Создание системы поддержки принятия решений в зданиях ОУ при возникновении ЧС.

**Задачи.**

**Введение**

В комплексе вопросов, связанных с обеспечение безопасности образовательных учреждений (ОУ) – особое место занимают технические аспекты обеспечения безопасности при пожарах и проявлениях террористической активности.

Обычно фаза ЧС, требующая безотлагательных действий, связанных с пожаром и (или) террористической активностью, развивается на временном горизонте нескольких минут. Учитывая, что источник опасности может изменяется на временном горизонте нескольких секунд, СППР должна обновлять свои рекомендации для людей в здании приблизительно один раз в секунду. То есть в интервале нескольких минут, когда ситуация на объекте защиты постоянно изменяется, система поддержки принятия решений должна каждую секунду получать и обрабатывать информацию о состоянии людей в здании, о развитии источника опасности. На основании полученной информации СППР должна с дискретностью порядка секунды формировать предложения по рекомендуемым действиям людям на объекте защиты.

**Построение BIM –модели здания ОУ**

**Построение цифрового двойника здания ОУ для СППР.**

Цифровой двойник представляется в виде Json-файла, который используется для десериализации

В нотации классов цифровой двойник -

public class BuildingPIM {  
 public String nameBuilding; *// Общее название здания* public String program\_name; *// Программа создания файла Json* public String version\_program; */\* Версия (дата последней модификации программы создания файла Json) \*/* public String date\_creation\_Json; *// Дата создания файла Json* public Address\_Building address\_building; *// Адресные данные* public Level\_Building[] Level; *// Класс уровней здания  
  
 // --------------------------------------------------  
  
 // \* Внутренний класс для указания адреса - Address.* public class Address\_Building {  
 public String city; *// Город.* public String streetAddress; *// Улица, дом.* public String addInfo; *// Допол. инфо* }  
  
 *// \* Внутренний класс для указания уровней здания - Level\_Bilding.* public class Level\_Building {  
 public String NameLevel; *// Имя уровня\* public double ZLevel; *//Положение уровня по высоте, мм* public Element\_Json[] BuildElement;  
  
 *// \* Внутренний класс для указания элемента Json - Element\_Json.* public class Element\_Json {  
 public String Name; *// Имя элемента* public String Id; *// Id элемента* public String Sign; *// Код элемента* public double SizeZ; *// Размер по высоте элемента, мм* public double Wide; *// Размер по ширине элемента, мм* public double Sroom; *// Площадь элемента, м\*м* public String[] Output; *// Список указателей на элементы* }  
 }

Sign ::= { Room, Staircase, Outside, Safety zone, DoorWayOut, DoorWayInt, DoorWay} /\* Код элемента

Room - помещение

Staircase - лестница

Outside - нерасчетная область

Safety zone - область зоны безопасности

DoorWayOut - дверь внешняя

DoorWayInt - дверь внутренняя

DoorWay - проем на пути движения внутри здания \*/

Проверки Json

1. Внутренних дверей и внешних дверей
2. Помещений (по количеству дверей

**Характеристики элемента здания ОУ при моделировании управляемого движения людей в условиях ЧС.**

public class Element { // 26.04.2022  
 public String Id; // Уникальный Идентификатор элемента  
 public double dPeople; // Количество людей (текущее) в элементе,

// человек

public String NameLevel; // Название уровня в здании  
 public double ZLevel; // Значение уровня (этажа) помещения, м  
 public double SizeZ; // Высота элемента (Определено для Sign = Staircase ), м  
 public double Wide; // Ширина портала (элемента) (Sign = { DoorWayOut, DoorWayInt, DoorWay}), м // DoorWay – Дверной проем  
 public String Name; // Название элемента в соответствии с планом  
 public String Sign; // Код элемента Sign = { Room, Staircase, Outside, Safety zone} [ Постоянные коды ]  
 public double probability; // Вероятность 0-1 прохождения элемента Room,

Staircase, [Изменяется в процессе моделирования]  
 public double Sroom; // Значение площади элемента, м\*м Определена для Room, Staircase - лестница

public int direct; // Направление движения (определяется при работе программы для Sign = Staircase )  
 public int ntay; // Время обработки Element в единицах tay, мин  
 public int numberOutput; // Номер выхода DoorWayOut 0, 1, 2...  
 public double timeout; // Время достижения зоны безопасности (улицы) через DoorWayOut  
  
 public ArrayList<String> Neigh = new ArrayList<String>(); // Идентификаторы дверей (порталов) из помещения  
 // Идентификаторы помещений, граничащих с дверью

**Распределение людей по элементам здания в условиях ЧС.**

Начальное распределение людей по зданию определяется на основании расписания.

Должны быть сформированы три таблицы:

- Количество людей в группе, которая может находиться в здании ОУ

- Расписание, связывающее момент времени и группы, находящиеся в ОУ

- Количество персонала (преподаватели, учебно-вспомогательный персонал и пр. ) в здании ОУ в текущий момент времени.

**Модель поведения террориста в здании**

Исходим, что модель поведения террориста подчиняется принципу нанесения максимального ущерба для людей, находящихся в здании. определения направления движения нужно рассмотреть несколько моделей:

- Направление обусловлено максимальным градиентом плотности;

- Направление связано с направлением максимального количества людей;

- и т.д.

Возможно, придется ввести категорирование террористов по степени опасности.

**Программы графического интерфейса**