


МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК
Кафедра программного обеспечения

РЕКОМЕНДОВАНО К ЗАЩИТЕ В ГЭК
И ПРОВЕРЕНО НА ОБЪЕМ
ЗАИМСТВОВАНИЯ

Заведующий кафедрой
д.п.н., профессор

 И.Г. Захарова

29 июля 2018 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПОДСИСТЕМЫ МОДЕЛИРОВАНИЯ
РЕЖИМОВ УПРАВЛЕНИЯ ТЕПЛОСНАБЖЕНИЕМ ПОМЕЩЕНИЙ

02.04.03. Математическое обеспечение и администрирование информационных
систем

Магистерская программа «Высокопроизводительные вычислительные
системы»

Выполнил работу
Студент 2 курса
очной формы обучения



Ромазанов
Артур
Ринатович

Научный руководитель
д.п.н., профессор



Захарова
Ирина
Гелиевна

Рецензент
д.т.н., профессор



Захаров
Александр
Анатольевич

Тюмень – 2018

Содержание

Введение.....	4
Глава 1. Моделирование теплообмена в помещении	6
1.1. Способы отопления помещений.....	6
1.2. Схема обогрева помещения системами конвекционного типа	6
1.3. Факторы, влияющие на поглощение тепла	7
1.4. Режимы использования помещений	8
1.5. Регулирование температуры теплоносителя.....	8
1.6. Описание процесса теплообмена в помещении в рамках здания	8
Глава 2. Математическая модель теплообмена в помещениях здания.....	10
2.1. Классификация помещений в здании	10
2.2. Теплообмен со смежными помещениями	10
2.3. Потери тепла в результате нарушения целостности наружных ограждений	11
2.4. Поступающая тепловая энергия	11
2.5. Побочная поступающая тепловая энергия	11
2.6. Постановка прикладной задачи	11
2.7. Основные структуры данных.....	11
2.7.1. Входные данные.....	11
2.7.2. Модель здания.....	12
Глава 3. Архитектура и реализация подсистемы	14
3.1. Архитектура подсистемы	14
3.2. Тепловая карта здания	16
3.2.1. Кригинг	16
3.3. Классификация помещений	17

3.3.1. Классификация по положению.....	17
3.3.2. Классификация по отклонениям от температурного режима.....	18
3.3.3. Классификация по отклонениям от температурных норм отопительных систем.....	18
3.4. Генерация температурного режима.....	19
3.5. Программная реализация.....	20
3.6. Технические характеристики.....	20
3.7. Модуль мониторинга состояния тепловых характеристик здания.....	20
3.7.1. Построение тепловой карты.....	22
3.7.2. Реализация классификации.....	23
3.8. Модуль имитационного моделирования температурного режима.....	24
Глава 4. Вычислительный эксперимент.....	25
4.1. Характеристики здания.....	25
4.2. Генерация температурного режима.....	25
4.3. Классификация по положению.....	26
4.4. Классификация по отклонениям от температурного режима.....	27
Заключение.....	29
Список литературы.....	30
Приложение 1.....	32
Приложение 2.....	33
Приложение 3.....	37
Приложение 4.....	40

Введение

Основным способом отопления помещения являются конвекционные системы. Данные системы достаточно просты в эксплуатации, недороги и эффективны. Зачастую, регулирование их режима работы зависит лишь от погодных условий, однако, технические нормы позволяют производить дополнительную регулировку без риска повреждения сетей теплоснабжения.

Нормативные документы по отоплению помещений и справочные таблицы приведены в пособиях и ГОСТ [1, 2, 3]. Математические модели теплообмена в помещении описаны во множестве книг и статей [4, 5]. Однако в них не приводятся модели режимов теплоснабжения, позволяющих учесть фактическую потребность в обогреве как отдельных помещений, так и всего здания.

Таким образом, существует проблема регулирования режимов работы систем отопления, мониторинга теплового режима и состояния ограждений в помещениях. Изучение теплообмена в помещении позволит составить математическую модель данного процесса и производить имитационное моделирование с целью установления тепловых характеристик помещений и разработки режимов теплоснабжения.

Подходы к решению данной проблемы, описанные в статьях [6, 7, 8], опираются лишь на физические уравнения и не учитывают возможности получения данных с помощью датчиков и прогнозирования ситуации на их основе.

Предлагаемые подходы [9, 10, 11, 12] к реализации системы опираются на работу с микроконтроллерами и не предоставляют удобного инструмента для взаимодействия с данными мониторинга значений тепловых характеристик здания.

Цель магистерской диссертации – разработка подсистемы, позволяющей осуществлять мониторинг теплового режима и производить имитационное моделирование для определения тепловых характеристик здания с целью разработки режимов теплоснабжения.

Задачи:

- изучение процессов теплообмена в здании;
- изучение нормативных документов по нормам отопительных систем;
- разработка модели здания для мониторинга теплового режима, имитационного моделирования для определения тепловых характеристик и определения режима теплоснабжения здания;
- проектирование и разработка подсистемы;
- апробация разработанной подсистемы на реальном здании.

Глава 1. Моделирование теплообмена в помещении

1.1. Способы отопления помещений

Система отопления – система технических средств, позволяющих доставлять тепло от источника до помещения.

Основы типы систем отопления для помещений:

- Водяные системы – тепло передается от нагревательных элементов, через которые проходит нагретая вода;
- Паровые системы – нагрев происходит за счет движения нагретого водяного пара;
- Воздушные системы – пассивная генерация тепла и распространение с воздушными потоками;
- Системы с инфракрасным излучением – нагрев с помощью элементов, выдающих ИК-излучение с нагревом объектов, а не воздуха.

Наиболее распространенная система отопления – водяная, работающая по конвекционной схеме обогрева помещения.

1.2. Схема обогрева помещения системами конвекционного типа

Основным способом обогрева помещений являются отопительные системы конвекционного типа с водяными теплоносителями. Преимущество подобных систем состоит в дешевизне и надежности эксплуатации.

Конвекционные отопительные системы предполагают нагрев воздуха в нижней части помещения, его последующий подъем к верхней части, охлаждение и опускание вниз. На рисунке 1.1 приведена схема конвекционного отопления.

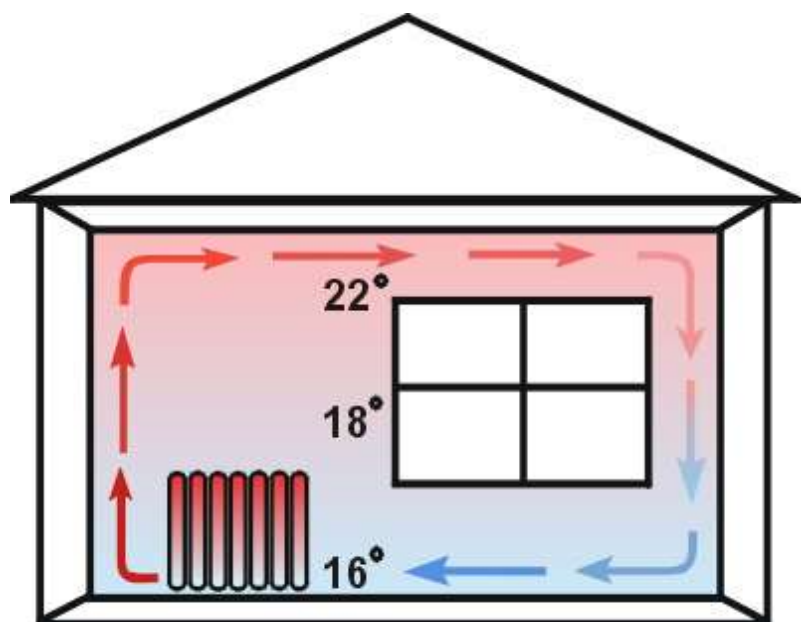


Рис. 1.1. Конвекционная схема отопления помещения

1.3. Факторы, влияющие на поглощение тепла

Системы отопления призваны бороться с рассеиванием тепла в окружающую среду. Основными поглощающими объектами являются:

- Внешние стены и окна (окружающая среда)
- Внутренние стены и перекрытия (смежные помещения)

При учете процесса теплообмена важны характеристики теплоемкости и коэффициента теплопередачи материалов, из которых выполнены объекты [3].

