

АВТОМАТИЗИРОВАННОЕ ФОРМАТИРОВАНИЕ ИСХОДНЫХ ДАННЫХ ОТ СЛУЖБЫ ГЕОЛОГИИ И РАЗРАБОТКИ



КОМБАРОВ С.В.
ЛЕТО 2024 г.

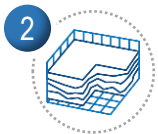


СОДЕРЖАНИЕ ПРЕЗЕНТАЦИИ



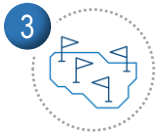
Введение

- Область применения проекта



Специфика предметной области

- Особенности подземной части
- Особенности наземной части
- Корректировки и актуализации



Цели и задачи проекта

- Ключевые запросы
- Описание функционала прототипа



Подходы к реализации прототипа

- Применяемые инструменты и модули
- Алгоритмы используемые в прототипе
- Планируемый результат

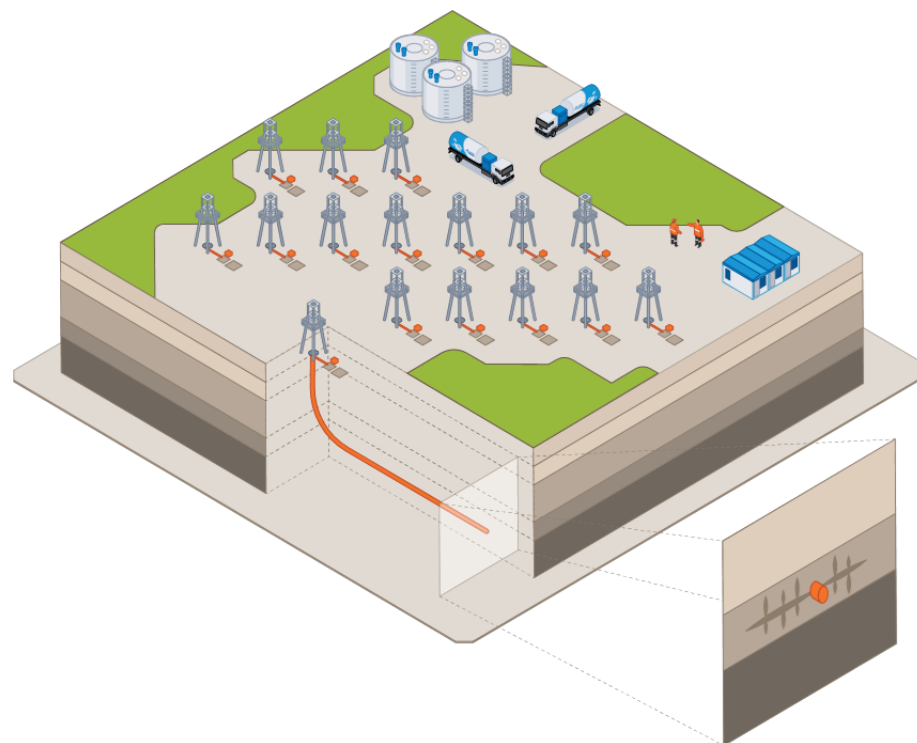


Дорожная карта

- Текущий статус проработки прототипа
- Ключевые вехи



Выводы



ПРЕДМЕТНАЯ ОБЛАСТЬ РАССМАТРИВАЕМОГО ПРОЕКТА

РАЗРАБОТКА И ЭКСПЛУАТАЦИЯ МЕСТОРОЖДЕНИЙ УГЛЕВОДОРОДОВ



КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ

предпроектная кроссфункциональная интеллектуальная деятельность, предшествующая принятию решения об инвестировании проекта

