Описание задачи**

При выезде машины на вызов в указанный пункт и последующим прибытии к подстанции, указать для данной машины:

- пройденный путь а) до места вызова, б) от места вызова до подстанции, в) общее расстояние, пройденное для работы по вызову

- затраченное время а) прибытия по вызову, б) движение от места вызова к подстанции, в) общее время за один вызов;

- расход бензина за вызов;

**Порядок подготовки к заданию**

1. В Composer ThingWorx для объектов Маршрут установить правильные значения для параметра Расстояние. Установить взаимосвязь между параметрами Расстояние, Время и Расход бензина, используя необходимые формулы рассчета.
2. В Composer ThingWorx создать mashup для отображения показателей движения машины по вызову.
3. Добавить на mashup список, и установить связь с объектами Машина, Начальная Парковка, Вызов, Конечная парковка, Маршрут к вызову, Маршрут на парковку.
4. Установить взаимосвязь между выбранными значениями в списке со значениями параметров машины.
5. Добавить кнопку, которая будет устанавливать соответственные значения параметрам, на основе выбранных объектов в списках.
6. Добавить на mashup текстовые поля, которые будут выводить название показателей, необходымых по заданию.
7. Добавить на mashup поля и связать их со значениями соответсвующих параметров машины, которые будут выводить значения соответсвующих параметров на экран.
8. Запустить просмотр mashup в действии для проверки правильности отображения данных.

**Порядок выполнения задания**

Судья объявляет начало выполнения задания и дает время (1 мин), чтобы команды зафиксировали текущее положение машин на плане (макете).

По завершении отведенного времени, судья переставляет одну из машин на плане (макете) города со стоянки одной из подстанций, мимо пукта вызова, на стоянку другой подстанции.

Командам дается 2 минуты, чтобы ввести информацию в программу о маршруте, по которому прошла машина и вывести на веб-страницу своего приложения указанную в задании информацию.

**Задание считается завершенным когда:**  
1. На экране компьютера в режиме реального времени отображается данные по данной машине скорой помощи, а именно:

Пройденное расстояние (км)

Расстояние от подстанции до пункта вызова

Расстояние от пункта вызова до подстанции

Общее расстояние за вызов

Затраченное время (мин)

Прибытие по вызову

Движение от места вызова к подстанции

Общее время

Расход бензина (л)

Расход за вызов

**Задание 3.**

**Обеспечение поддержки диспетчера**

Время выполнения задания – 3 минуты

Время подготовки к заданию – 4 часа

**Описание задачи**

При поступлении вызова из данного пункта, указать диспетчеру номер подстанции, с которой должна выехать машина, и на какую подстанции вернуться, используя критерии:

- минимальное время прибытия по вызову;

- минимальное время возвращения на подстанцию;

- минимальный расход бензина.

Указать также следующую возможную альтернативу выбора подстанции и маршрута движения.

При отображении маршрута должны указываться следующие показатели:

Время прибытия на вызов;

Время прибытия обратно на подстанцию;

Расход бензина.

Выбор машины должен осуществляться в соответсвии с текущем наличием машин на подстанциях.

**Порядок подготовки к заданию**

1. В Composer ThingWorx создать шаблон объекта маршрутов, с атрибутами «Время прибытия на вызов», «Время прибытия на подстанцию», «Расход бензина», «Название подстанции выезда на вызов» «Название подстанции прибытия после вызова»
2. Создать все возможные объекты на основе шаблона «Маршрут» и назначить значения параметров.
3. Создать список и связать его с объектом «Критерий выбора маршрута».
4. Создать поле для вывода информации о стартовой подстанции, установить правило, чтобы выводилось минимальное значение по выбранному критерию.
5. Создать поле для вывода информации о финишной подстанции, установить правило, чтобы выводилось минимальное значение по выбранному критерию.
6. Запустить просмотр mashup в действии для проверки правильности отображения данных.

**Порядок выполнения задания**

Судья объявляет начало выполнения задания и дает время (1 мин), чтобы команды зафиксировали текущее положение машин на плане (макете).

По завершении отведенного времени, судья объявляет участникам пункт вызова.

Командам дается 2 минуты, чтобы вывести на веб-страницу своего приложения указанную в задании информацию.

**Задание считается завершенным когда:**  
1. На экране компьютера в режиме реального времени при поступлении вызова отображается рекомендация с какой постанции отправить машину скорой помощи на вызов, и на какую постанцию данной машине необходимо вернуться, учитывая наличие машин на подстанциях.

2. На экране отображаются данные о маршруте (время прибытия на вызов, время возвращения на подтанцию, расход бензина)

3. Есть возможность выбрать по какому критерию подбирается маршрут (минимальное время, минимальное время возвращения, минимальный расход бензина). *Оценивается оптимальность выбранного пути по критериям.*

4. На экране должно отображаться наличие машин на подстанциях.