В случае износа поглощающих объектов, поглощение ими тепловой энергии может быть увеличено. Износ может стать причиной появления холодных зон в помещении.

1.4. Режимы использования помещений

Жилые и общественные помещения требуют специальных температурных условий для комфортного нахождения в них [1].

Однако, в случаях, когда помещение не используется, допускаются другие температурные нормы, что позволяет экономить энергию [2].

1.5. Регулирование температуры теплоносителя

В зависимости от температуры наружного воздуха изменяются требования к температуре водяных тепловых сетей [6].

Основные параметры водяной тепловой сети, используемые при расчете итоговой температуры помещения:

- расчетная температура в подающем трубопроводе тепловой сети T_1
- расчетная температура в обратном трубопроводе тепловой сети T_2
- расчетная температура в подающем трубопроводе системы отопления T_3
- Температура наружного воздуха $T_{н.в}$;
- Температура внутри помещения $T_{в.п}$.

1.6. Описание процесса теплообмена в помещении в рамках здания

В процессе теплообмена в помещении участвуют 4 типа объектов:

- Источники тепла
- Внутренние стены и перекрытия
- Смежные помещения
- Внешние стены и окна

Уравнение изменения тепловой энергии воздушной массы Q_1 , заключенной в комнате объемом V :

$$\Delta Q_1 = Q_{п} - Q_{н} - Q_{о} - Q_{д} \quad (1.1)$$

где $Q_{п}$ – поступающая тепловая энергия; $Q_{н}$ – тепловая энергия, поглощаемая смежными помещениями; $Q_{о}$ – тепловая энергия, требуемая на

нагрев объектов в помещении; Q_d – добавочные потери на испарение влаги и т.д.

Аналогичное уравнение для системы контура, в которую входят стены и перекрытия, смежные помещения и система отопительных приборов (стояки и батареи).

$$\Delta Q_2 = C_k(T_1 - T_2) - R_{cm}(T_2 - T_{cm})$$

где C_k – теплоемкость контура; R_{cm} – сопротивление излучения в соседнее помещение; T_{cm} – температура воздуха смежного помещения (°C).

Глава 2. Математическая модель теплообмена в помещениях здания

2.1. Классификация помещений в здании

Количество внешних стен является важным фактором при расчетах температурного баланса. Помещения, в которых количество внешних стен больше, чем в прочих, требуют большей подачи тепла.

2.2. Теплообмен со смежными помещениями

Главная причина тепловых потерь помещения – теплообмен со смежными помещениями. Тепловая энергия Q_n (уравнение 1.1) состоит суммы тепловых энергий относительно каждого смежного помещения

$$Q_n = \sum_{i=1}^n Q_i \quad (2.1)$$

где Q_i – тепловая энергия при обмене с i -ым смежным помещением, n – количество смежных помещений.

Тепловая энергия теплообмена со смежным помещением Q вычисляется по формуле

$$Q = \frac{F}{R_0} (t_B - t_H) n \quad (2.2)$$

где F – площадь ограждения, R_0 – общее сопротивление теплопередаче, t_B – расчетная температура внутреннего воздуха, t_H – расчетная температура наружного воздуха, n – коэффициент, зависящий от положения наружной поверхности ограждения по отношению к наружному воздуху.

Общее сопротивление теплопередаче R_0 вычисляется по формуле

$$R_0 = R_B + \sum_{i=1}^m \frac{\delta_i}{\lambda_i} + R_H$$

где R_B – термическое сопротивление тепловосприятию внутренней поверхности ограждения, R_H – термическое сопротивление теплоотдаче наружной поверхности ограждения, m – количество слоев ограждения, δ_i – толщина i -го слоя, λ_i – коэффициент теплопроводности материала i -го слоя.

2.3. Потери тепла в результате нарушения целостности наружных ограждений

Эксплуатация помещений приводит к износу ограждающих конструкций, в результате которого происходят дополнительные потери тепла. Задача системы отопления учитывать и эти потери.

2.4. Поступающая тепловая энергия

Поступление тепловой энергии происходит за счет работы нагревательных элементов. Отопительные системы подают тепловую энергию не только по средствам радиаторов, но и коммуникационных сетей, которые также можно рассматривать как нагревательные элементы, но менее эффективные.

2.5. Побочная поступающая тепловая энергия

Выделение тепла в помещении также может происходить за счет нахождения людей в помещении и от внесенных объектов, температура которых выше, чем установившаяся в момент появления объектов в помещении. Кроме того, работа оборудования также может вносить значительный вклад в тепловой баланс в помещении. [6]

2.6. Постановка прикладной задачи

На основе показаний датчиков температуры выполнить классификацию помещений и сопоставить с ожидаемыми температурными режимами с точки зрения расположения помещения внутри здания.

2.7. Основные структуры данных

2.7.1. Входные данные

Основными входными данными являются показатели $T = T(x, y, z, t)$ с датчиков температуры (наружных, для измерения температуры окружающей среды и внутренних), где (x, y, z) – координаты относительно плана здания, и t – время замера. Также производится учет данных, получаемых с датчиков ветра в формате (v, d, t) , где v – скорость ветра, d – направление ветра, и t – время замера. Эти данные периодически (в соответствии с заданным

расписанием) запрашиваются сервером, где обеспечивается их хранение в базе данных в случае решения задачи мониторинга. При решении задачи моделирования данные загружаются в том же формате из модуля имитационного моделирования целиком для всего рассматриваемого временного периода.

2.7.2. Модель здания

Отдельное помещение P определяется набором стен (перекрытий) $\{w_i\}_{i=1}^{N_P}$, $N_P \geq 6$. Помещение может содержать дополнительные объекты, влияющие на процессы теплообмена (в частности, окна, двери). Коллекция помещений является зданием.

Помещения рассматриваемого здания являются набором стен и перекрытий и описываются их идентификационными номерами в отдельном файле. При этом помещение может содержать дополнительные объекты, влияющие на процессы теплообмена, возможно задать следующие их виды:

- элементы отопительной системы;
- окно;
- дверь;
- дополнительные элементы, влияющие на циркуляцию воздуха.

Эти объекты описываются координатами и номером соответствующего помещения, к которому они относятся в отдельном файле.

Каждая стена w_i описывается координатами двух точек M_{begin}^i, M_{end}^i , начала стены и ее конца, толщина стены при этом не учитывается. Реальная стена может быть разбита на несколько логических фрагментов (по горизонтали) для более детального описания особенностей помещения. Тепловые характеристики каждого фрагмента стены включают:

- Количество слоев, состоящих из m_K^i разных материалов:
 - Толщину каждого слоя $h_j^i, j = 1..m_K^i$
 - Теплопроводность каждого слоя $a_j^i, j = 1..m_K^i$
- Сопротивляемость теплопередаче внешнего слоя $R_{outside}^i$

- Сопротивляемость тепловосприятию внутреннего слоя R_{inside}^i

Эти характеристики агрегированы в отдельный коэффициент, используемый в уравнении теплового баланса.

Перечень стен задается в отдельном для каждого этажа в файле со сквозной нумерацией для всего рассматриваемого объекта.

