

Казанский (Приволжский) федеральный университет
Олимпиада для поступающих в магистратуру

Место печати

Рабочий лист №1

Дата "1" февраля 2025 г.
(заполняется оргкомитетом)

Шифр ПИ-25
(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																77
№ задания	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Балл																73

Магистрия Уи

(название олимпиады, заполняется участником)

Прикладная информатика

(профиль олимпиады, заполняется участником)

Задание 1.

Диаграмма компонентов

Телефонный разговор

Phone Call Service

↓ направление данных для анализа

Анализатор телефонного звонка

↓ отправка результатов анализа

Обработка результатов

↓ сохранение в БД

БД

Дизайн классов

```
class PhoneCall
```

```
phone-number-caller1: str  
phone-number-caller2: str  
phonecall-text: str  
phonecall-time:  
phonecall-otherdata  
phonecall-analysis-result: Proceed  
Result
```

```
class PhoneCallService
```

```
get-PhoneClass(): PhoneCall  
get-data-from-phonecall(PhoneCall)  
pass-data-for-analysis(PhoneCall)
```



```
class AnalysisPhoneCall
```

```
analyze-phone-number(PhoneCall)  
analyze-phonecall-text(PhoneCall)  
analyze-phonecall-otherdata(PhoneCall)
```



```
class ProceedResults
```

```
phone-number-analyze-results  
phonecall-text-analyze-results  
phonecall-otherdata-analyze-results  
phonecall-analyze-result
```

```
proceedresult(...): phonecall-analyze-result
```



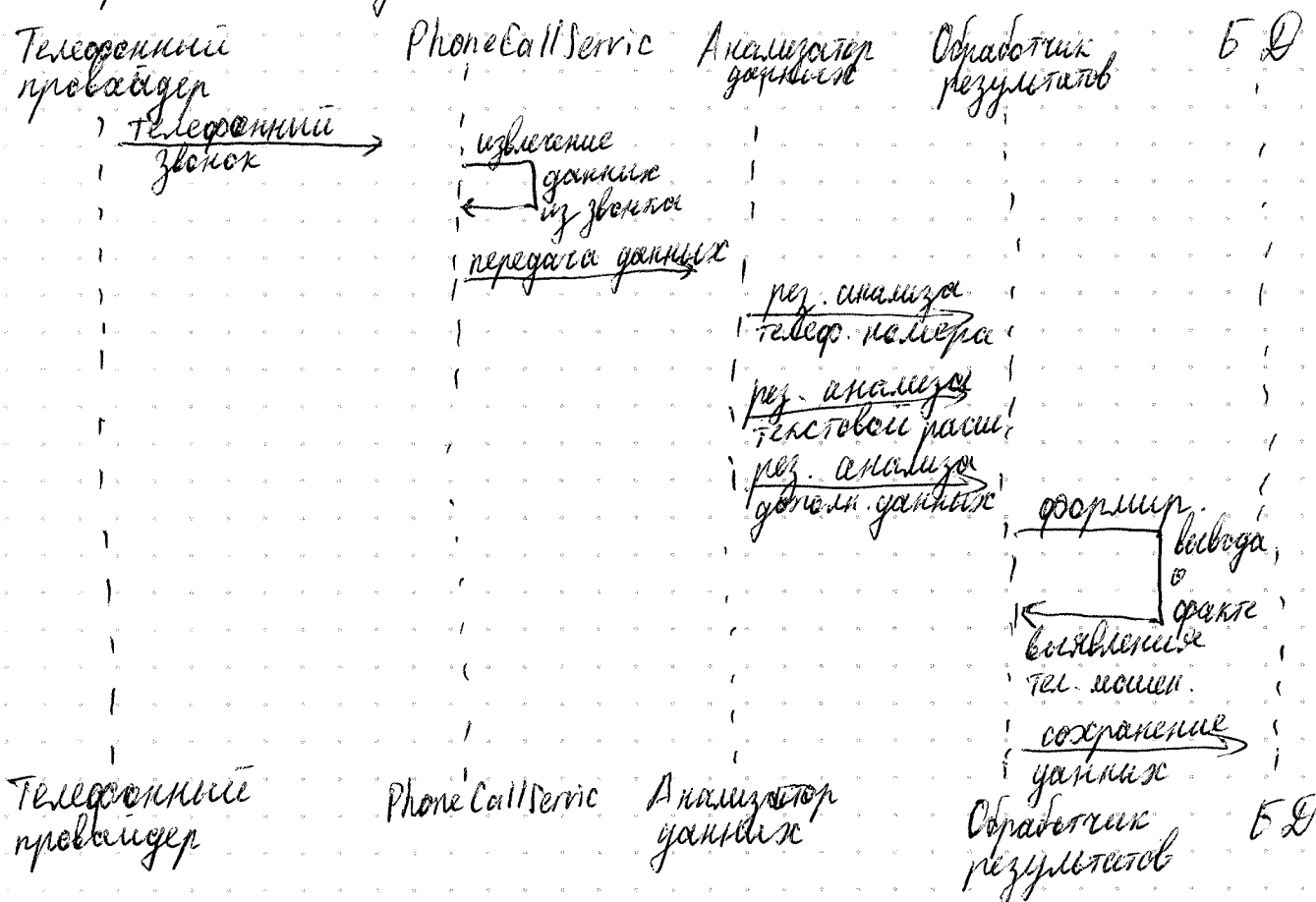
```
class ConnectDB
```

```
connect()  
updateDB(PhoneCall)
```

Дата "1" декабря 2025 г.
(заполняется участником)

Шифр ПА-25
(заполняется участником)

Диаграмма последовательности



Задание 2

```
text = '...' // текстовая расшифровка
patterns = '...' // методы выявл. телеф. мошенничества

res = proceed_text(text) // обработка текста
analyze_text(text, patterns) // ф-ия опр-ия телеф. мошен.
value = 0, 1 // вер-ть, с кот мы счит, что соверш. телеф. мошенничество
if (res >= true value):
    print('соверш. телеф. мошен. совершено')
else:
    print('не обнаружен факт телеф. мошен.')

```

- 1) сохраняем текст. расшифр.
- 2) опр. методы выявления телеф. мошенничества
- 3) используем ф-ию анализа текст. расшифр., предварительно обработав текст. расшифр.

```
def proceed_text(text):
    res = text.ToLower()
    return res

```

```
def analyze_text(text, patterns):
    if patterns in text:
        return 1
    else:
        return 0

```

- 4) опр. вер-ть, с которой мы считаем, что телеф. мошен. совершено
- 5) выводим результат.

Задание 3

Python:

```
import re

```

```
call_text_data = '...' // текстовая расшифровка

```

```
patterns = [r'[a-zA-Z._+-]+@[a-zA-Z]+[a-zA-Z]',

```

```
re.find r'[d14]{1,3}',

```

```
r'[d14]{1,3}+[d14]{1,3}']

```

Казанский (Приволжский) федеральный университет
Олимпиады школьников и студентов

Итого баллов

Дополнительный рабочий лист
(без рабочего листа №1 недействителен)

Дата " 1 " октября 20 25 г.
(заполняется участником)

Шифр МН-25
(заполняется участником)

```
for i in patterns:
    if re.find(text, i):
        if call_text_data.remove(re.find(text, i))
```