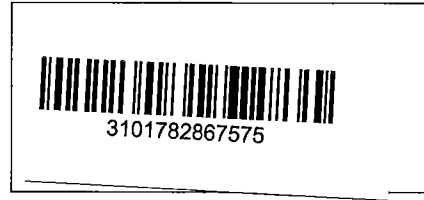


ИЗУМРУД СТУДЕНТ
И ИАДА АЛБС ОГ ЕД АЛЬНОГО У И Т



Титульный лист

Направление Естественные науки Инженерные науки
 Математика и информатика Социальные и гуманитарные науки
 Экономика и управление

Вариативный блок 1 2 3 4 5

Курс 1 2 3 4 5 отсутствует

Фамилия Д Е Н И С О В

Имя А Л Е К С А Н Д Р

Отчество А Л Е К С Е Е В И Ч

Дата рождения 16 08 2005

Город участия Е К А Т И Р И Н Б У Р Г

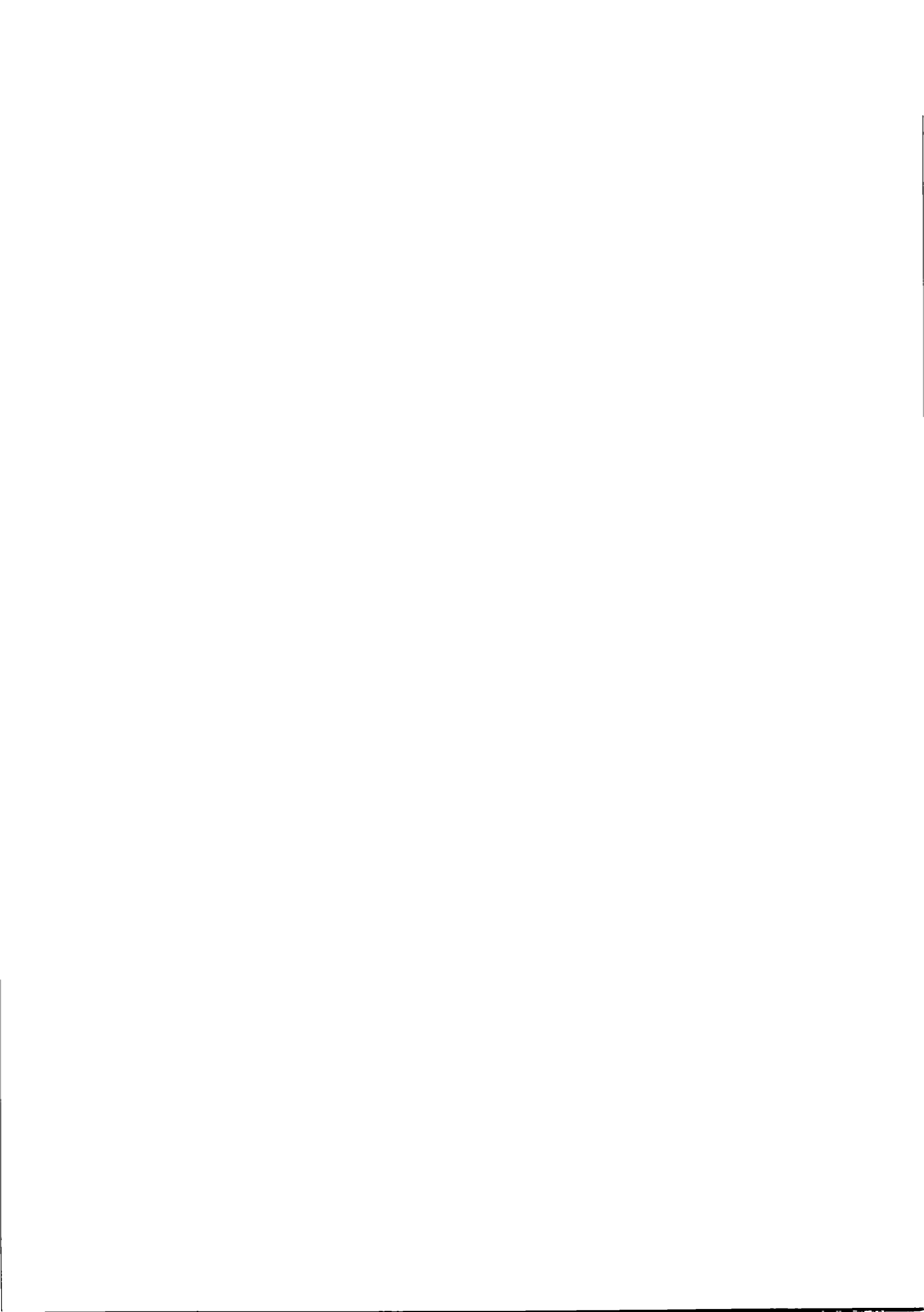
Аудитория 206

Дата 02 02 2026

Подпись

Пример
заполнения

А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У Ф
 Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я 1 2 3 4 5 6 7 8 9 0



ВАРИАНТЫ И БЛОК №4
ТЕПЛИЦА ТРОПИКИ

1) Для каждого показателя будет своя структура с датчиками для удобства поиска и сертировки

t_{emp} - отвечает за температуру
она сертирует

ξt_{act} - показатель с датчика в момент измерения

$t_{perfmax}$ - показатель максимальной температуры, которая устроит все растения

$t_{perfmin}$ - показатель минимальной температуры, которая устроит все растения

t_{dif} - разница между t_{act} и ($t_{perfmin}$ или $t_{perfmax}$)
если выходит в их границы, то равна 0

air - показатель для влажности воздуха
сертирует

ξa_{act} - влажность в момент замера

$a_{perfmax}$ - максимальная

$a_{perfmin}$ - минимальная влажность для всех растений

a_{dif} - разница между a_{act} и ($a_{perfmin}$ или $a_{perfmax}$)
или 0

$light$ - отвечает за показатели света

ξl_{act} - освещенность в момент замера

$l_{perfmax}$ - максимальная

$l_{perfmin}$ - минимальная освещенности для растений