Коллекция из K помещений $Z = \{P_j\}_{j=1}^K$ определяет модель здания.

Глава 3. Архитектура и реализация подсистемы

3.1. Архитектура подсистемы

Основные задачи подсистемы – мониторинг тепловых характеристик рассматриваемого здания и моделирование температурных режимов.

Функциональные возможности подсистемы, обеспечивающие решение задач:

- построение и визуализация модели здания;
- моделирование температурного режима здания;
- визуализация тепловой карты модели здания;
- визуализация классификации помещений;

Подсистема разделяется на два модуля:

- модуль мониторинга состояния тепловых характеристик здания;
- модуль имитационного моделирования температурного режима.

Схема взаимодействия модулей подсистемы приведена на рисунке 3.1, где классификации представлены следующим образом:

- Классификация 1 по положению помещений в рамках модели здания.
- Классификация 2 по отклонениям от норм заданного температурного режима.
- Классификация 3 по отклонениям от температурных норм отопительных систем.

Представленная архитектура подсистемы предоставляет инструменты мониторинга, моделирования и прогнозирования, необходимые для решения соответствующих проблем.

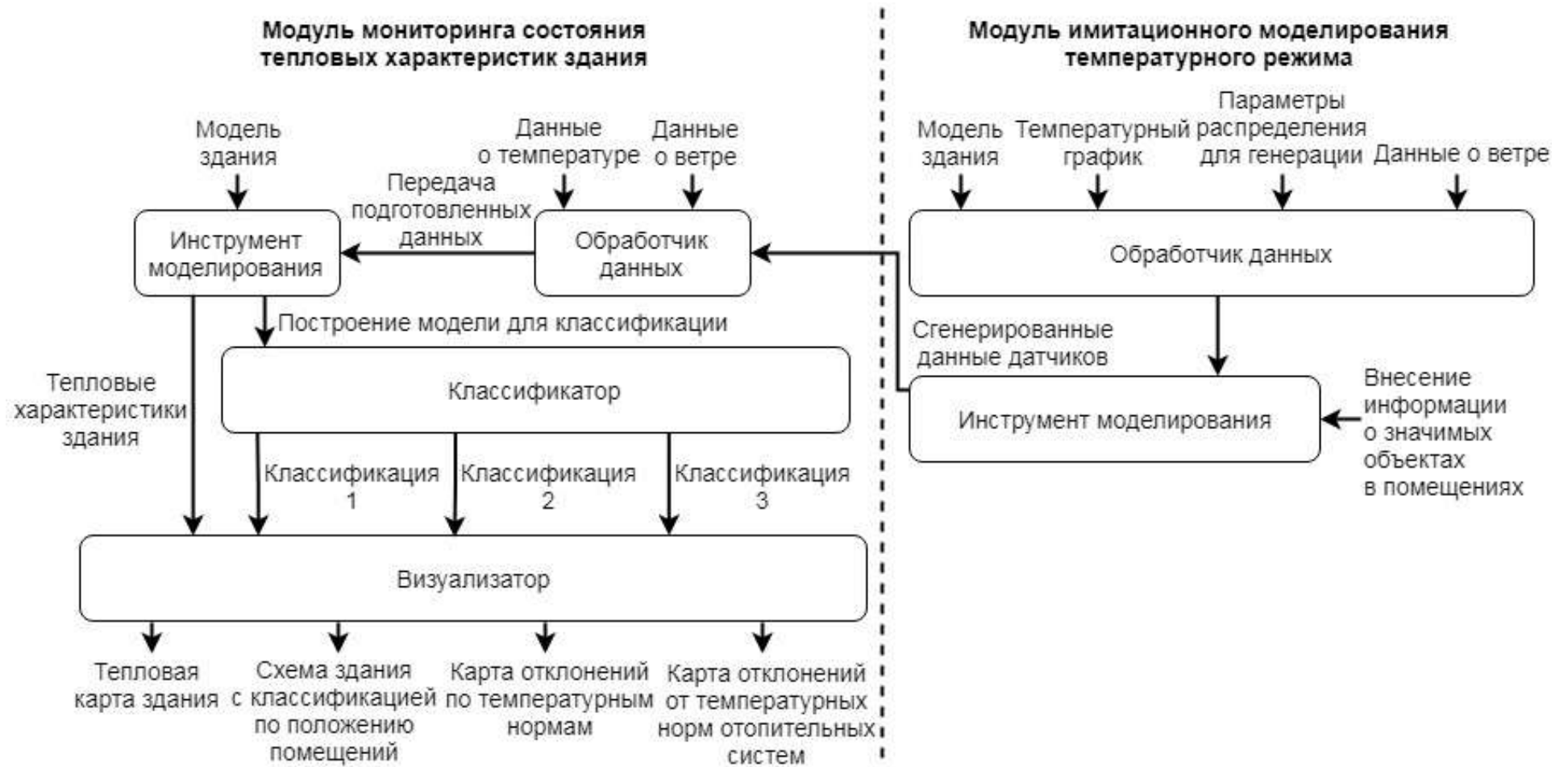


Рис. 3.1. Схема взаимодействия модулей подсистемы

3.2. Тепловая карта здания

Для избранного временного промежутка возможно выполнить визуализацию температурной карты здания. Интерполяция температурных показателей производится с помощью метода кригинг [13] на основании имеющихся данных о температуре для каждого помещения.

Для результатов классификации составляются дополнительные размеченные модели здания.

3.2.1. Кригинг

Кригинг – это улучшенный геостатистический метод, который позволяет строить предполагаемую поверхность из набора точек с z -значениями. В отличие от других методов интерполяции, кригинг предполагает интерактивное исследование пространственного поведения явления, представленного z -значениями, до выбора вами оптимального метода оценки для построения результирующей поверхности [13].

$$\tilde{Z}(s_0) = \sum_{i=1}^N \lambda_i Z(s_i), \text{ где}$$

- s_i - измеряемое значение в местоположении i (координаты на плоскости);
- $Z(s_i)$ - измеряемое значение в местоположении i ;
- λ_i - неизвестный вес для измеряемого значения в местоположении i ;
- s_0 - местоположение прогноза (координаты на плоскости);
- N - количество измеряемых значений.

Методы интерполяции ОВР (обратно взвешенных расстояний) и сплайн относятся к детерминированным методам интерполяции, поскольку они напрямую основаны на измеренных значениях, попадающих в окрестность интерполируемой точки, и на заданных математических формулах, которые определяют сглаженность результирующей поверхности. Вторая группа методов интерполяции состоит из геостатистических методов, таких как кригинг, которые основываются на статистических моделях,

включающих анализ автокорреляции (статистических отношений между измеренными точками). В результате этого геостатистические методы не только имеют возможность создавать поверхность прогнозируемых значений, а также предоставляют некоторые измерения достоверности или точности прогнозируемых значений [13].

При кригинге предполагается, что расстояние или направление между опорными точками отражает пространственную корреляцию, которая может использоваться для объяснения изменения на поверхности. Метод «Кригинг» использует математическую функцию для определенного количества точек или всех точек в пределах заданного радиуса, чтобы определить выходное значение для всех направлений. Кригинг – пошаговый процесс, включающий поисковый статистический анализ данных, моделирование вариограммы, создание поверхности и (дополнительно) изучение поверхности дисперсии. Кригинг лучше всего подходит, если известно, что присутствует пространственно коррелированное расстояние или направленное смещение в данных [13].

3.3. Классификация помещений

Исходя из характера рассматриваемых характеристик помещений и устанавливаемых параметров классификации все используемые классификаторы являются пороговыми. Значения по умолчанию для порогов задаются исходя из требований [1, 2] к теплоснабжению помещений.

