

Вариативная часть Блок 4

В рамках проектирования структуры хранения данных были созданы 3 таблицы: requirements, data, devices

Таблица requirements содержит данные о названии растений и индивидуальные требования к условиям содержания. name - имя, temp - температура, water - влажность, lk - освещенность, pl - полив, min - минимум, max - максимум, id - идентификатор

id	name	temp_min	temp_max	water_min	water_max	lk_min	lk_max	pl
0	Орхидея Фаленопсис	22	28	60	80	10000	15000	1 y
1	Врацена Фраконова	18	26	50	70	5000	10000	1 v
2	Папоротник Неролепис	18	24	70	90	3000	8000	2 y, v

Таблица data заполнена тестовыми данными для примера
Таблица devices содержит данные об устройствах и датчиках

id	name	type	class
0	Датчик температуры воздуха	датчик	temp
1	Датчик влажности	датчик	water
2	Датчик освещенности	датчик	lk
3	Датчик влажности почвы	датчик	water
4	Увлажнитель	устройство	water
5	Обогреватель	устройство	temp
6	Система полива	устройство	water
7	Талкози	устройство	lk
8	LED-лампа	устройство	lk

name - название, type - датчик или устройство, class - как это влияет
Таблица data заполнена тестовыми данными для примера

Таблица data содержит показания датчиков в определенный момент времени



id	date	time	water temp	water	lk	vl
0	02.02.2026	10:00	20	55	4000	40
1	02.02.2026	18:00	22	70	2000	60
2	03.02.2026	10:00	17	70	8000	65

id - auto-increment идентификатор, date - дата, time - время, temp - температура, water - влажность, lk - освещенность, vl - влажность почвы. Таблица заполнена тестовыми данными

Упрощенная модель (описание функций) и пример написания модуля представления в виде 10

② Алгоритм анализирует текущие показания из таблицы data, а также требования растений из таблицы requirements. В случае, если показания вне допустимого диапазона, то принимается решение об изменении параметров центрального устройства из таблицы devices в соответствии с классом class

```
Database SELECT * FROM 'data' DESC LIMIT 1 id, записана в переменную data
```

```
Database SELECT * FROM 'requirements', записана в переменную requirements,
```

```
Database SELECT * FROM 'devices', записана в переменную devices
let rtemp_min, rtemp_max, rwater_min, rwater_max, rlk_min, rlk_max,
priority-temp=3, priority-water=2, priority-lk=1,
```

```
foreach item in requirements
```

```
if (!(item['temp-min'] >= data['temp'] and item['temp-max'] <= data['temp']))
  if (item['temp-min'] < data['temp']) rtemp_min = item['temp-min'],
  else rtemp_max = item['temp-max'],
```

```
if (!(item['water-min'] >= data['water'] and item['water-max'] <= data['water']))
  if (item['water-min'] < data['water']) rwater_min = item['water-min'],
  else rwater_max = item['water-max'],
```

```
if (!(item['lk-min'] >= data['lk'] and item['lk-max'] <= data['lk']))
  if (item['lk-min'] < data['lk-min']) rlk_min = item['lk-min'],
  else rlk_max = item['lk-max'],
```

```
* rtemp_min = null
```

```
def set-temp(temp, devices)
  foreach item in devices
    if (item['class'] == 'temp' and type == 'устройство')
      item.set-temp(temp)
```

```
def set-water(water, devices)
  foreach item in devices
    if (item['class'] == 'water' and type == 'устройство')
      item.set-water(water)
```