l_{dif} - ~~разн~~ разница между l_{act} и ($l_{perfmin}$ или $l_{perfmax}$)
или 0

ground - уровень земли в абсолютных координатах

{ gact - высота здания в абсолютных координатах
замера

gperfmin - минимальная

высота здания над уровнем

gperfmax - максимальная

gdif - разница между gact и (gperfmax или gperfmin) и
или 0

Все эти значения можно хранить в одном

JSON файле, который будет вызывать здание и параметры
замеров

Формат записей

02 02 2026_10.00.json

```
{ // temp
```

```
{ tact - 20
```

```
  tperfmax - 24
```

```
  tperfmin - 22
```

```
  tdif 2
```

```
}
```

```
// air
```

```
{ aact = 55
```

```
  aperfmax 70
```

```
  aperfmin - 70
```

```
  adif adif 15
```

```
}
```

```
// light
```

```
{ lact = 4000
```

```
  lperfmin - 8000
```

```
  lperfmax = 10000
```

```
  ldif = 4000
```

```
}
```

```
// ground
```

```
{ gact = 40
```

```
  gperfmax = 100
```

```
  gperfmin - 100
```

```
  gdif = 20 }
```

N2

Поскольку в данных уже есть отклонение от нуля, можно считать как его невозможность функцией на m уровне

```
def dif(max, min, act):
```

```
    return min(abs(max-act), abs(act-min))
```

Также функция где находим perf min и perf max
 предположим что все требования это губернский
 массив ~~где~~ $A[i] = [min, i, max], \dots$

тогда

```
def perfmax(A):
    ans = A[0][0]
    for i in range(len(A)):
        ans = max(A[i][0], ans)
        ans = min(A[i][0], ans)
    return ans
```

```
def perfmin(A):
    ans = A[0][0]
    for i in range(len(A))
    ans = max(ans, A[i][0])
```