3.3.1. Классификация по положению

Значительную роль играет общая площадь стен, выходящих во внешнюю среду. Чем больше эта площадь, тем большая расчетная температура учитывается при определении температурного графика для теплосети, отапливающей соответствующие помещения.

Параметром классификации является количество внешних стен, который задается в интерфейсе программы. Правила разметки при

соответствии помещения конкретному классу задаются в виде записей «количество внешних стен – цвет разметки».

3.3.2. Классификация по отклонениям от температурного режима

Отклонения от температурного режима являются одним из основных показателей, требующих внимания и детального рассмотрения ситуации в конкретном помещении.

Параметрами классификации являются температурные диапазоны, отражающие среднюю температуру в помещении, задаваемыми в интерфейсе.

Правила разметки при соответствии помещения конкретному классу задаются в виде записей «диапазон температур – цвет разметки».

3.3.3. Классификация по отклонениям от температурных норм отопительных систем

Одной из наиболее вероятных причин отклонений от заданного температурного режима является несоответствие тепловой энергии, получаемой от теплоносителей и расчетной температуры.

Параметрами классификации являются температурные диапазоны, отражающие температуру, фиксируемую на элементах отопительной системы в помещении, задаваемыми в интерфейсе. Результатом классификации является размеченная, в соответствии с заданными правилами, модель здания.

Правила разметки при соответствии помещения конкретному классу задаются в виде записей «диапазон температур – цвет разметки».

3.4. Генерация температурного режима

Генерация температурного режима происходит с учетом заданных параметров в том числе и положения здания относительно сторон света.

Важной особенностью является то, что при генерации температурного режима учитывается положение помещения. При визуализации модели в данном модуле, выполняется ее поворот для сопоставления со сторонами света, верхняя часть формы соответствует северу. Положение помещения относительно направления ветра влияет на расчет конечной эффективной температуры наружного воздуха для каждого помещения в отдельности.

Генерация температур происходит относительно заданного температурного графика, который определяет изменение погодных условий в течение суток. Сгенерированные значения являются нормально распределенными, основные параметры распределения (среднесуточная температура и стандартное отклонение) задаются в интерфейсе. Предусмотрена возможность задания направления и скорости ветра для каждого часа в сутках. Периодом генерации по умолчанию являются полные сутки с шагом между контрольными измерениями в 10 минут.

Если для какого-то помещения указаны дополнительные объекты, требующие учета при генерации, то они также обрабатываются. При генерации могут быть учтены следующие виды объектов:

1. Элементы отопительной системы
2. Двери
3. Окна
4. Дополнительные элементы, влияющие на циркуляцию воздуха

Температура отопительной системы генерируется исходя из заданного соответствия температуры воздуха вне здания, относительно каждого помещения, требуемой температуре на нагревательных элементах. Зависимости задаются в интерфейсе программы.

Влияние дверей и окон задается в виде коэффициента, который определяется в интерфейсе программы. Параметр влияет на конечную среднюю температуру в помещении.

При генерации возможна установка отклонений от температурных норм на заданные коэффициенты, которые указываются для каждого помещения. Возможно введение коэффициента для отопительных систем и средней температуры.

В результате генерации для каждой временной отметки из рассматриваемого промежутка создается файл с указанием значений на датчиках с их координатами в здании.

3.5. Программная реализация

Для реализации описанной архитектуры была разработана подсистема, разделенная на два модуля:

- модуль мониторинга состояния тепловых характеристик здания;
- модуль имитационного моделирования температурного режима.

3.6. Технические характеристики

Подсистема разработана на языке программирования C# в среде разработки программного обеспечения Microsoft Visual Studio 2017.

Минимальные системные требования:

- операционная система: Windows 7, 8.1;
- оперативная память: 1 Гб;
- жесткий диск: 10 Мб;
- клавиатура, мышь.

3.7. Модуль мониторинга состояния тепловых характеристик здания

Данный модуль выполняет мониторинг состояния тепловых характеристик здания и классификацию помещений.

На рисунке 3.2 приведена диаграмма классов, используемых для хранения входных данных модуля.

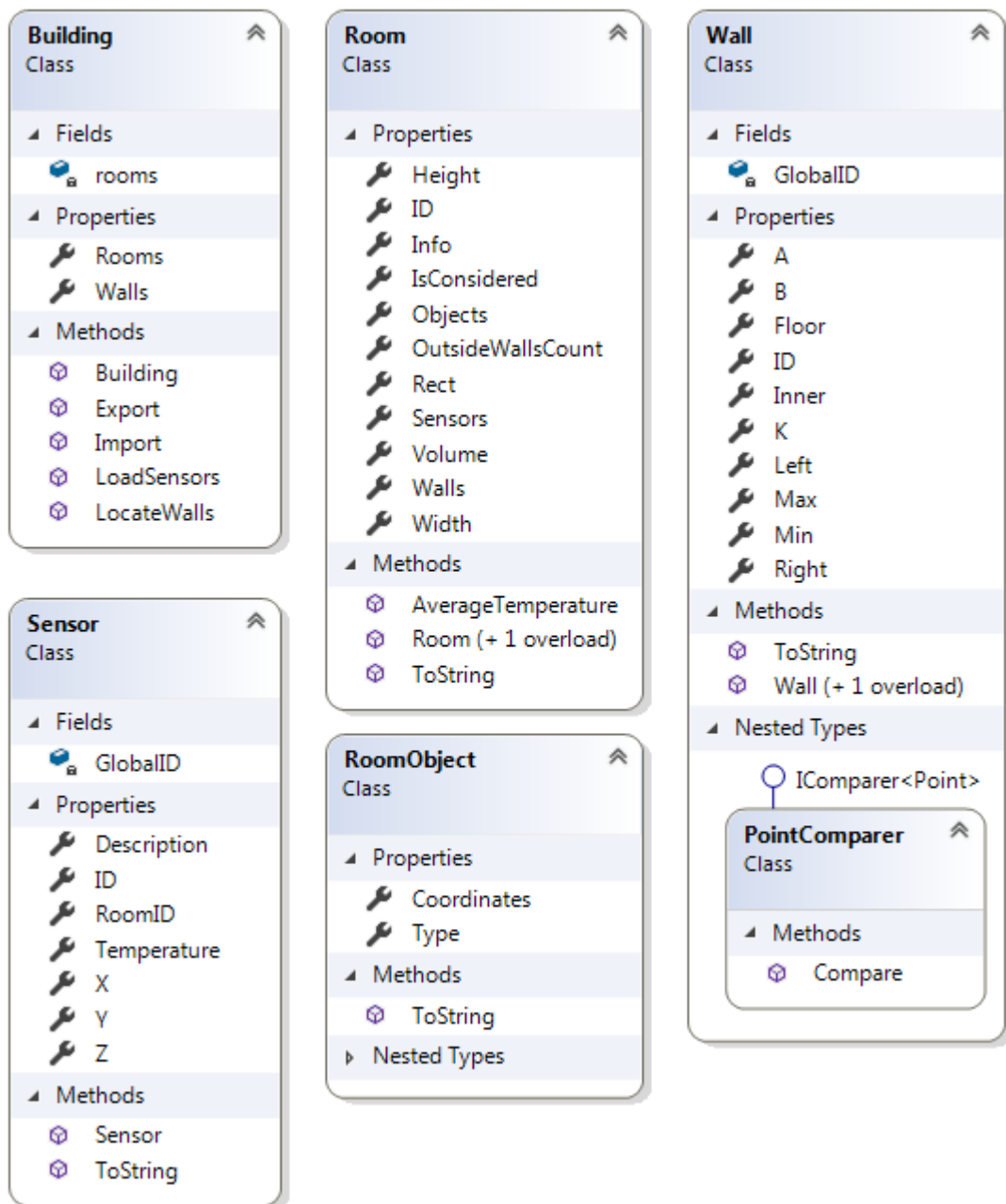


Рис. 3.2. Диаграмма классов для входных данных

Входными данными для модуля служат описанная ранее модель здания, для хранения которой разработаны следующие классы:

- Wall – описание стены.
- RoomObject – описывает доступные объекты для детализации состояния помещения.
- Room – описание помещения, является коллекцией объектов класса Wall и при необходимости дополняется коллекцией объектов класса RoomObject.

