

Установка

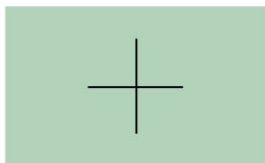
1. Разархивировать файл в удобную для работы директорию
2. Установить Docker Desktop:
<https://docs.docker.com/desktop/install/windows-install/>
3. Запустить приложение
4. Открыть PowerShell, зайти в директорию с проектом командой `cd <path_to_directory>`:

```
PS C:\Users\Anna> cd C:\Users\Anna\Desktop\GitHub\snow_level_report
```
5. Выполнить команду `docker-compose up -d --build`
6. Открыть в браузере `http://localhost:3000`

Работа с приложением

Откроется приветственная страница:

Создать отчёт



Чтобы создать объект наблюдения, жмём на плюсики

Создать отчёт



Название объекта

ГП-2

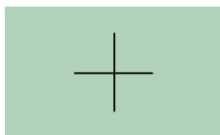
Комментарий

комментарий

Сохранить

Нажимаем на объект, в который хотим загрузить набор изображений

Создать отчёт



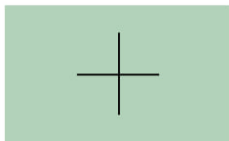
ГП-2

0 папок

ГП-2

комментарий

Удалить



Назад

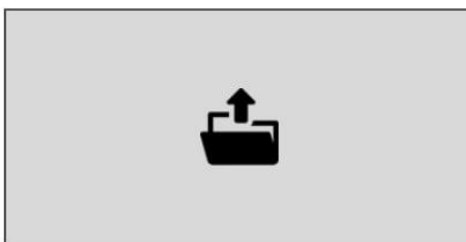
Тут тоже нажимаем на плюс. Загружаем zip-архив с изображениями (без вложенных папок)

ГП-2/Новый набор

1 Upload — 2 Validation — 3 Report

Комментарий

отличный батч



GP2-1.zip

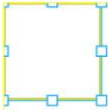
Загрузить

ГП-2/Новый набор

✓ Upload

2 Validation

3 Report



Рейка
Верхний левый угол: (0.000, 0.000)
Нижний правый угол: (0.000, 0.000)
Высота рейки (см):
Распознанное значение (см): 0.000

Тип фото
Верхний левый угол: (0.000, 0.000)
Нижний правый угол: (0.000, 0.000)
Распознанное значение: ✗

Дата и время
Верхний левый угол: (0.000, 0.000)
Нижний правый угол: (0.000, 0.000)
Распознанное значение:

Температура
Верхний левый угол: (0.000, 0.000)
Нижний правый угол: (0.000, 0.000)
Распознанное значение: ✗

Попробовать случайную фотографию из набора

Распознать

Обработать набор данных

Объект


Нажимаем “Попробовать случайную фотографию из набора”. Наводим прямоугольники на области вырезки и жмём “Распознать” для проверки. Высоту рейки указывать без части, закрытой травой. Высота снега вычисляется относительно самой нижней видимой части рейки.

ГП-2/Новый набор

✓ Upload

2 Validation

3 Report



Рейка
Верхний левый угол: (2666, 248.2)
Нижний правый угол: (3003, 2795)
Высота рейки (см):
Распознанное значение (см): 0.1963

Тип фото
Верхний левый угол: (1426, 2946)
Нижний правый угол: (1586, 3031)
Распознанное значение: Т ✗

Дата и время
Верхний левый угол: (2084, 2951)
Нижний правый угол: (2613, 3026)
Распознанное значение: 2022-10-12 10:00:01

Температура
Верхний левый угол: (3221, 2957)
Нижний правый угол: (3397, 554.7)
Распознанное значение: -10 ✗

Попробовать случайную фотографию из набора

Распознать


Обработать набор данных

ГП-2/Новый набор

✓ Upload

2 Validation

3 Report



Рейка
Верхний левый угол: (2666, 248.2)
Нижний правый угол: (3003, 2795)
Высота рейки (см):
Распознанное значение (см): 9.620

Тип фото
Верхний левый угол: (1426, 2946)
Нижний правый угол: (1586, 3031)
Распознанное значение: T
✗