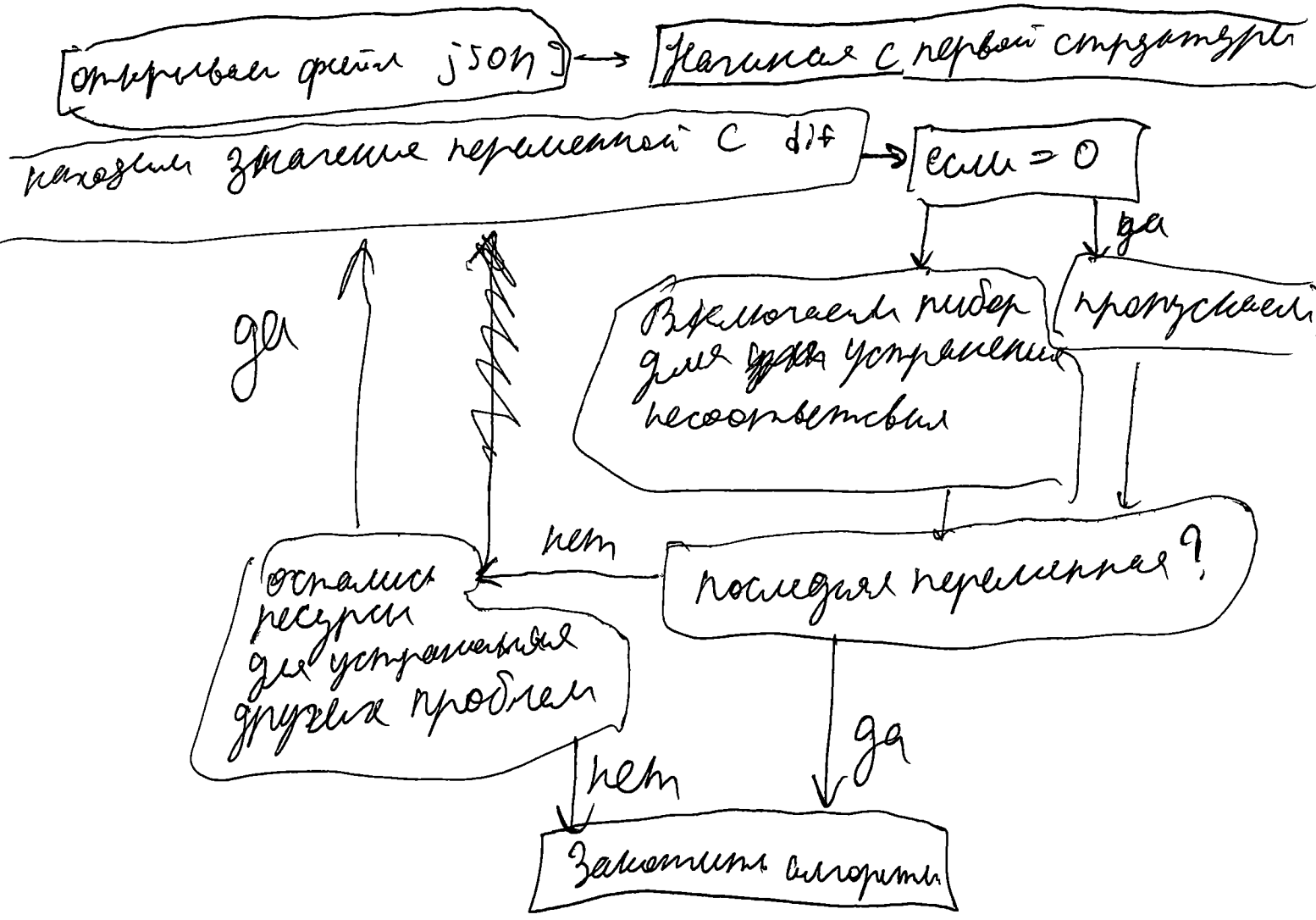
```
if perfmin > perfmax:
```

print("Невозможно создать идеальную среду для
 всех растений одновременно")

```
temp = perfmin
perfmin = perfmax
perfmax = temp
```

Для ~~правильной~~ реализации алгоритма нужно знать ~~какие~~ приоритеты у показателей

Для удобства возьмем их как в нашей статье ~~json~~ json, при необходимости в будущем можно будет поменять порядок по приоритетам.



Будущий алгоритм будет управлять проблемами по приоритету (расположение в файле json) и по оставшимся ресурсам.