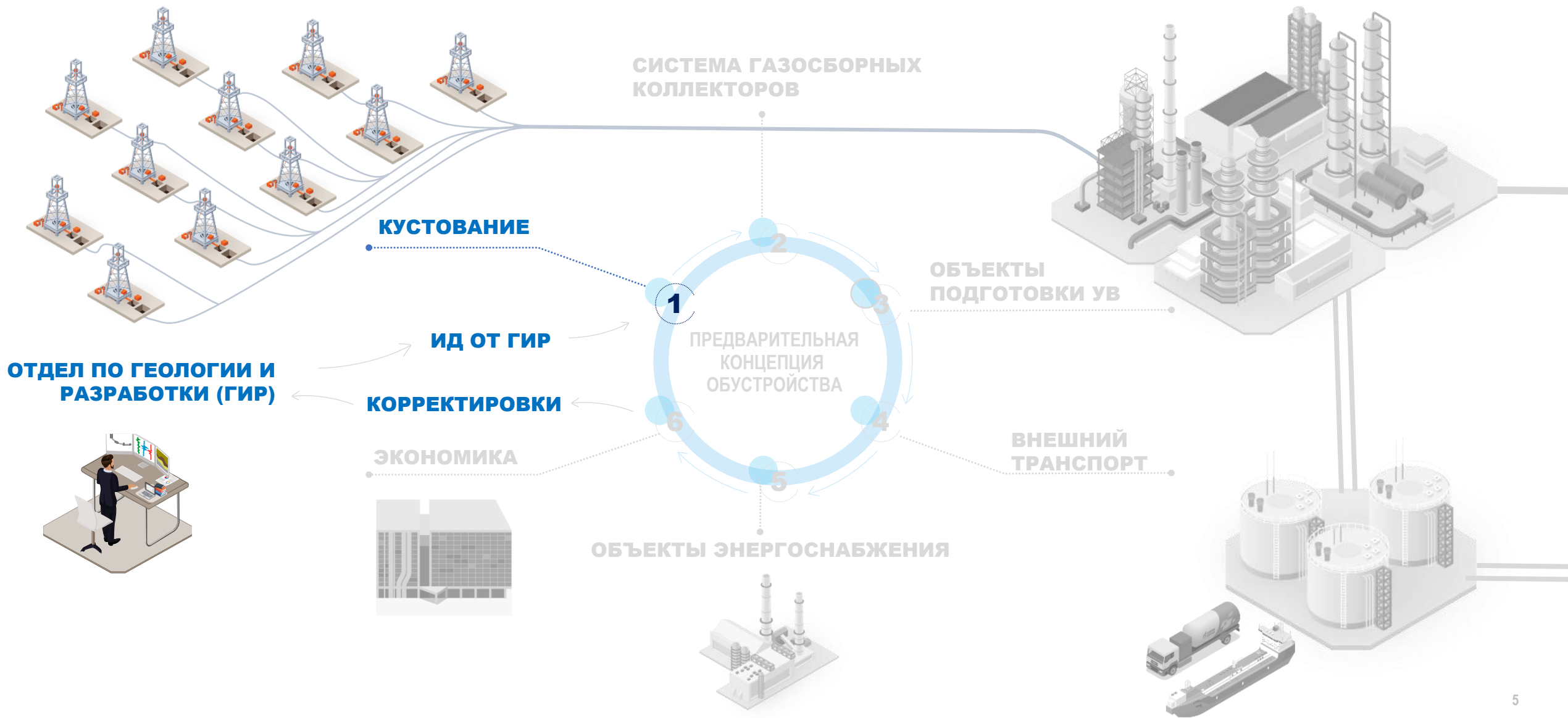
ПРЕДМЕТНАЯ ОБЛАСТЬ РАССМАТРИВАЕМОГО ПРОЕКТА

ОБУСТРОЙСТВО ГАЗОКОНДЕНАТНЫХ МЕСТОРОЖДЕНИЙ