- `Building` – определяет здание, содержит коллекцию объектов класса `Room`.

Класс `Building` является сериализуемым для обеспечения возможности импорта и экспорта его экземпляров через файлы `xml`.

Также входными данными являются показания датчиков температуры для помещений, которые описываются классом `Sensor`.

3.7.1. Построение тепловой карты

Тепловая карта является визуализацией модели здания с учетом показаний датчиков для каждого помещения. Для описания визуальных компонентов используются данные хранящиеся в классах `Wall` и `Room`.

Определение раскраски отдельного помещения основывается на показаниях датчиков `Sensor`. В приложении 1 приведен программный код, который отвечает за преобразование числовых значений температуры в цвет.

Для равномерного заполнения помещения используется метод кригинга, программная реализация приведена в приложении 2.

3.7.2. Реализация классификации

Для реализации классификации помещений была реализована классовая структура приведенная на рисунке 3.3.

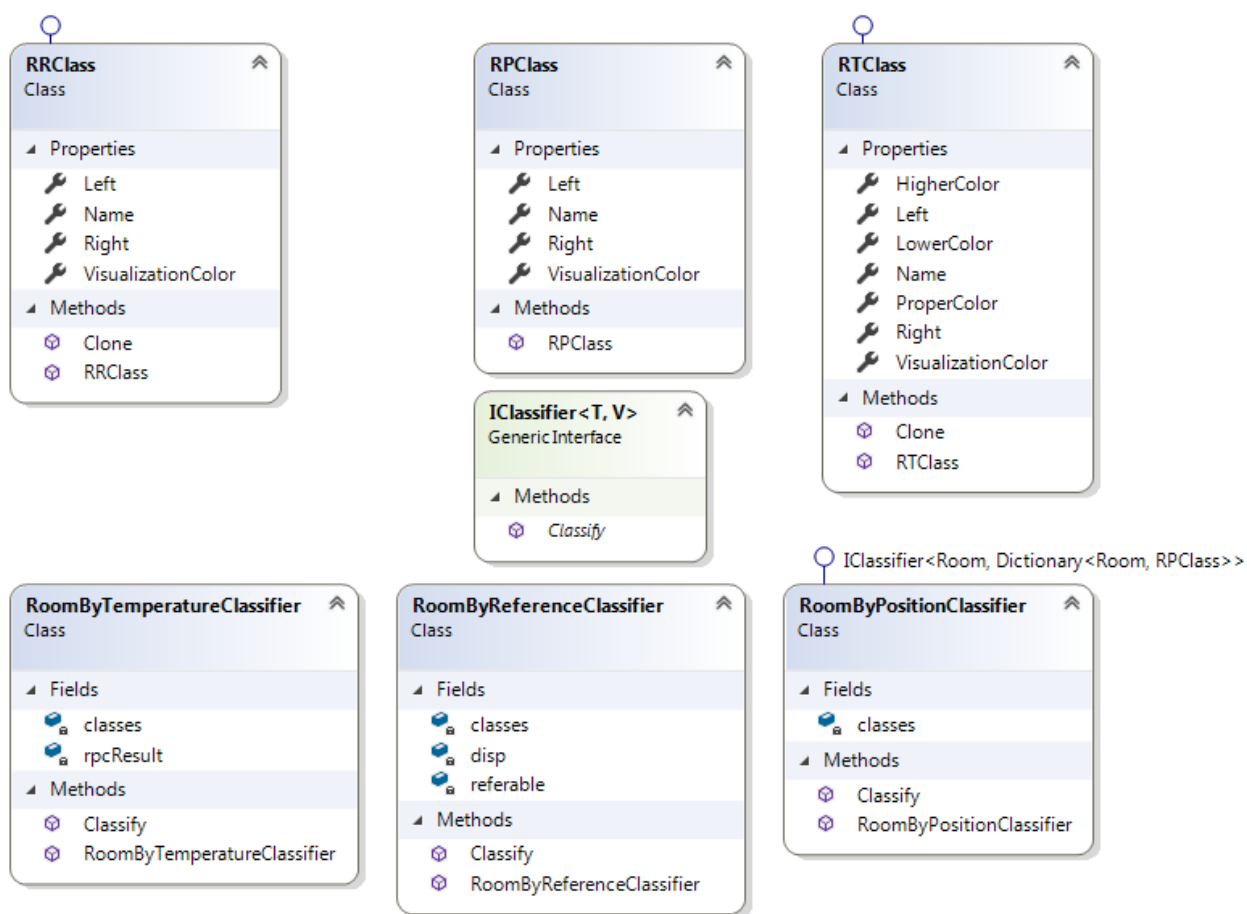


Рис. 3.3. Диаграмма классов для описания классификации

Классы `RPCClass`, `RTClass`, `RRClass` отвечают за хранение результатов классификации по положению, отклонению от температурного режима, отклонению от температурных норм отопительного режима соответственно.

Классы `RoomByPositionClassifier`, `RoomByTemperatureClassifier`, `RoomByReferenceClassifier` отвечают за выполнение классификации по положению, отклонению от температурного режима, отклонению от температурных норм отопительного режима соответственно.

В приложении 4 приведен программный код класса `RoomByReferenceClassifier`.

3.8. Модуль имитационного моделирования температурного режима

Данный модуль выполняет имитационное моделирование температурного режима здания.

Для описания входных данных используется аналогичная приведенной на рисунке 3.2 классовая структура.

При генерации температурного режима помимо описания здания используются задаваемые пользователем показатели отвечающие за генерацию погодных условий (средняя температура, разброс) и описание задаваемых отклонений для выбранных помещений.

В приложении 3 приведен программный код для вычисления эффективной температуры с учетом положения конкретного помещения, направления ветра и заданных параметров генерации.

Глава 4. Вычислительный эксперимент

4.1. Характеристики здания

Для проведения вычислительного эксперимента в качестве здания использовался УЛК №5 Тюменского государственного университета.

Особенности рассматриваемого здания:

- неравномерная этажность по всей площади здания;
- помещения со вторым светом.

4.2. Генерация температурного режима

Для вычислительного эксперимента были заданы следующие параметры:

- среднесуточная внешняя температура $-7,7\text{ }^{\circ}\text{C}$;
- стандартное отклонение $2\text{ }^{\circ}\text{C}$;
- ветер преимущественно северный, средняя скорость ветра 10 м/с .

Данные вводятся в соответствующие поля в интерфейсе модуля имитационного моделирования (рис. 4.1).