3) Все несоответствия

1. повысить температуру на 2°C (до 22°C)

2. увеличить влажность до 70% (на 15%)

3. повысить освещенность до 1000лк (на 4000лк)

Это все еще не обеспечит оптимальный рост всех

но будет более близкое значение для орхидеи

4. увеличить полив (до 60% минимума)

Для расчета предположили что каждое растение занимает $\frac{1}{3}$ от всей площади

$$72 \cdot 30 \cdot \frac{1}{3} = 720 \text{ м}^2$$

Умножив площадь на количество растений в день
 $720 \cdot 1 + 720 \cdot 1 + 720 \cdot 1 = 2160$ столько нужно поливать в день

$720 \cdot 2 = 1440$ л воды в день на полив 10

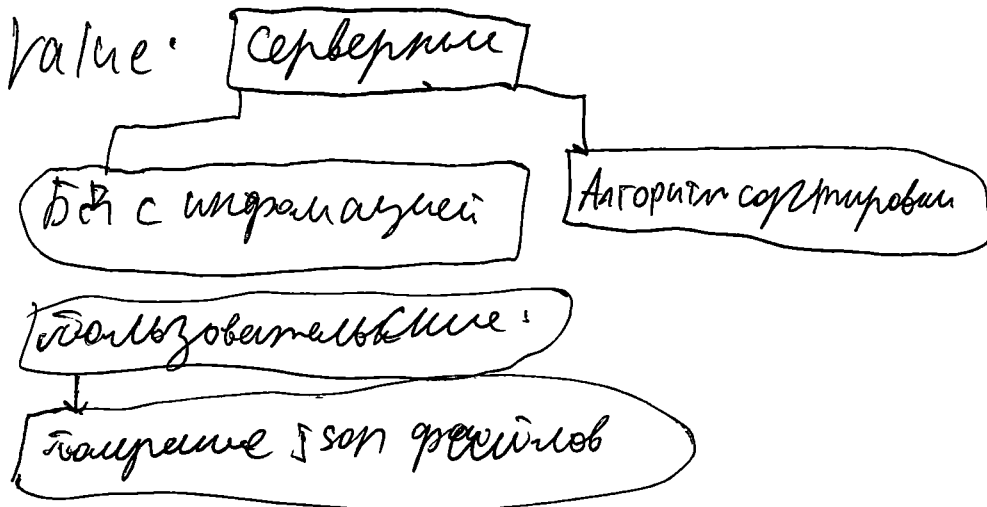
4) Даша

/value - для данных

/ML для решения алгоритмов

/control - для управ устройств

/UI - для взаимодействия



ML Серверные

Тринадцатые решения и их воспроизведение

Обработка данных для принятия решений

Клиентские

Документы ~~по~~ информации о критериях решений

Control

Серверные

~~по~~ отправка данных с сервера в БД

проверка оборудования клиента по данным

Клиентские

Документы отчет о состоянии приборов

UI

Серверные

интерфейсы со всеми группами данных

Клиентские

документы данные с групп данных по выбору

Документы таблицы с данными

Документы графики с данными

Алгоритмические решения тех, где алгоритмы могут найти оптимальное решение или наоборот увидеть ^{Клиент-сер} ~~Клиент-сер~~ ~~Таблицы~~ и обработка информации ~~Эксперт~~ Эксперт нужен при 1) обработке оборудования

2) алгоритмы не могут найти оптимальное решение

ДОПОЛНИТЕЛЬНЫЕ ЛИСТЫ

Для агентами системы к решению можно
предложить использовать блокную структуру
где каждый блок отвечает за свою часть
Например блок данных в БД, блок записей
для пользователей и другие компоненты ИС, которые
можно вынести в отдельные блоки

Также стоит отказаться от констант в алгоритмах
ведь это может привести к ошибкам в будущем

Уровни агентами. 1) Базовые изменения.

Нужно будет заново определить какие данные
будут с новыми растениями. Обновить БД растениями
добавить новые проверить корректность работы
системы

2) Расширение функций. Блокный подход
поможет изменить код только в одном блоке
сохраняя работоспособность других

3) ~~Клиент~~ Нововведения: открыть сайт и др

Нужно будет написать новый фронт и использовать
данные которые уже есть в существующей
базе как

15