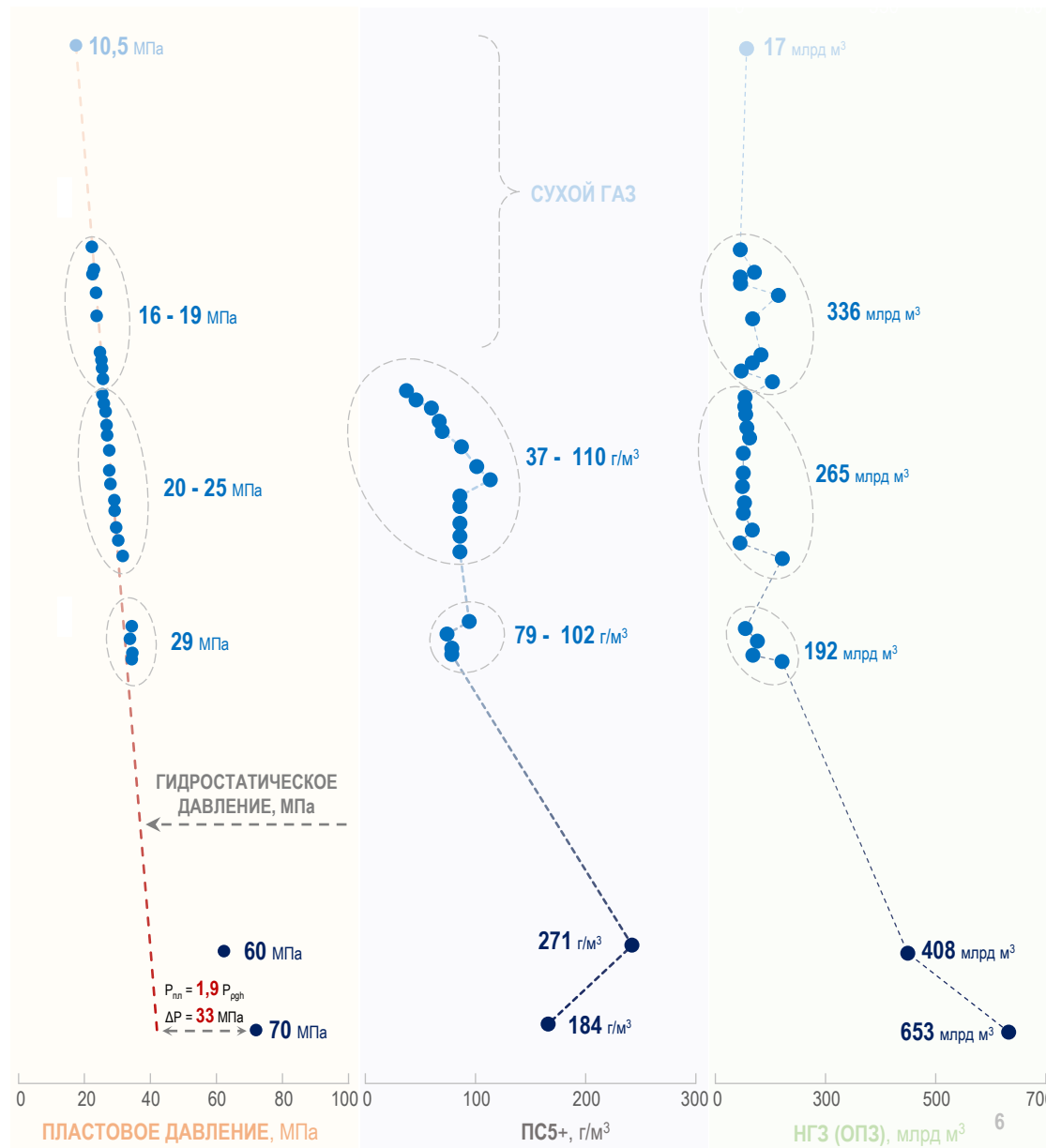
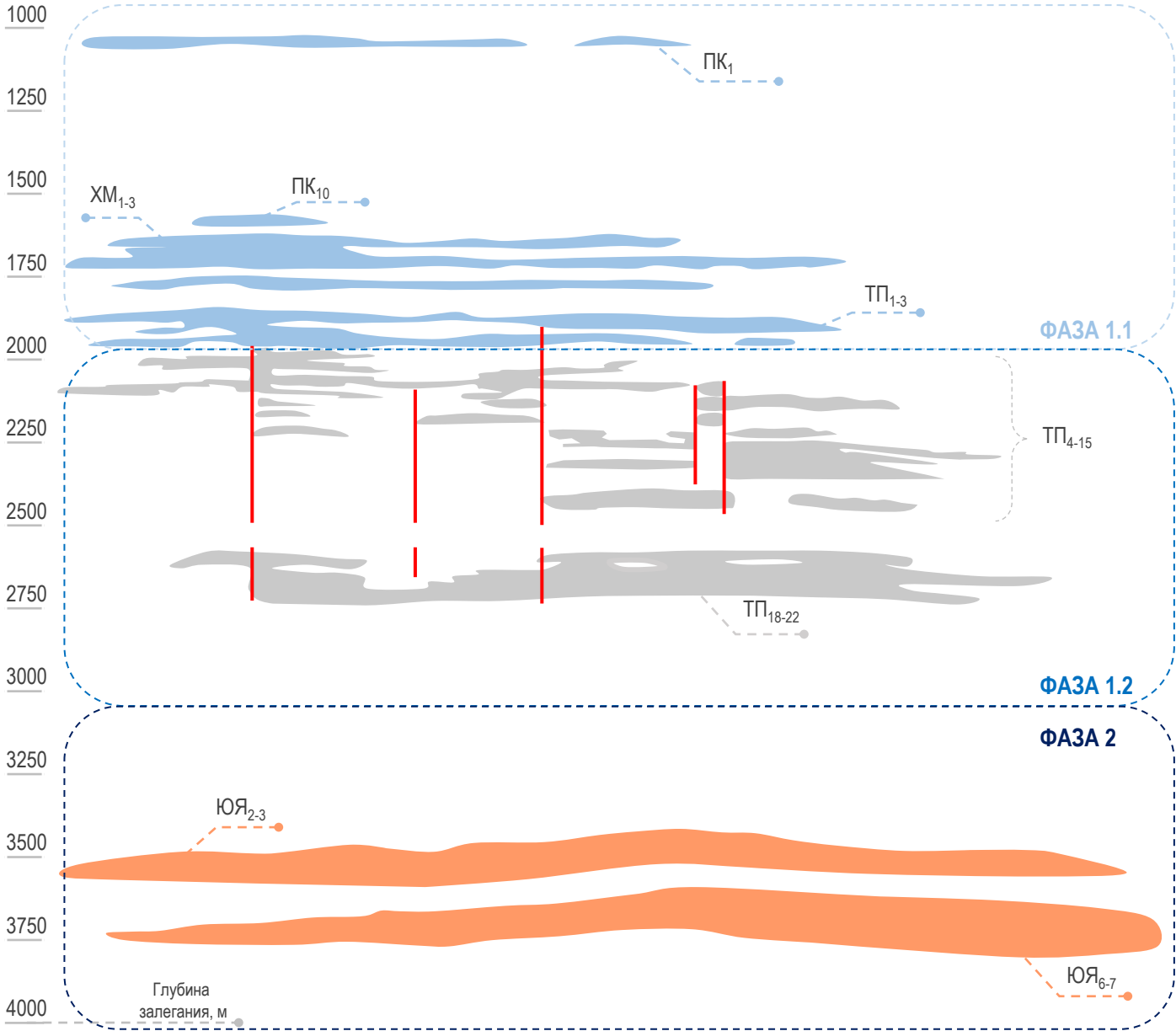
ПРЕДМЕТНАЯ ОБЛАСТЬ РАССМАТРИВАЕМОГО ПРОЕКТА