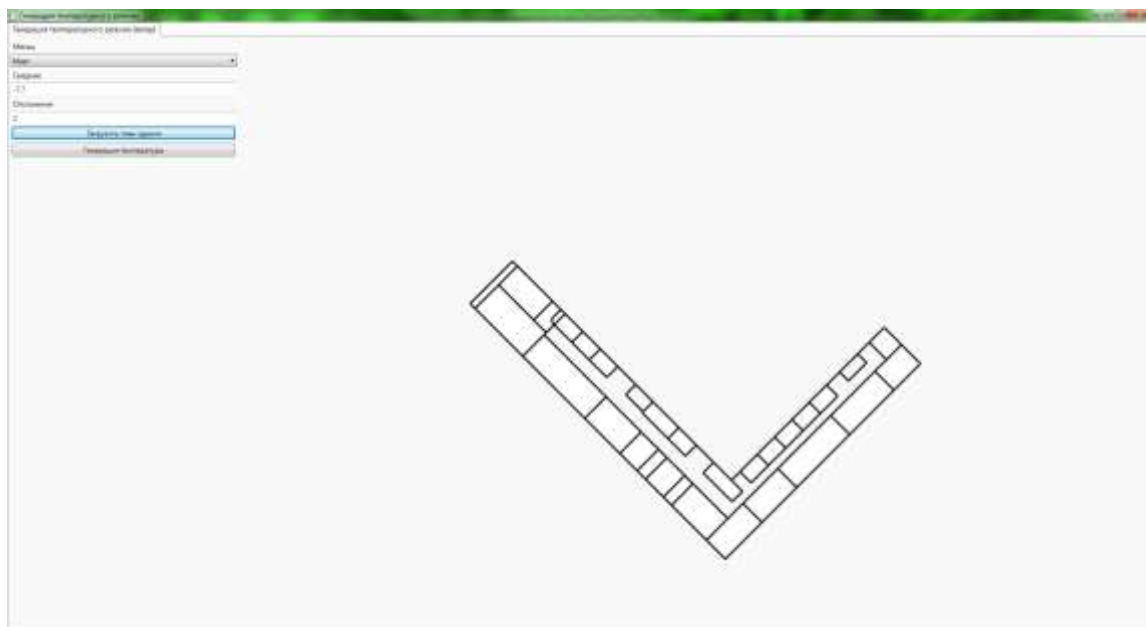


Рис. 4.1. Интерфейс модуля имитационного моделирования

На рисунке 4.2 приведена модель 4 этажа рассматриваемого здания с учетом ориентации по сторонам света.

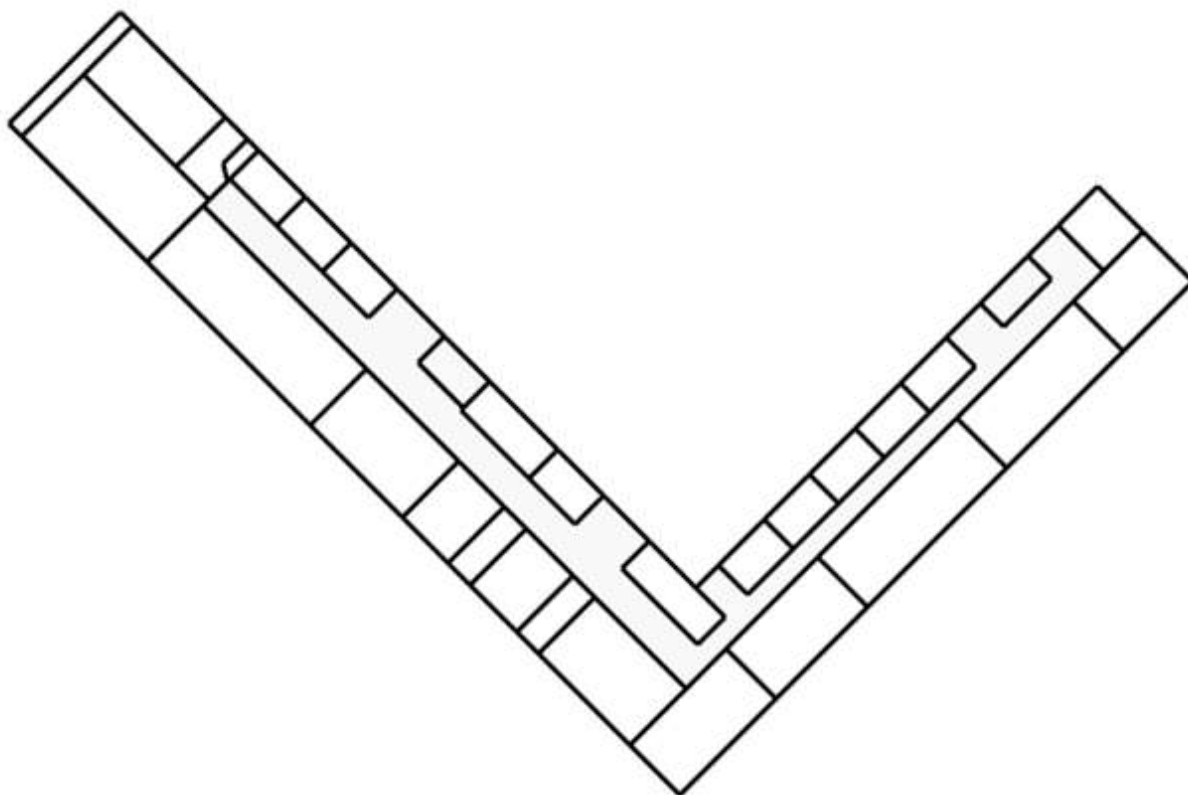


Рис. 4.2. Модель 4 этажа

4.3. Классификация по положению

Для классификации комнат по положению установлены следующие критерии по количеству внешних стен:

- угловая: 2 внешние стены, цвет разметки – красный;
- внешняя: 1 внешняя стена, цвет разметки – желтый;
- внутренняя: внешние стены отсутствуют, цвет разметки – зеленый;
- отдельная: 3 или более внешних стен.

На рисунке 4.3 приведен результат классификации по заданным правилам.

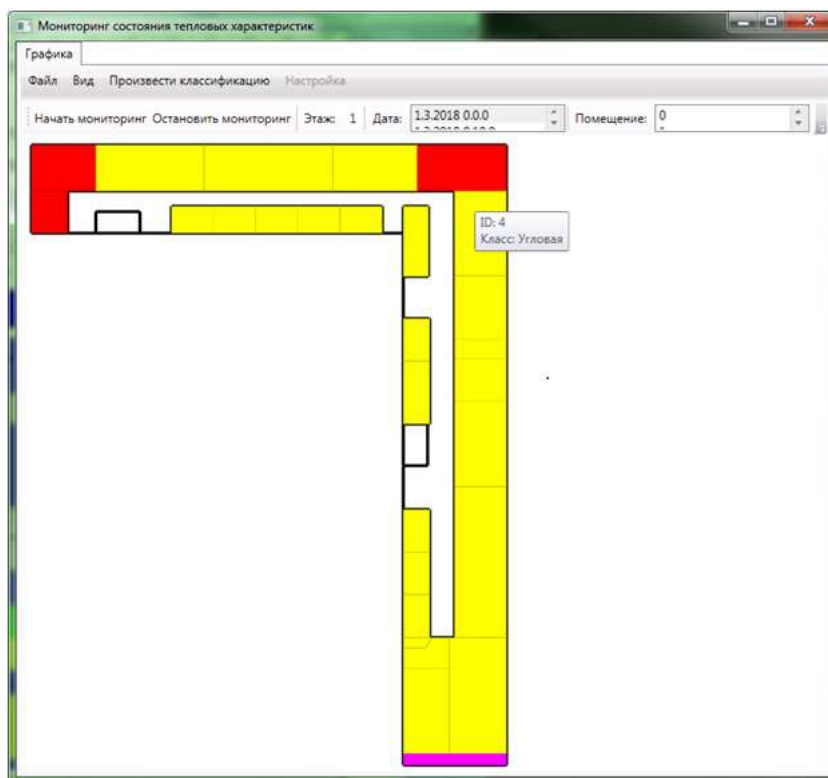


Рис. 4.3. Результат классификации по положению

Всплывающая подсказка содержит информацию о ID помещения и присвоенном классе.