Дата и время
Верхний левый угол: (2084, 2951)
Нижний правый угол: (2613, 3026)
Распознанное значение: 2022-11-25 14:00:01

Температура
Верхний левый угол: (3221, 2957)
Нижний правый угол: (3397, 554.7)
Распознанное значение: -25
✗

Попробовать случайную фотографию из набора

Распознать

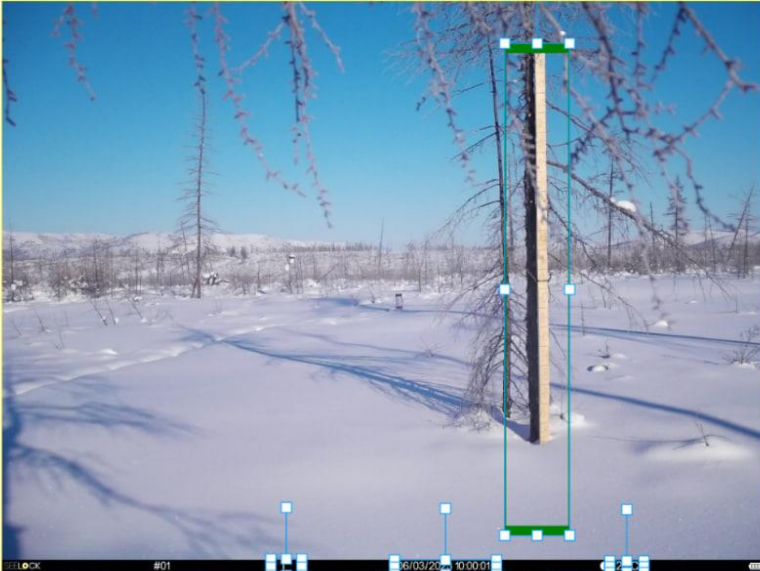
Обработать набор данных

ГП-2/Новый набор

✓ Upload

2 Validation

3 Report



Рейка
Верхний левый угол: (2666, 248.2)
Нижний правый угол: (3003, 2795)
Высота рейки (см):
Распознанное значение (см): 43.00

Тип фото
Верхний левый угол: (1426, 2946)
Нижний правый угол: (1586, 3031)
Распознанное значение: T
✗

Дата и время
Верхний левый угол: (2084, 2951)
Нижний правый угол: (2613, 3026)
Распознанное значение: 2023-03-06 10:00:01

Температура
Верхний левый угол: (3221, 2957)
Нижний правый угол: (3397, 554.7)
Распознанное значение: -23
✗

Попробовать случайную фотографию из набора


Распознать

Обработать набор данных

Также можно нажать на крестик, если тип или температура необязательны

CALM/Новый набор

✓ Upload — 2 Validation — 3 Report



Рейка
Верхний левый угол: (1875, 204.4)
Нижний правый угол: (2276, 2170)
Высота рейки (см): 250
Распознанное значение (см): 17.17

Тип фото
Отсутствует
+

Дата и время
Верхний левый угол: (2068, 2951)
Нижний правый угол: (2608, 3037)
Распознанное значение: 2022-11-14 14:00:02

Температура
Отсутствует
+

Попробовать случайную фотографию из набора Распознать Обработать набор данных

Если что-то не распознаёт, стоит подкорректировать прямоугольники. Если всё в порядке, нажимаем “Обработать набор данных”. В процессе обработки набора можно смотреть график

