

Рабочий лист №1

Дата "01" февраля 2025 г.  
(заполняется оргкомитетом)

Шифр ЖИ-31  
(заполняется оргкомитетом)

Оценка работы

(таблица заполняется по итогам проверки работы членами жюри олимпиады)

№ задания	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Итого (итоговый балл, подпись председателя жюри)
Балл																<u>87</u>
№ задания	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Балл																<u>93</u>

Магистры ИИ

(название олимпиады, заполняется участником)

Прикладная информатика

(профиль олимпиады, заполняется участником)

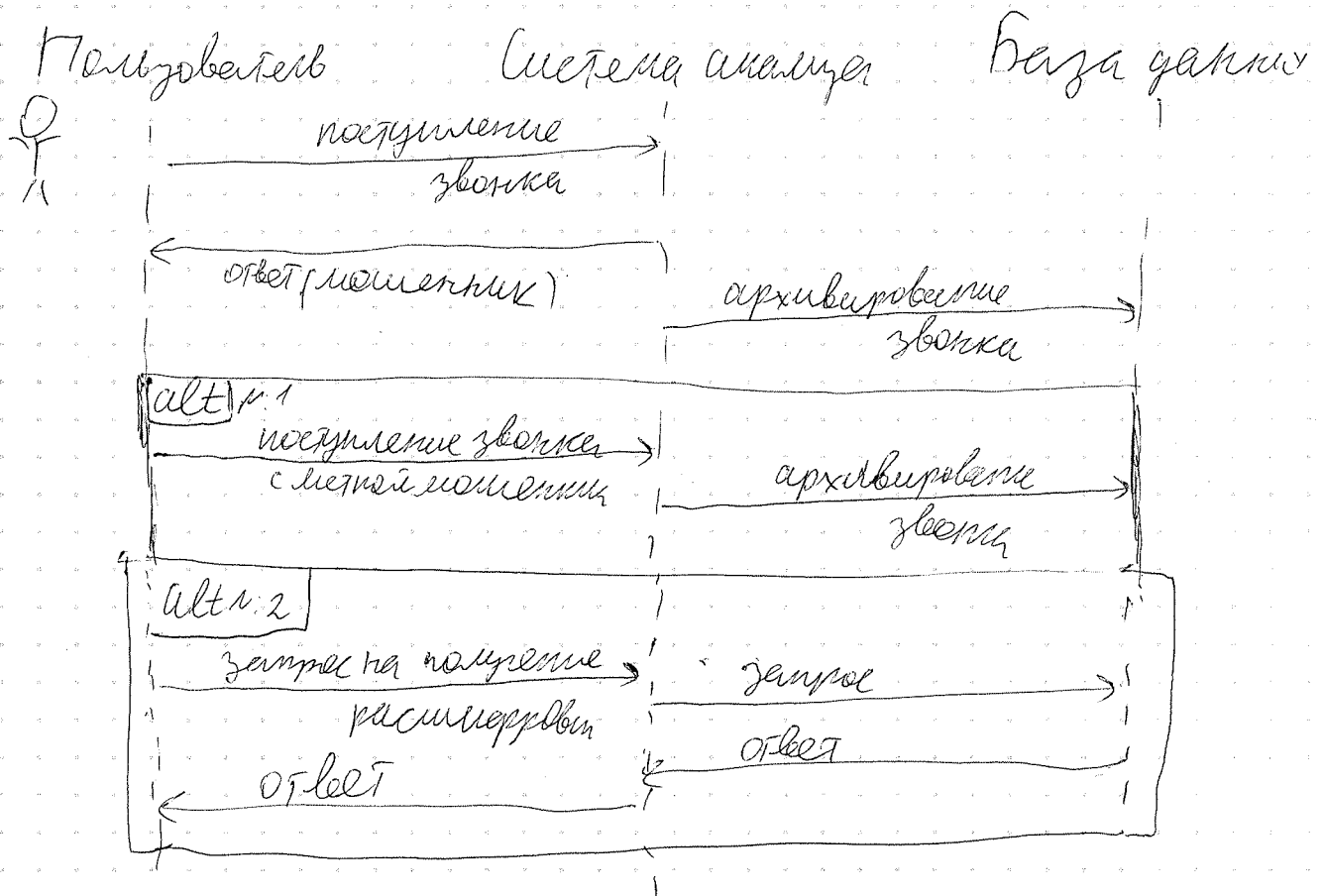
Задача: Проектирование архитектуры.

Пример работы.