4.4. Классификация по отклонениям от температурного режима

Классификация помещений по отклонениям от температурного режима зависит от результатов их классификации по положению, для соответствующих классов установлены следующие диапазоны допустимых температур:

- угловая: 21-24 °С;
- внешняя: 21-24 °С;
- внутренняя: 19-23 °С;
- отдельная: 21-24 °С.

Определено 3 возможных класса, зависящих от отклонения от заданных диапазонов:

1. Отклонение в меньшую сторону: цвет разметки – синий.
2. Соответствие диапазону: цвет разметки – зеленый.
3. Отклонение в большую сторону: цвет разметки – желтый.

На рисунке 4.4 приведен результат классификации по заданным правилам.

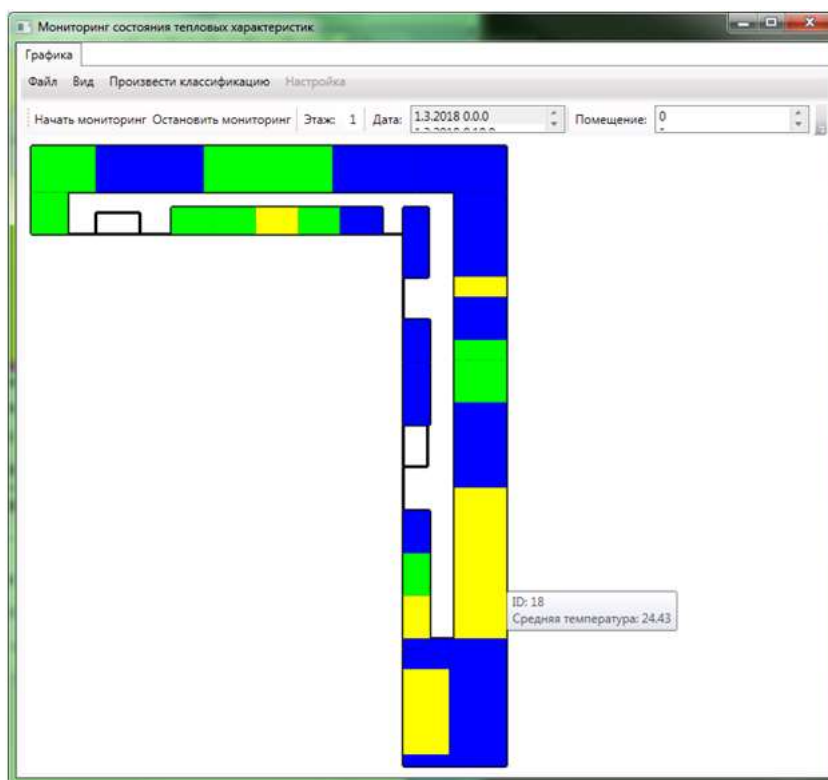


Рис. 4.4. Результат классификации по отклонениям от температурного режима
Всплывающая подсказка содержит информацию о ID помещения и средней температуре в помещении.

Заключение

В ходе выполнения магистерской диссертации была предложена модель здания, позволяющая учитывать необходимые особенности с целью мониторинга состояния тепловых характеристик помещений.

Результатом проделанной работы является подсистема, позволяющая производить мониторинг состояния рассматриваемого здания и выполнять имитационное моделирование режимов теплоснабжения помещений. Проведена апробация подсистемы на УЛК №5 Тюменского государственного университета.

Разработанная подсистема является частью интеллектуальной диспетчерской, которая будет использоваться для управления процессами энергосбережения в обозреваемых зданиях.

Перспективами развития подсистемы являются внедрение автономного управления системами отопления исходя из имеющихся статистических данных и заданных режимов работы.

Список литературы

1. ГОСТ 30494. Здания жилые и общественные. Параметры микроклимата в помещениях. М.: Стандартинформ, 2013. – 11с.
2. СП 60.13330.2012 Отопление, вентиляция и кондиционирование воздуха. Математическая модель регулирования объекта теплопотребления. М.: Минрегион России, 2012. – 75с.
3. Манюк В.И. Наладка и эксплуатация водяных тепловых сетей. / Я.И. Каплинский, Э.Б. Хиж, А.И. Манюк, В.К. Ильин. М.: Стройиздат, 1988 – 432 с.
4. Пуговкин А.В. Математическая модель теплоснабжения помещений для АСУ энергосбережения / А.В. Пуговкин, С.В. Купреков, Д.В. Абушкин, И.А. Заречная, Н.И. Муслимова. – Доклады ТУСУРа, № 2 (22), часть 1, декабрь 2010.
5. Табунщиков Ю.А. Тепловая защита ограждающих конструкций зданий и сооружений / Ю.А. Табунщиков, Д.Ю. Хромец, Ю.А. Матросов. – М.: Стройиздат, 1986. – 380 с.
6. Бурцев В.В. Математическая модель регулирования объекта теплопотребления / В.В. Бурцев, А.С. Басин, Н.М. Байтингер. Ползуновский вестник, №1, 2004 – 302 с.
7. Стрижак П.А. Энергоэффективность системы теплоснабжения зданий при различных методах регулирования теплопотребления / П.А. Стрижак, М.Н. Морозов. Научно-технические ведомости СПбГПУ. Естественные и инженерные науки, 2014 – 202 с.
8. Senave Mariline. A simulation exercise to improve building energy performance characterization via on-board monitoring / Mariline Senave, Glenn Reynders, Stijn Verbeke, Dirk Saelens. 11th Nordic Symposium on Building Physics, 2017 – 1023 с.
9. Gorecki T.T. Predictive Control methods for Building Control and Demand Response / T.T. Gorecki // ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2017 – 171 с.

10. Anvari-Moghaddam A. Cost-Effective and Comfort-Aware Residential Energy Management under Different Pricing Schemes and Weather Conditions / A. Anvari-Moghaddam, H. Monsef, A. Rahimi-Kian // *Energy and Buildings*. 2015. Vol. 86. Pp. 782-793.
11. Ascione F. Simulation-Based Model Predictive Control by the Multi-Objective Optimization of Building Energy Performance and Thermal Comfort / F. Ascione, N. Bianco, C. De Stasio, G. M. Mauro, G. P. Vanoli // *Energy and Buildings*. 2016. Vol. 111. Pp. 131-144.
12. Figueiredo J. A SCADA System for Energy Management in Intelligent Buildings / J. Figueiredo, J. S. da Costa // *Energy and Buildings*. 2012. Vol. 49. Pp. 85-98.
13. Кошель С. М. Методы цифрового моделирования: кригинг и радиальная интерполяция / С. М. Кошель, О. Р. Мусин // *Информационный бюллетень ГИС-Ассоциации*. 2000. No 4. С. 26-30.