ГП-2/Новый набор

✓ Upload — ✓ Validation — 3 Report

По дням

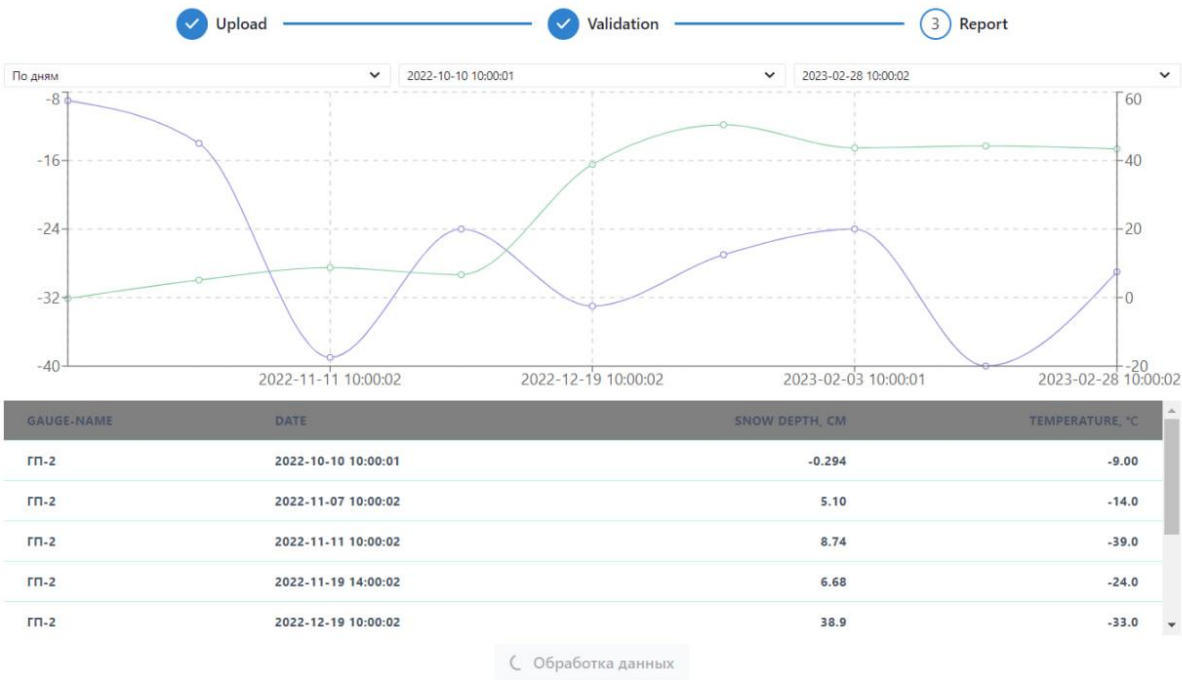
GAUGE-NAME	DATE	SNOW DEPTH, CM	TEMPERATURE, °C
------------	------	----------------	-----------------