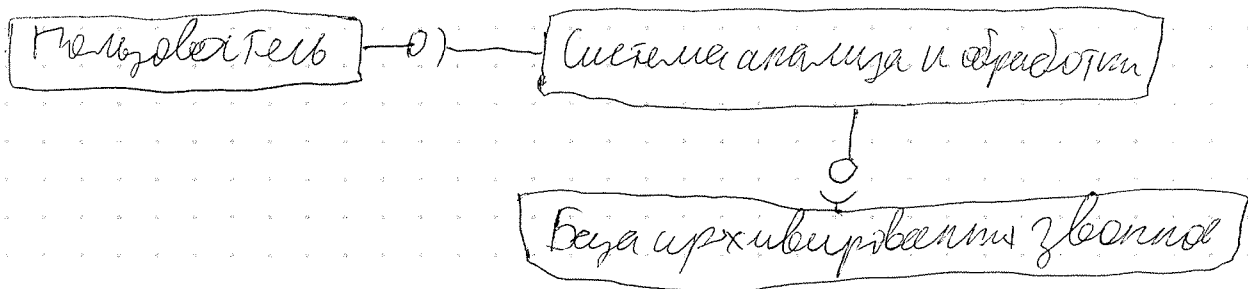
После совершенного телефонного звонка с телефона пользователя отправляется записанный звонок в систему анализа и обработки. В результате, если это мошенник система отправляет предупреждение на телефон пользователя. Пользователь также имеет возможность самостоятельно пометить мошенника при окончании звонка, если звонок с незнакомого номера. Если система решит, что звонок был от мошенников или пользователей сам это сообщил, то звонок архивируется. При запросе пользователя можно получить расшифровку звонка