Преобразование температуры в цвет

```
public static Brush Translate(double t)
{
    int N = 100;
    double h = (maxTemperature - minTemperature) /
N;

    double middle = maxTemperature -
(maxTemperature - minTemperature) / 2;
    byte r = (byte)Math.Max((Math.Min(255 - 255 *
(maxTemperature - t) / (maxTemperature - middle),
255)), 0);
    byte g = (byte)(255 - 255 * Math.Abs(middle -
t) / (maxTemperature - middle));
    byte b = t < middle ? (byte)(255 - 255 *
Math.Abs(t - minTemperature) / (middle -
minTemperature)) : (byte)0;
    return new SolidColorBrush(Color.FromRgb(r, g,
b));
}
```


Кригинг

```

int density = 10;
foreach (var room in building.Rooms.Values.Where(r
=> r.IsConsidered))
{
    List<Sensor> dateSensors =
room.Sensors[dateLB.SelectedValue.ToString()];
    double width = (room.Width - 2 * wallWidth) /
density;
    double height = (room.Height - 2 * wallWidth)
/ density;

    double avgT = dateSensors.Average(s =>
s.Temperature);

    double[,] MatrixK = new
double[dateSensors.Count, dateSensors.Count];
    for (int m = 0; m < dateSensors.Count; m++)
    {
        for (int n = 0; n < dateSensors.Count;
n++)
            MatrixK[m, n] =
Math.Sqrt(Math.Pow(dateSensors[m].X -
dateSensors[n].X, 2) +
Math.Pow(dateSensors[m].Y -
dateSensors[n].Y, 2));
        MatrixK[m, m] = 1;
    }
}

```

```

        JGInvert(ref MatrixK, dateSensors.Count);

        for (double i = room.Rect.Item1.X + wallWidth;
i < room.Rect.Item2.X - wallWidth; i += width)
        {
            width = (room.Width - 2 * wallWidth) /
density;

            i -= width / 10;
            if (i + width >= room.Rect.Item2.X -
wallWidth)

                width = room.Rect.Item2.X - i;

                for (double j = room.Rect.Item1.Y +
wallWidth; j < room.Rect.Item2.Y - wallWidth; j +=
height)
                {
                    height = (room.Height - 2 * wallWidth)
/ density;

                    j -= height / 10;
                    if (j + height >= room.Rect.Item2.Y
- wallWidth)

                        height = room.Rect.Item2.Y - j;

                        double[] k = new
double[dateSensors.Count];

                        double minDistance =
Math.Sqrt(Math.Pow(dateSensors[0].X - i, 2) +
Math.Pow(dateSensors[0].Y - j, 2));

                        for (int z = 0; z < k.Length; z++)

```

```

        {
            double distance =
Math.Sqrt(Math.Pow(dateSensors[z].X - i, 2) +
Math.Pow(dateSensors[z].Y - j, 2));
            k[z] = distance;
            minDistance =
Math.Min(distance, minDistance);
        }

        double sum = 0;
        for (int z = 0; z <
dateSensors.Count; z++)
        {
            double SK = 0;
            for (int l = 0; l <
dateSensors.Count; l++)
                SK += MatrixK[z, l] * k[l];
            sum +=
dateSensors[z].Temperature * SK;
        }

        double range =
Math.Sqrt(Math.Pow(room.Width, 2) +
Math.Pow(room.Height, 2));
        double lambda =
Math.Min(minDistance / (range / 4), 1);

        double t = sum * (1 - lambda) +
avgT * lambda;

```

```

        Rectangle r = new Rectangle
        {
            Width = width,
            Height = height,
            Fill =
TemperatureTranslator.Translate(t),
            Tooltip =
string.Format("{0}\t{1}", room.ID, t.ToString("0.00"))
        };
        r.SetValue(Canvas.ZIndexProperty,
Convert.ToInt32(ZIndexTextBox.Text));
        r.SetValue(Canvas.LeftProperty, i);
        r.SetValue(Canvas.TopProperty, j);
        heatMap.Add(r);
        canvas.Children.Add(r);
    }
}

```

Расчет эффективной температуры при имитационном моделировании

```
double mu = Convert.ToDouble(MuTB.Text);
double sigma = Convert.ToDouble(SigmaTB.Text);
double temperature = mu;
string side = cs;
double velocity = cv + (nv - cv) * (dt.Minute /
60.0);

double dx = canvas.ActualWidth / 2 *
(side.Contains('W') ? 0 : 1) * (side.Contains('E') ? 2
: 1);
double dy = canvas.ActualHeight / 2 *
(side.Contains('N') ? 0 : 1) * (side.Contains('S') ? 2
: 1);

Point WindPOV = new Point(dx, dy);
Matrix matrix = new Matrix();

double tr = Math.Max(canvas.ActualWidth / 2,
canvas.ActualHeight / 2);
matrix.Translate(-tr, -tr);
matrix.Rotate(360 - 135);
matrix.Translate(tr, tr);
WindPOV = matrix.Transform(WindPOV);
double effectiveTemperature = 33 + (0.478 + 0.237 *
Math.Sqrt(velocity) - 0.0124 * velocity) * (temperature
- 33);
```

```

    string date = string.Format("{0}.{1}.{2}
{3}.{4}.{5}", dt.Day, dt.Month, dt.Year, dt.Hour,
dt.Minute, dt.Second);

    foreach (var room in rooms.Where(r =>
r.IsConsidered))
    {
        int crosses = walls.Count;
        foreach (var wall in room.Walls.Where(w =>
!w.Inner))
        {
            Point center = new Point((wall.A.X + wall.B.X) /
2, (wall.A.Y + wall.B.Y) / 2);
            var coll = walls.Where(w => Crossing(WindPOV,
center, w.A, w.B)).ToList();
            crosses = Math.Min(crosses, coll.Count);
        }

        bool Windward = crosses == 1;
        List<Sensor> sensors = new List<Sensor>();
        double radiatorTemp =
radiatorRates[Math.Max(Windward ?
(int)effectiveTemperature : (int)temperature,
(int)radiatorRates.Keys.Min())] * room.Radiator;
        foreach (var obj in room.Objects)
            sensors.Add(new Sensor() { RoomID = room.ID,
Coordinates = obj.Coordinates, Temperature =
GaussDistribution.GetNumber(radiatorTemp, sigma) });
        sensors.Add(new Sensor() { RoomID = room.ID,
Coordinates = new Point((room.Rect.Item1.X +

```

```
room.Rect.Item2.X) / 2, (room.Rect.Item1.Y +
room.Rect.Item2.Y) / 2), Temperature =
GaussDistribution.GetNumber(21 * room.General, sigma)
});
    inTime[room.ID] = sensors;
}
```

Расчет эффективной температуры при имитационном моделировании

```
class RoomByReferenceClassifier : IClassifier<Room,
Dictionary<Room, RTClass>>
{
    List<RRClass> classes;
    double referable;
    double disp;

    public RoomByReferenceClassifier(string path)
    {
        referable = 20;
        disp = 3;
        classes = new List<RRClass>();
        string t = "";
        using (StreamReader sr = new
StreamReader(path, Encoding.GetEncoding(1251))) t =
sr.ReadToEnd();
        foreach (var row in t.Split(new string[] {
"\r\n" }, StringSplitOptions.RemoveEmptyEntries))
        {
            string[] data = row.Split(' ');
            Tuple<int, int> range;
            if (data[1].Contains('-')) range = new
Tuple<int, int>(Convert.ToInt32(data[1].Split('-')[0]),
Convert.ToInt32(data[1].Split('-')[1]));
            else range = new Tuple<int,
int>(Convert.ToInt32(data[1]),
Convert.ToInt32(data[1]));
```



```

        Color c =
(Color)ColorConverter.ConvertFromString(data[2]);
        classes.Add(new RRClass(data[0],
range.Item1, range.Item2, c));
    }
}

public Dictionary<Room, RRClass>
Classify(List<Room> rooms)
{
    Dictionary<Room, RRClass> res = new
Dictionary<Room, RRClass>();

    foreach (var room in rooms.Where(r =>
r.IsConsidered))
    {
        int missesCount = 0;
        foreach (var date in
room.Sensors.Keys)
            if (referable - disp >
room.AverageTemperature(date) &&
room.AverageTemperature(date) < referable + disp)
                missesCount++;

        res[room] = (RRClass)classes.Find(c =>
c.Left >= missesCount && missesCount <=
c.Right).Clone();
    }
    return res;
}}

```