↶ Обработка данных

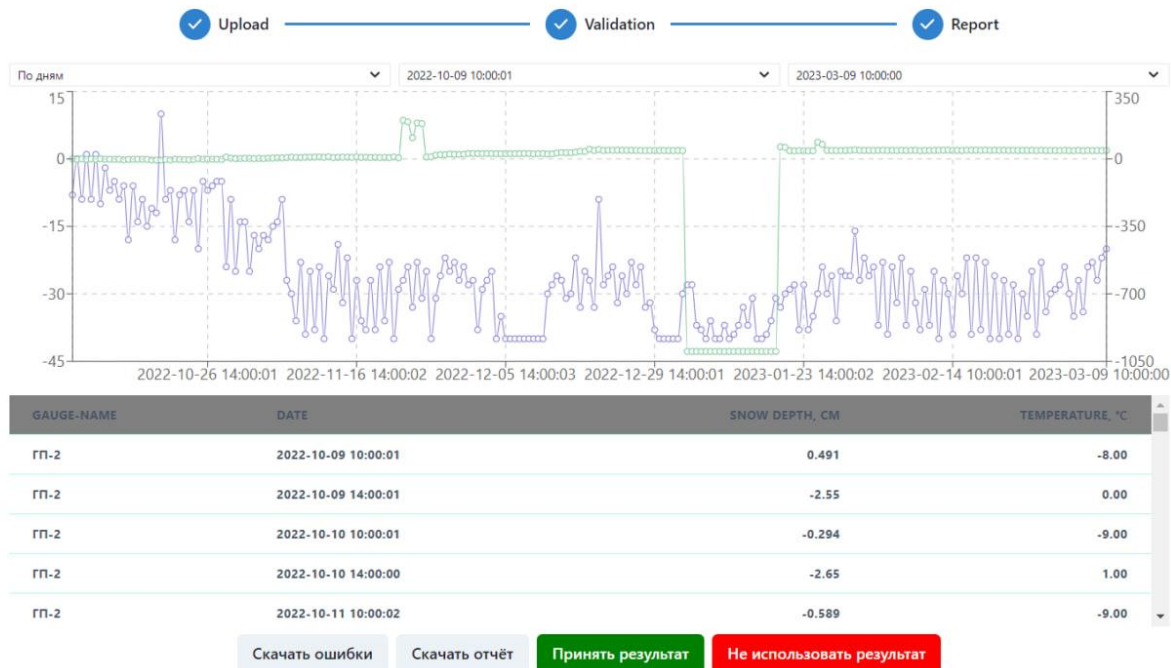
ГП-2/Новый набор



ГП-2/Новый набор

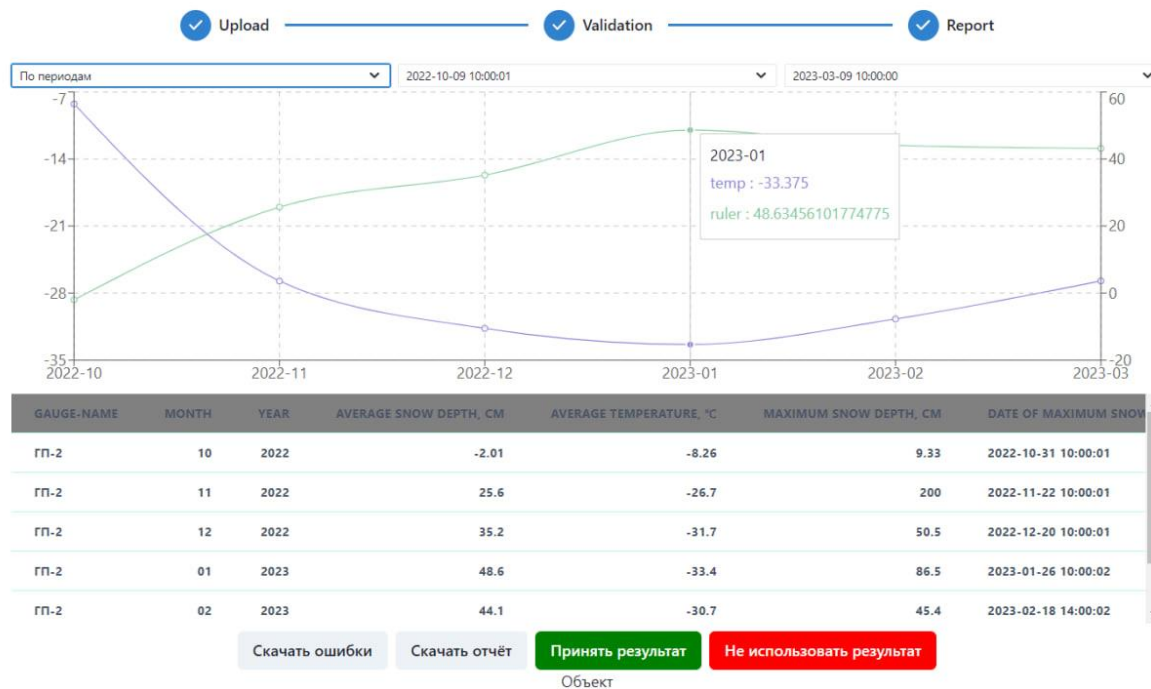


ГП-2/Новый набор



Сверху можно выбирать даты или тип отчёта

ГП-2/Новый набор



“Скачать ошибки” загружает zip-архив следующего вида:

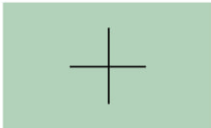
Имя	Тип	Путь	Дата и время
..	Папка с файлами		
coordinates	Папка с файлами		05.09.2023 23:00
datetime	Папка с файлами		05.09.2023 23:02
snow_level	Папка с файлами		05.09.2023 23:03
temp	Папка с файлами		05.09.2023 23:03
type	Папка с файлами		05.09.2023 23:00

В каждой папке изображения с ошибкой распознавания ключа, указанного в названии папки. В coordinates изображения сохраняются в случае если прямоугольники находятся за рамками изображения.