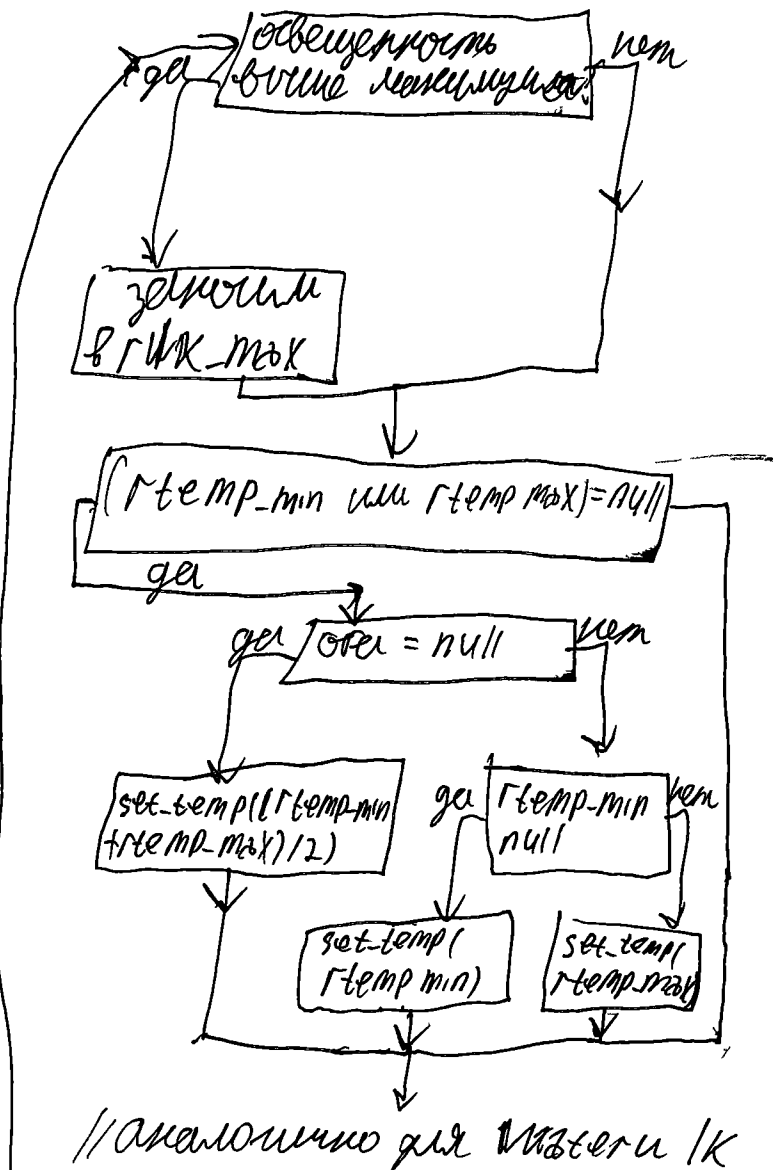
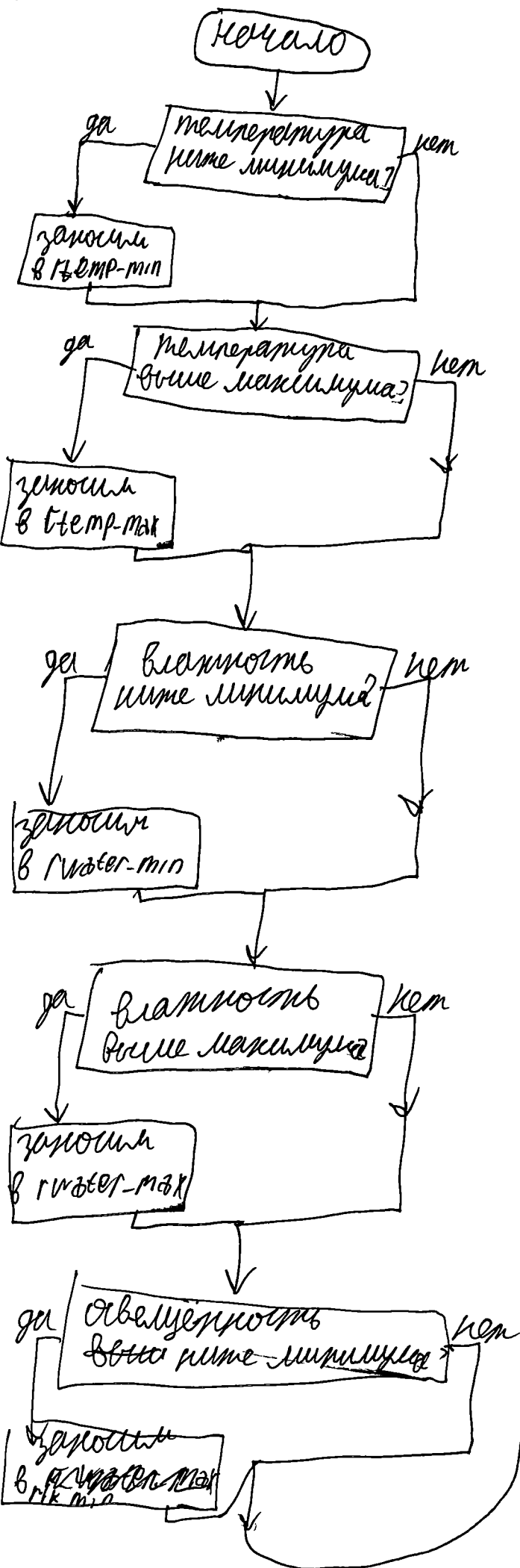
ЛИНИИ УР

```

def set-ik (ik, devices)
  foreach item in devices'
    if (item['class'] == 'ik' and item['type'] == 'устройство')
      item set-ik(ik)
  if (rtemp-min' = null or rtemp-max' = null)
    if (rtemp-min' = null and rtemp-max' = null)
      set-temp((rtemp-min + rtemp-max) / 2)
    elif (rtemp-min' = null)
      set-temp(rtemp-min)
    else
      set-temp(rtemp-max)
  if (rwater-min' = null or rwater-max' = null)
    if (rwater-min' = null and rwater-max' = null)
      set-water((rwater-min + rwater-max) / 2)
    elif (rwater-min' = null)
      set-water((rwater-min));
    else
      set-water(rwater-max)
  if (rik-min' = null or rik-max' = null)
    if (rik-min' = null and rik-max' = null)
      set-ik((rik-min + rik-max) / 2),
    elif (rik-min' = null)
      set-ik(rik-min)
    else
      set-ik(rik-max)
  