ОБУСТРОЙСТВО ГАЗОКОНДЕНАТНЫХ МЕСТОРОЖДЕНИЙ



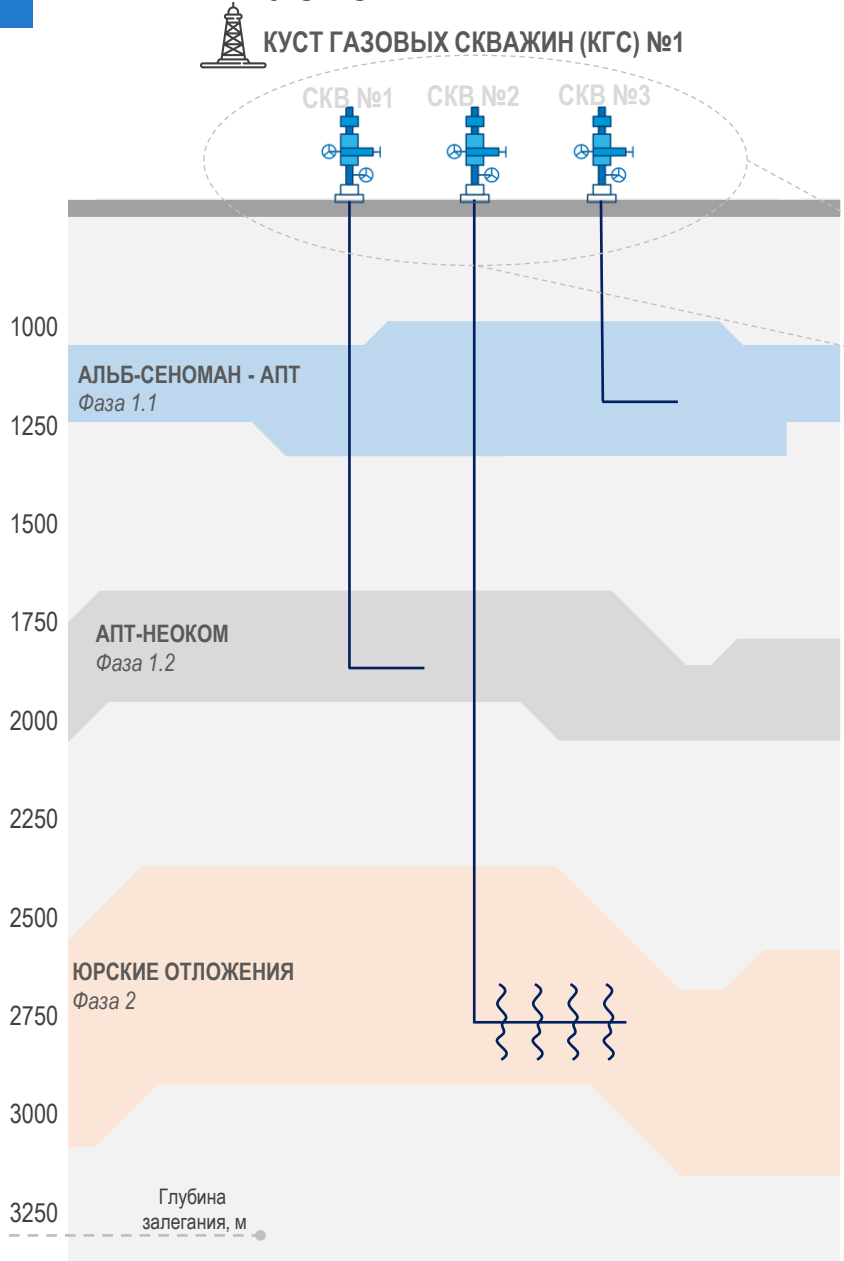
ОСОБЕННОСТИ ИСХОДНЫХ ДАННЫХ ПО ПОДЗЕМНОЙ ЧАСТИ

ПРИМЕР ФАЗИРОВАНИЯ ПРОЕКТА В ГЕОЛОГИЧЕСКИХ ПРЕДПОСЫЛКАХ

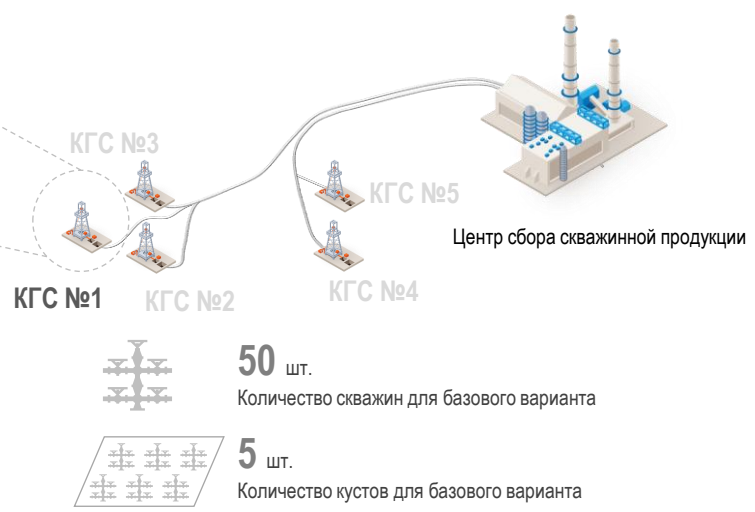


ОСОБЕННОСТИ ИСХОДНЫХ ДАННЫХ ПО НАЗЕМНОЙ ЧАСТИ

ПРИМЕР КУСТОВАНИЯ



БАЗОВЫЙ ВАРИАНТ



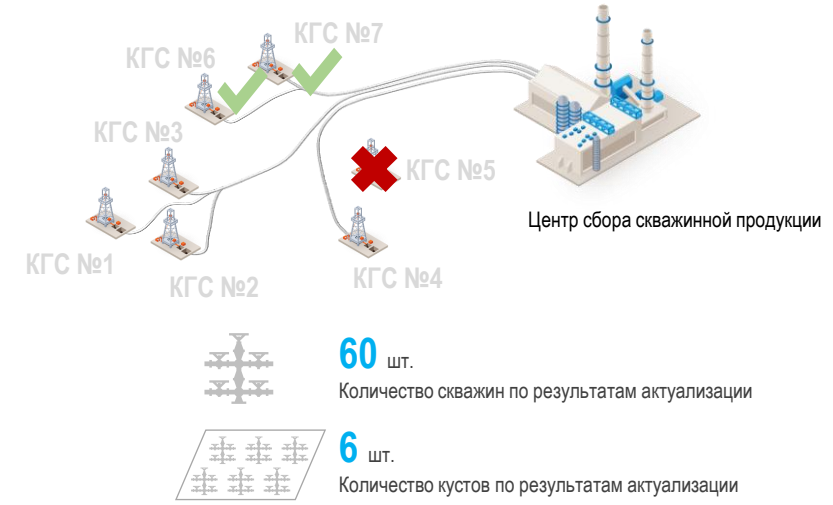
КЛЮЧЕВЫЕ ИЗМЕНЕНИЯ ПО РЕЗУЛЬТАТАМ АКТУАЛИЗАЦИИ

- Исключение КГС №5 из профиля добычи
- Добавлены КГС №6 и КГС №7
- На КГС №1-№3 увеличены темпы отбора по скважинам