“Скачать отчёт” загружает excel-файл.

Нажимаем “Принять результат” или “Не использовать результат” в зависимости от того, устраивает ли он нас.

Создать отчёт

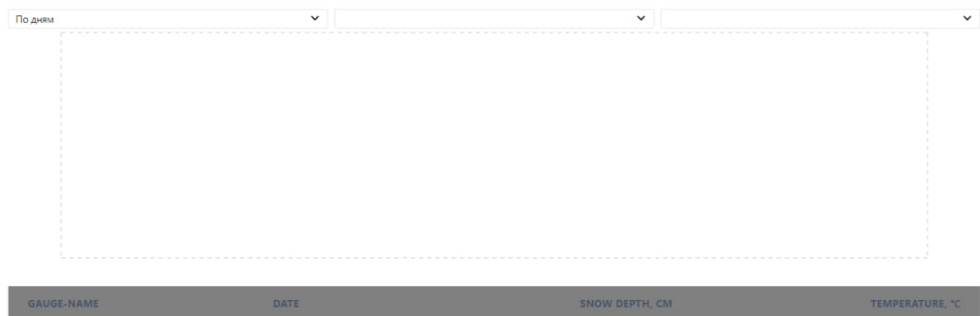


ГП-2

1 пачек
09.10.2022-09.03.2023

Загрузим ещё несколько пачек и нажмём “Создать отчёт”

- ☐ ГП-2
- ☐ 09.10.2022-09.03.2023
- ☐ 31.03.2023-08.05.2023
- ☐ CALM
- ☐ 02.10.2022-20.05.2023



Скачать отчёт

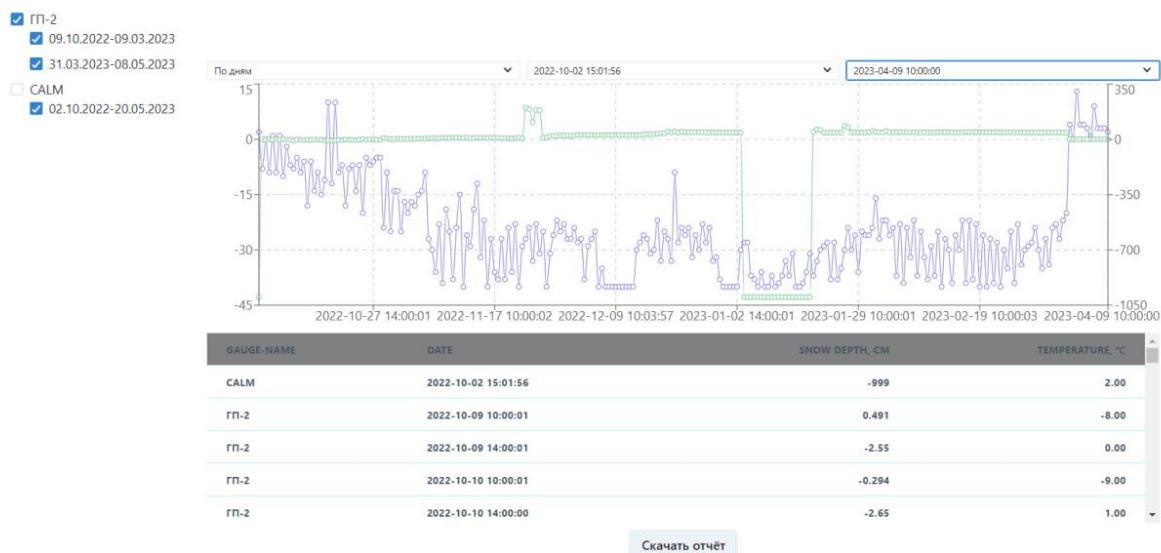
Тут отображаются только пачки со статусом “accepted” (задаётся при нажатии “Принять”)