```

С учетом приоритетов, что температура важнее всего, далее идет влажность, а потом освещенность, код выше получает данные из дат, пропускает по индивидуальным требованиям растений и определяет критические отклонения функции set x изменяют температуру характеристики устройств, отвечающих за отклонения если для одного растения температура критически важна, а для другого "низкая", то определяется "золотая середина"

Flow-схема



③

1) В примере id = 1 из таблицы data все показатели в норме, кроме освещенности

~~temp~~ r1k_min = 3000

set - ik (3000, devices) установит освещенность в 3000 лк, что устроит критическое отклонение и обеспечит минимизацию энергозатрат

2) В примере id = 2 из таблицы data все показатели в норме, кроме температуры

rtemp_min = 78

set - temp 718, devices) установит температуру в 78°C, что устроит критическое отклонение и обеспечит минимизацию затрат

3) В примере id = 3

Для полива всей теплицы за раз нужно $30 \cdot 2 = 60$ литров в сутки в приведенном примере ~~это~~ полив орхидей факелотеле и драцены происходит утром или днем. При этом, полив папоротника неорхидеями происходит утром и днем ^{драценова} следовательно, нужно поливать в сутки теплицу от 1 до 2 раз каждыми "10"

4) Требования к механизмам принятия решений
 а) Такая автоматизация определит необходимые показатели для предотвращения отклонений это полив, изменение температуры, освещенности и влажности

б) Сенсоры, предусматривающие человеческое вмешательство
 датчик влажности, датчик температуры, датчик освещенности, датчик влажности, датчик температуры, датчик влажности

2) Требования к адаптивности системы

а) Функциональные решения, которые помогут адаптировать систему при добавлении новых растений код выше берет данные о растениях из базы данных чтобы подключить сайт или

бота в мессенджере, например МАХ, в котором
будет добавляться в requirements новая строка с
действиями

б) Ботские настройки позволят анализ, изменить
метрики расширенные функции: отчет в формате
pdf/excel со статистикой, объемами работы
системы

3 Архитектура с выделением функциональных
блоков

а) Сбор и обработка данных с датчиков IoT-усть-
ройств посредством sql/файл данных и запись ре-
зультатов в таблицу data при использовании таб-
лицы fences

б) Оптимизация и принятие решений с алгоритмами
ML и правилами: можно интегрировать модель
YOLO, которая будет следить за размером соствя-
нием растений по видеонамерам. Так как не
всегда показатели могут точно говорить о со-
стоянии растения фантомная идет посредством
анализа таблицы data и логов датчиков, вы-
пуске отчетов

в) Управление и мониторинг устройств
и контроль состояния: итеративные
устройства хранятся в таблице fences их
можно удалять, добавлять через интерфейс
СБД контроль состояния посредством логов

г) Взаимодействие с пользователем и визуали-
зация данных через бота в мессенджере или
веб-интерфейс. Пользователь работает с
таблицами в виде логов, может загрузить
ботские действия

Визуализация с помощью отчетов в
различных форматах