Сдано листов: 3

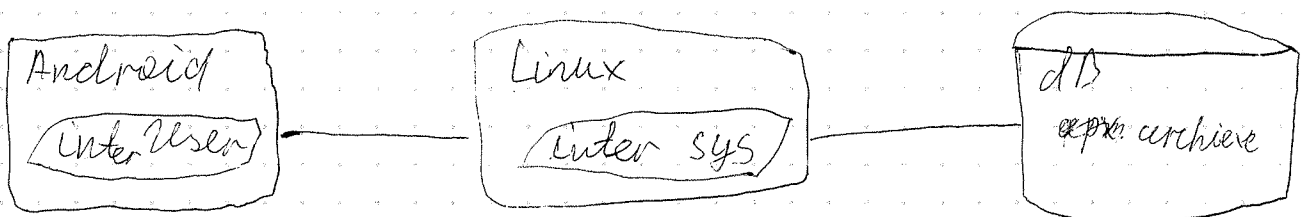
## Диаграмма последовательности



## Диаграмма компонентов



## Диаграмма развертывания

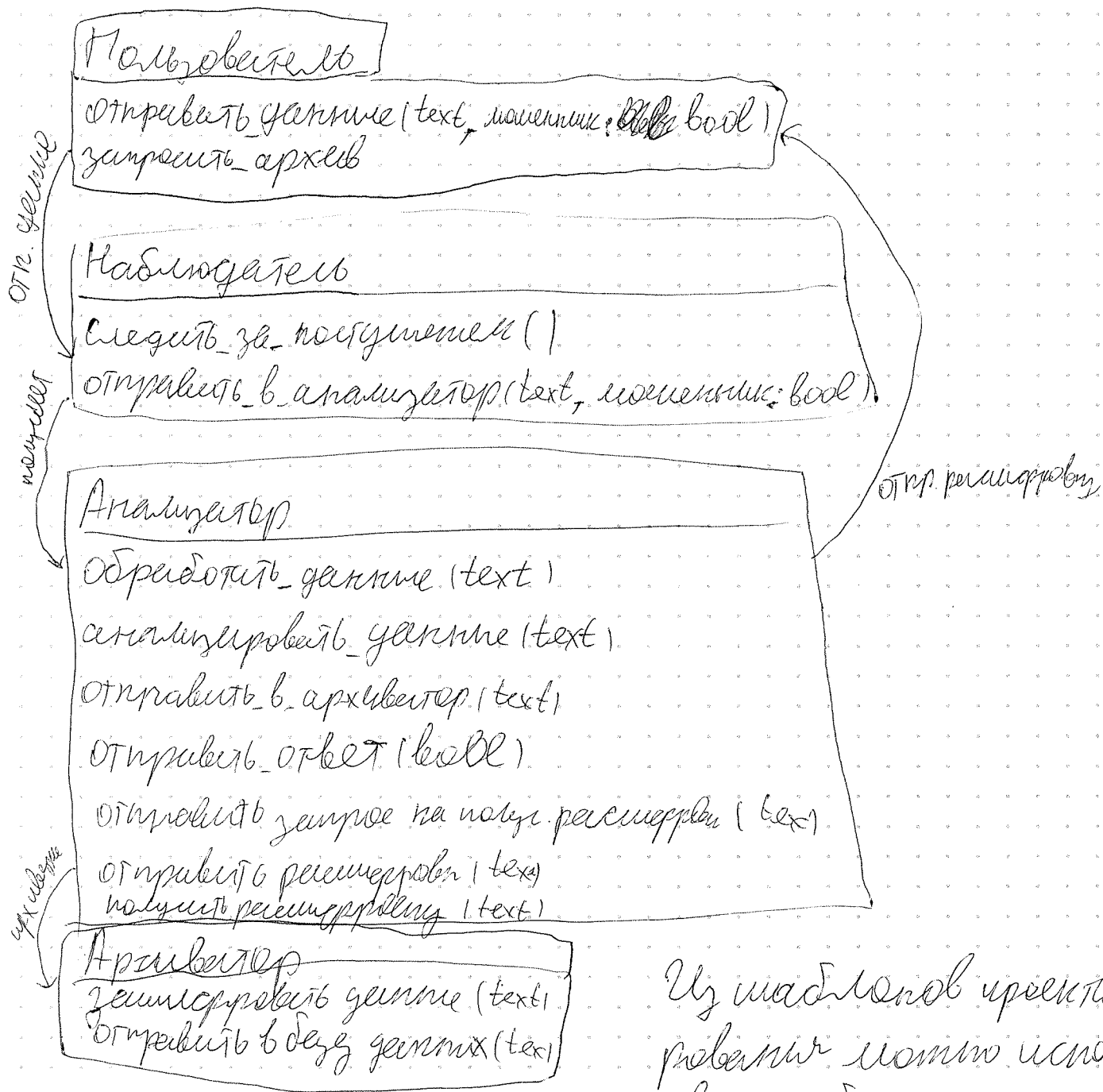


Дополнительный рабочий лист  
(без рабочего листа №1 недействителен)

Дата "21" февраля 2025 г.  
(заполняется участником)

Шифр ЛЧ-31  
(заполняется участником)

## Диаграмма классов



Из шаблонов проектирования можно использовать наблюдателя (observer) для класса наблюдателя

Задача 2. Для анализа выявление вероятности  
Телефонного мошенничества можно производить  
и следующие пункты.

- 1) При потере личной информации  
вероятность  $\pm 50\%$
- 2) при признаках номера  $\pm 25\%$
- 3) при словах "проги", "дешев", "скорее" и прочих  
похожих еще на  $30\%$

```
class Analyzer {  
    confident_data; patterns; numbers_from_book;  
    func analyse_from (number) {  
        if (number != numbers_from_book) {  
            return 25; } else return 0; }  
    func analyse_text (text) {  
        if (text == confident_data) {  
            return 50;  
        } else {  
            return 0; } }  
    func analyse_how (text) {  
        if (text == patterns) {  
            return 25;  
        } else {  
            return 0; } }  
}  
main() {  
    text_call; number;
```

Дополнительный рабочий лист  
(без рабочего листа №1 недействителен)

Дата "01" февраля 2025 г.  
(заполняется участником)

Шифр JCI-31  
(заполняется участником)

```
percent += Analyzer.analyse_from(number)
percent += Analyzer.analyse_what(numbertext)
percent += Analyzer.analyse_who(text)
if (percent > 50)
    print('Подозрительно!')
else
    print('Всё хорошо!')
```

Задача 3

В задаче также необходимо преобразовать из аббревиатур слова в инициалы и наоборот

```
import re
def convert(text):
    words_to_digits = {
        '0': 'ноль',
        '1': 'один',
        '2': 'два',
        '3': 'три',
        '4': 'четыре',
        '5': 'пять',
        '6': 'шесть',
        '7': 'семь',
        '8': 'восемь',
    }
```

'9': 'gebr70',

'+' : 'more' ]

words = text.split(' '); structured\_words = []

for word in words:

if (word == words\_in\_digit):

structured\_words += words\_in\_digit(words)

structured\_words += word

~~return structured\_words~~

return structured\_words

def find\_phone(text):

pattern = ('r 7|8/d{103}')

~~words = text.split(' ')~~

result = []

~~for word in words:~~

~~if~~

~~re.Find(r~~

result = re.Find(pattern, text)

~~return result~~

def find\_card(text)

pattern = ('r [0-9]/d{103}')

result = re.Find(pattern, text)

return result

~~main()~~

main():

call\_text = [ ... ]

text = convert(call\_text)

phones = find\_phone(text)

cards = find\_cards(text); print(cards, phones)