Можно выбрать пачку или весь объект, немного подождать и появятся графики

- ☐ ГП-2
- ☒ 09.10.2022-09.03.2023
- ☐ 31.03.2023-08.05.2023
- ☐ CALM
- ☒ 02.10.2022-20.05.2023



Скачать отчёт



Если случайно нажал не на ту кнопку при выборе статуса отчёта последней пачки, можно зайти на localhost:8000 и исправить вручную:

```

    "id": "temp",
    "type": "celsius",
    "pos": [
      "3215",
      "2951",
      "3392",
      "3026"
    ]
  },
  {
    "id": "type",
    "type": "str",
    "pos": [
      "1415",
      "2951",
      "1645",
      "3010"
    ]
  }
],
"rejected",
"little batch"
]

```

PUT /batch/status Put Batch Status

Parameters

Cancel

Name	Description
status * required (query)	accepted

Execute

```
    ],  
    },  
    {  
      "id": "type",  
      "type": "str",  
      "pos": [  
        "1415",  
        "2951",  
        "1645",  
        "3010"  
      ]  
    }  
  ],  
  "accepted",  
  "little batch"  
],  
]
```

- ☐ ГП-2
 - ☐ 09.10.2022-09.03.2023
 - ☐ 31.03.2023-08.05.2023
 - ☐ 01.11.2022-08.01.2023
- ☐ CALM
 - ☐ 02.10.2022-20.05.2023

По дням

GAUGE-NAME	DATE	SNOW DEPTH, CM	TEMPERATURE, °C
------------	------	----------------	-----------------

Скачать отчёт

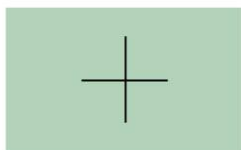
В отчёте появилась эта пачка

На странице объекта наблюдения можно выделить и удалить пачки

CALM

keep calm

Удалить



01.01.1900-01.01.1900

2023-09-05T12:58:09.391579

01.01.1900-01.01.1900

2023-09-05T13:15:50.839803

Назад

CALM

keep calm

Удалить



01.01.1900-01.01.1900

2023-09-05T12:58:09.391579

01.01.1900-01.01.1900

2023-09-05T13:15:50.839803

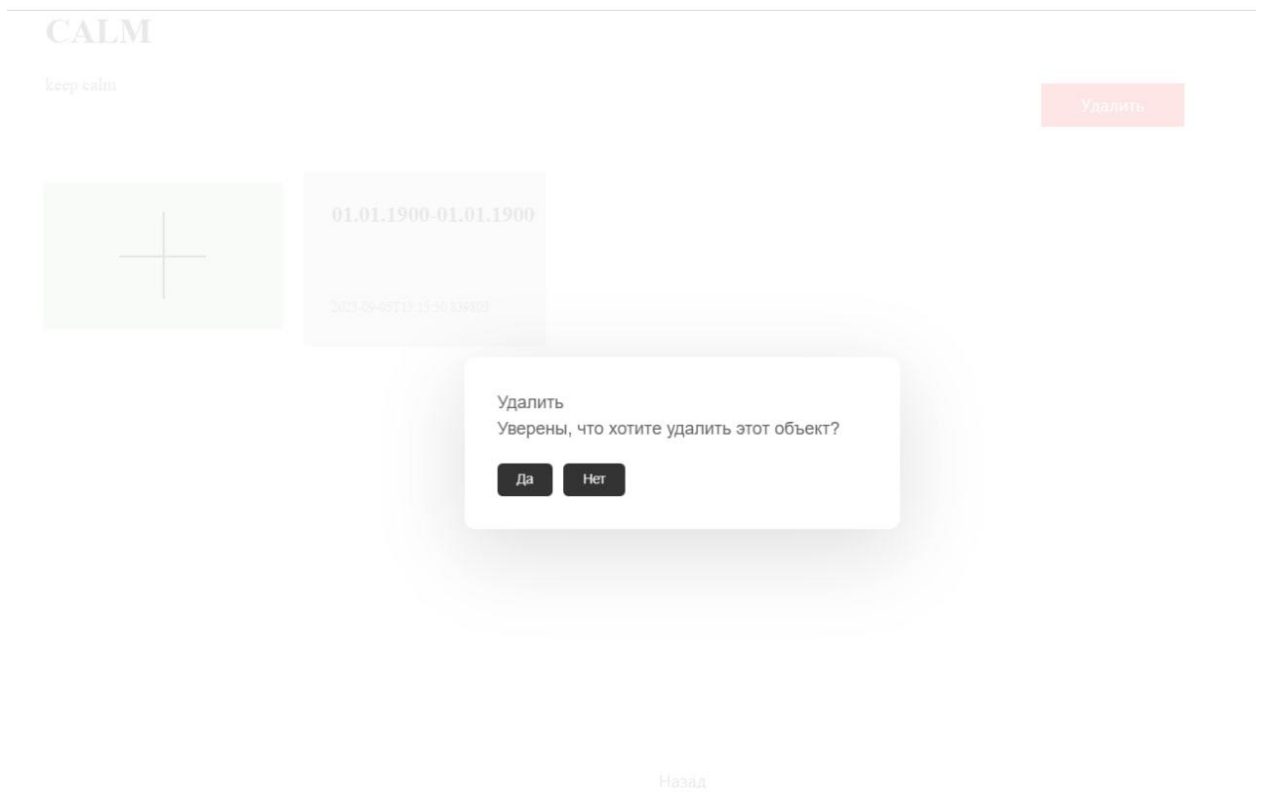
Удалить
Уверены, что хотите удалить эти пачки?