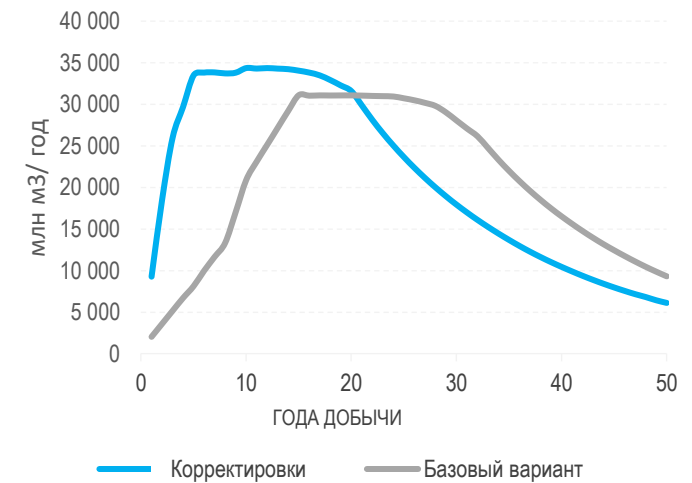
КОРРЕКТИРОВКИ ПО ИСХОДНЫМ ДАННЫМ

- КГС №1-3 – обновить профили по добыче
- КГС №4 – без изменений
- КГС №5 – исключить из расчёта
- КГС №6-7 – добавить в расчёт

КОРРЕКТИРОВКИ ПО БАЗОВОМУ ВАРИАНТУ



СРАВНЕНИЕ ПРОФИЛЕЙ ДОБЫЧИ ДО И ПОСЛЕ КОРРЕКТИРОВОК



ЦЕЛИ И ЗАДАЧА ПРОЕКТА

КЛЮЧЕВЫЕ ВЫЗОВЫ И ПРОБЛЕМАТИКА



ПРОБЛЕМА

Повышенные трудозатраты на ручное формирование свода поквартальной добычи на основании большего объёма исходных данных (профилей скважинной добычи) переданных от службы Геологии и Разработки (ГИР)



ВЫЗОВ

Повышение оперативности форматирования и минимизация ошибок свода исходных данных



ЦЕЛЬ ПРОЕКТА:

Автоматизация процесса формирования сводных таблиц на основании исходных данных для дальнейшего применения их в расчётах, а также аналитики, минимизация человеческого фактора



ЗАДАЧИ:

- **Определение требований и формата для аналитики** – на основании необходимой производственной потребности
- **Проектирование концепции работы прототипа** – определение используемых модулей и взаимодействия между ними через Python, создание алгоритмов обработки данных, их вывода и хранения
- **Реализация программного кода** – разработка скриптов
- **Проверка работы прототипа на текущих проектах** – верификация работы инструмента, отладка

ПРИМЕНЯЕМЫЕ ИНСТРУМЕНТЫ И МОДУЛИ

КОНЦЕПЦИЯ ВЗАИМОДЕЙСТВИЯ МОДУЛЕЙ И БИБЛИОТЕК В PYTHON

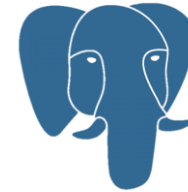
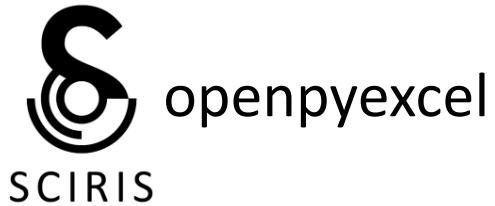
ИД

Формирование массивов

Обработка

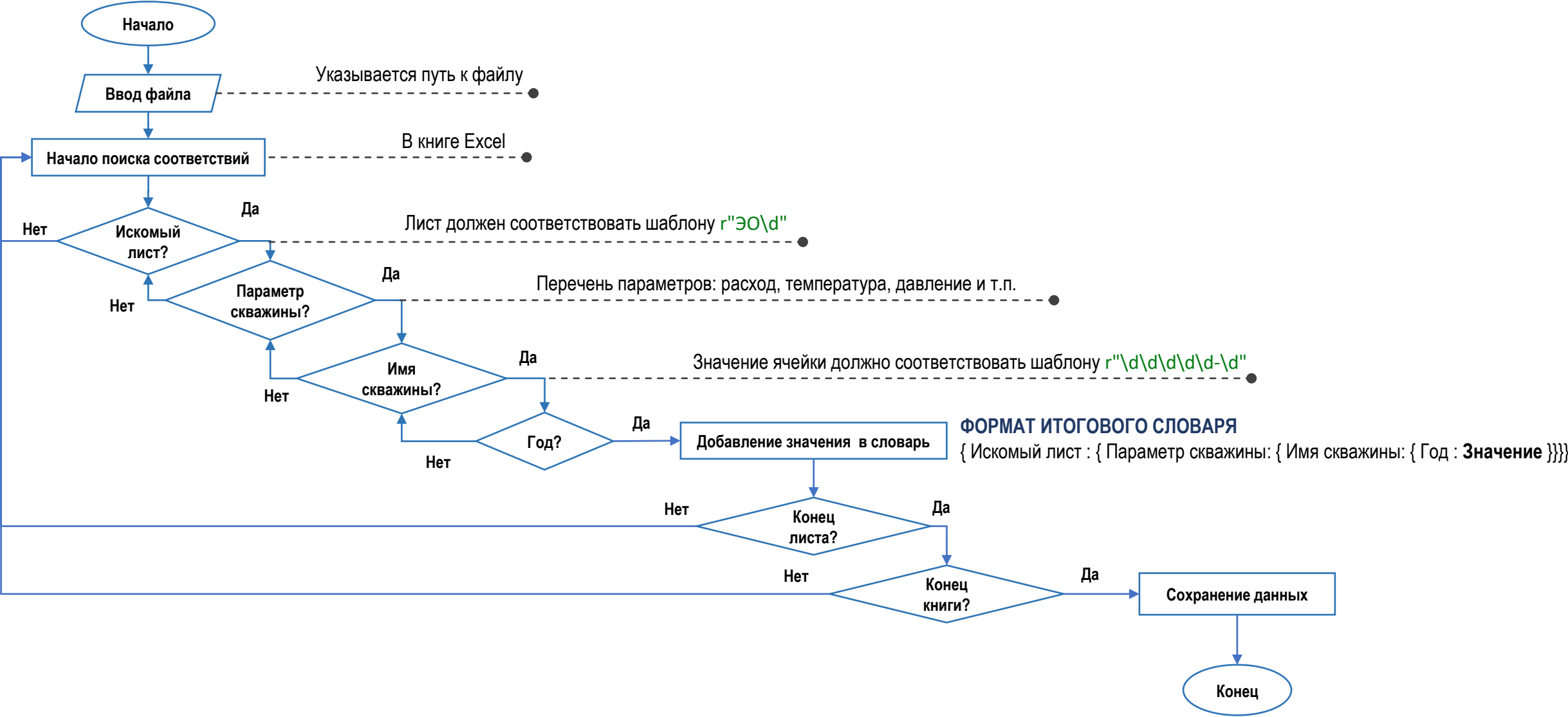
Вывод для аналитики

Хранение результатов обработки



АЛГОРИТМ ДЛЯ ПАРСИНГА ИСХОДНЫХ ДАННЫХ

ОПРЕДЕЛЕНИЕ ГРАНИЦ ЦИКЛА



ПРИМЕР РЕАЛИЗАЦИИ КОДА ДЛЯ ПАРСИНГА

СТРУКТУРА ЦИКЛА

ЦИКЛ ДЛЯ ПАРСИНГА

```

sn = wb.sheetnames
for j in sn:
    # Цикл для пробежки по листам книги
    if len(j) <= 3:
        wb.active = wb[j]
        ws = wb.active
        eo = j
        for i in range(ws.max_row):
            # Определение местоположения параметра
            if type(ws.cell(row=i + 1, column=1).value) == str:
                row_start = ws.cell(row=i + 1, column=1).row
                column_start = ws.cell(row=i + 1, column=1).column
                x = re.findall(r"(\S+)", ws.cell(row=i + 1, column=1).value)
                # Костыль
                parametr = 0
                for kostil in x:
                    parametr = kostil
            # Определение имен скважин
            for m in range(ws.max_column):
                if ws.cell(row=i + 2, column=m + 1).value != None and ws.cell(row=i + 2, column=m + 1).value != "Dates":
                    y = re.findall(r"d\d\d\d\d\d", ws.cell(row=i + 2, column=m + 1).value)
                    # Костыль
                    well_name = 0
                    for kostil in y:
                        well_name = kostil
                    # Разделение имени скважины на составные части - ЛУ, КГС, номер и ЭО
                    ly = str(well_name)[0]
                    kgs = str(well_name)[1:3]
                    well_number = str(well_name)[3:5]
                    eks_ob = str(well_name)[-1]
                    # Определение конкретной даты
                    for mm in range(ws.max_row):
                        if type(ws.cell(row=mm + 2, column=1).value) == int and int(str(ws.cell(row=mm + 2, column=2).value)[:4]) < 2150 and ws.cell(row=mm + 2, column=1).value != 0:
                            date = str(ws.cell(row=mm + 2, column=2).value)[:4]
                            dct[(parametr, (ly, kgs, well_number, eks_ob), date)] = dct.get((parametr, (ly, kgs, well_number, eks_ob), date), ws.cell(row=i + mm, column=m + 1).value)

# Запись результатов парсинга в файл, формирование ИД для датафреймов
lst = []
lst_pd = []
file_output_lite = open('output_lite.txt', 'w')
file_output_full = open('output_full.txt', 'w')
for k, v in dct.items():
    file_output_lite.writelines(f'{k}, {v}\n')
    file_output_full.writelines(f'[{k[0]}, {k[1][0]}, {k[1][1]}, {k[1][2]}, {k[1][3]}, {k[2]}, {v}]\n')
    lst.append([k[0], k[1], k[2], v])
    lst_pd.append([k[0], k[1][0], k[1][1], k[1][2], k[1][3], k[2], v])
file_output_lite.close()
file_output_full.close()

```

СОХРАНЕНИЕ РЕЗУЛЬТАТОВ ПРОЦЕДУРЫ

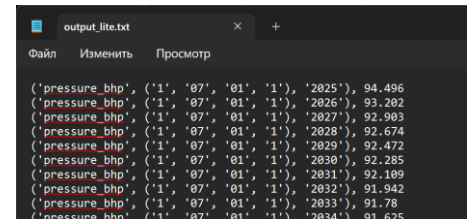
ИМПОРТ БИБЛИОТЕК

```

import re
import datetime
import pandas as pd
from openpyxl import load_workbook
start = datetime.datetime.now()
wb = load_workbook("Путь к файлу", data_only=True)
ws = wb.active
dct = {} # Словарь который ляжет в основу датафрейма
print('Начало выполнение процедуры парсинга')

```

ПРИМЕР КОНЕЧНОГО ФАЙЛА



```

('pressure_bhp', ('1', '07', '01', '1'), '2025'), 94.496
('pressure_bhp', ('1', '07', '01', '1'), '2026'), 93.202
('pressure_bhp', ('1', '07', '01', '1'), '2027'), 92.903
('pressure_bhp', ('1', '07', '01', '1'), '2028'), 92.674
('pressure_bhp', ('1', '07', '01', '1'), '2029'), 92.472
('pressure_bhp', ('1', '07', '01', '1'), '2030'), 92.285
('pressure_bhp', ('1', '07', '01', '1'), '2031'), 92.109
('pressure_bhp', ('1', '07', '01', '1'), '2032'), 91.942
('pressure_bhp', ('1', '07', '01', '1'), '2033'), 91.78
('pressure_bhp', ('1', '07', '01', '1'), '2034'), 91.625

```

ОБРАБОТКА ПОЛУЧЕННОГО МАССИВА ДАННЫХ

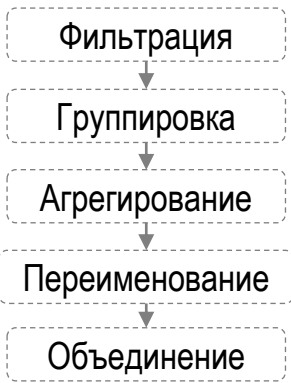
ФОРМИРОВАНИЕ КОНЕЧНОГО DATAFRAME ПОД ПРОИЗВОДСТВЕННУЮ ЗАДАЧУ

ИСХОДНЫЙ МАССИВ

	Параметр	Номер ЛУ	Номер КГС	...	Номер ЭО	Год	Значение
0	pressure_bhp	3	01	...	1	2029	0.0
1	pressure_bhp	3	01	...	1	2030	0.0
2	pressure_bhp	3	01	...	1	2031	0.0
3	pressure_bhp	3	01	...	1	2032	0.0
4	pressure_bhp	3	01	...	1	2033	0.0
...
29299	work_days_for_well	3	02	...	6	2083	0.0
29300	work_days_for_well	3	02	...	6	2084	0.0
29301	work_days_for_well	3	02	...	6	2085	0.0
29302	work_days_for_well	3	02	...	6	2086	0.0
29303	work_days_for_well	3	02	...	6	2087	0.0

[29304 rows x 7 columns]

ОБРАБОТКА

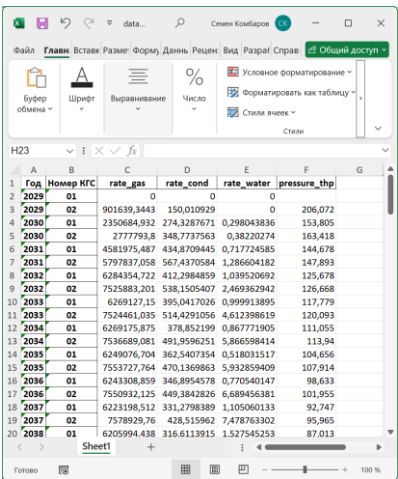


ИТОГОВЫЙ DATAFRAME

	Год	Номер КГС	rate_gas	rate_cond	rate_water	pressure_thp
2029	01	0.000000e+00	0.000000	0.000000	NaN	
	02	9.016393e+05	150.010929	0.000000	206.072	
2030	01	2.350685e+06	274.328767	0.298044	153.805	
	02	2.777794e+06	348.773756	0.382203	163.418	
2031	01	4.581975e+06	434.870944	0.717725	144.678	
...	
2085	02	4.601668e+05	34.406097	0.000000	7.569	
2086	01	7.144476e+05	33.935112	10.947487	7.698	
	02	4.210526e+05	32.547945	0.000000	7.524	
2087	01	6.802819e+05	33.441961	11.520327	7.608	
	02	3.720260e+05	27.896179	0.000000	7.453	

[118 rows x 4 columns]

ПРИМЕР КОНЕЧНОГО ФАЙЛА



#Формирование полного датафрейма

```
df = pd.DataFrame(lst, columns=['Параметр', 'Скважина', 'Год', 'Значение']) # Оптимизированная версия датафрейма
```

```
df_pd = pd.DataFrame(lst_pd, columns=['Параметр', 'Номер ЛУ', 'Номер КГС', 'Номер скважины', 'Номер ЭО', 'Год', 'Значение']) # Полная версия
```

#Группировка и фильтрация по датафреймам

```
wellpads_sum_gas_rate = df_pd[df_pd.Параметр == "rate_gas"].groupby(['Параметр', 'Номер КГС', 'Год']).agg({"Значение": 'sum'}).rename(columns = {"Значение": "rate_gas"})
```

```
wellpads_sum_gas_rate.to_csv('wellpads_sum_gas_rate.txt', sep='\t', index = True)
```

```
wellpads_min_thp = df_pd[df_pd.Параметр == "pressure_thp"].replace(0, None).groupby(['Параметр', 'Номер КГС', 'Год']).agg({"Значение": 'min'}).rename(columns = {"Значение": "pressure_thp"})
```

```
wellpads_min_thp.to_csv('wellpads_min_thp.txt', sep='\t', index = True)
```

```
wellpads_sum_condensate_rate = df_pd[df_pd.Параметр == "rate_cond"].groupby(['Параметр', 'Номер КГС', 'Год']).agg({"Значение": 'sum'}).rename(columns = {"Значение": "rate_cond"})
```

```
wellpads_sum_condensate_rate.to_csv('wellpads_sum_condensate_rate.txt', sep='\t', index = True)
```

```
wellpads_sum_water_rate = df_pd[df_pd.Параметр == "rate_water"].groupby(['Параметр', 'Номер КГС', 'Год']).agg({"Значение": 'sum'}).rename(columns = {"Значение": "rate_water"})
```

```
wellpads_sum_water_rate.to_csv('wellpads_sum_water_rate.txt', sep='\t', index = True)
```

```
ly_sum_gas_rate = (df_pd[df_pd.Параметр == "rate_gas"].groupby(['Год']).agg({"Значение": 'sum'})).max().reset_index()
```

```
ly_sum_cond_rate = (df_pd[df_pd.Параметр == "rate_cond"].groupby(['Год']).agg({"Значение": 'sum'})).max().reset_index()
```

#Merge для группы датафреймов

```
data = pd.merge(wellpads_sum_gas_rate, wellpads_sum_condensate_rate, how= "outer", on=["Год", "Номер КГС"])
```

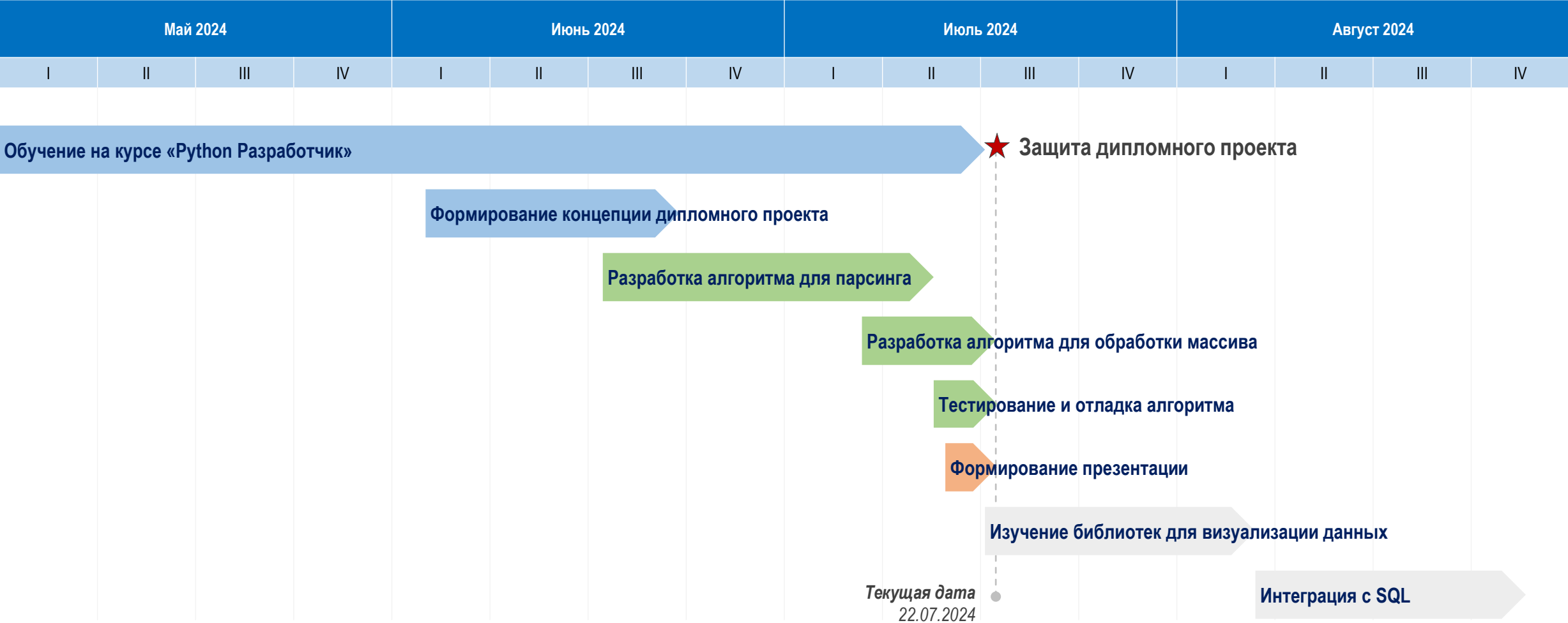
```
data = pd.merge(data, wellpads_sum_water_rate, how= "outer", on=["Год", "Номер КГС"])
```

```
data = pd.merge(data, wellpads_min_thp, how= "outer", on=["Год", "Номер КГС"])
```

```
data.to_csv('data.txt', sep='\t', index = True)
```

```
data.to_excel('data.xlsx', index = True, merge_cells=False)
```

ДОРОЖНАЯ КАРТА РЕАЛИЗАЦИИ ПРОТОТИПА



ВЫВОДЫ

По итогам обучения на курсе «Python разработчик» были освоены:

- Базовые навыки программирования
- Базовые функции и библиотеки Python
- Библиотеки openpyxl, pandas и т.п.
- Алгоритмические подходы работы с данными

Разработан и реализован:

- Алгоритм и программная процедура для обработки исходных данных от службы ГиР, потенциальное **сокращение трудозатрат** на формирование свода покустовой добычи приблизительно **с одного часа** реального времени **до нескольких секунд** на один массив исходных данных