Да

Нет

Назад

Если ничего не выделять и нажать “Удалить”, то удаляется объект



Работа с Docker

Остановить работу приложения:

```
docker compose stop
```

Возобновить работу приложения:

```
docker compose start
```

Хранить базу данных в докер-контейнере без бекапа – плохая идея, поэтому лучше периодически копировать её на основной хост.

Посмотреть id контейнеров:


```
docker ps
```

Копируем id в строчке с postgres

Сохранить БД в файл `init.sql` в папке `bd`:

```
docker exec -t 1091f023250e pg_dumpall -c -U lucky > db/init.sql
```

```
PS C:\Users\Anna\Desktop\GitHub\snow_level_report> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
b59964f3868f   snow_level_report-frontend         "docker-entrypoint.s..." 35 minutes ago Up 5 minutes   0.0.0.0:3000->3000/tcp          snow_level_report-frontend-1
a8bdc65628f    snow_level_report-worker           "celery -A image_pro..." 35 minutes ago Up 5 minutes                   snow_level_report-worker-1
b70c21a3bbc5   snow_level_report-backend         "uvicorn main:app --..." 35 minutes ago Up 5 minutes   0.0.0.0:8000->8000/tcp          snow_level_report-backend-1
d1715e8eb87e   snow_level_report-server          "python3 -m http.ser..." 35 minutes ago Up 5 minutes   0.0.0.0:8080->8080/tcp          snow_level_report-server-1
b3123cf1eb75   redis:latest                       "docker-entrypoint.s..." 35 minutes ago Up 5 minutes   6379/tcp                        snow_level_report-redis-1
1f3f272f0f56   caddy:latest                       "caddy run --config ..." 35 minutes ago Up 5 minutes   0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp, 0.0.0.0:443->443/udp, 2019/tcp  snow_level_report-proxy-1
1091f023250e   postgres:14                        "docker-entrypoint.s..." 35 minutes ago Up 5 minutes   0.0.0.0:5432->5432/tcp          snow_level_report-db-1
PS C:\Users\Anna\Desktop\GitHub\snow_level_report> docker exec -t 1091f023250e pg_dumpall -c -U lucky > db/init.sql
```


GitHub > snow_level_report > db			
Имя	Дата изменения	Тип	Размер
 init.sql	06.09.2023 0:20	Файл "SQL"	33 КБ

Если удалить все docker-контейнеры, через этот файл можно будет восстановить БД, а также работать с ней в СУБД вручную. Изначально в db лежит файл init.sql, который создаёт пустые таблицы.

Чтобы удалить программу, нужно удалить все docker-контейнеры:

`docker system prune -a`

Также необходимо зайти в десктопное приложение и почистить память:

