

Рабочий лист №1

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

Шифр ПЧ-16
(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																85
№ задания	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Балл																

Мастер Ум
(название олимпиады, заполняется участником)

Трибунальная информация
(профиль олимпиады, заполняется участником)

Задание 1.

Компоненты с-мы: Web-интерфейс (микро фронтенд, Flask, + HTML, CSS, JS, Jinja2 (машинизатор) для отображения интерфейса. Источники данных - база данных с записями звонков от менеджеров операторов, предоставляется по API. Для обработки естественного языка в текст предоставляется использование Yandex Speech. А для анализа текста nlp-модели и машинное обучение для распознавания смыслов и различных параметров(на). Также для корректного распознавания диалога с клиентом будет использоваться база данных с текстом вступлений.

Сдано 2 листа.

вариантами именительных падежей.
Для хранения данных будем использована MySQL + ORM с-ма SQL Alchemy для удобства работы с данными. В базе будем храниться номер звонящего, номер абонента, а также текстовая расшифровка диалога. Работа каждого модуля будет осуществляться в контейнере (микросервисная архитектура), для оркестрации будем использоваться managed kubernetes ^(yandex cloud), для более отказоустойчивой с-мы. При проектировании с-мы будет использована модель "водопад", чтобы быть уверенным во всех аспектах работы системы.

Задание 2.

1. Инициализируется список ключевых слов и фраз.
• ключевые_слова = ["запасной счёт", "код из СМС", "пенсионный фонд", "ваши права"]
2. Инициализация переменной - счётчика,
• счётчик_подозрительных_слов = 0
3. Разбить текстовую расшифровку на части.
• def разбиение(текст) → return слова.
4. Для каждого элемента из "слова" проверяем на наличие "ключевых слов", и увеличиваем счётчик, если слово встретилось.

→ см. далее

Дополнительный рабочий лист
(без рабочего листа №1 недействителен)

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

Шифр ПИ - 16
(заполняется участником)

5. Рассчитываем длину текста.
`def длина(текст) → return: длина.`

6. Определим пороговое значение.

$\text{порог} = 0.1 \cdot (\text{длина_текста})$
(10 процентов)

7. Если $\text{оценка} \geq \text{порог}$:

вернуть „высокая вероятность мошенничества“

вернуть: „низкая вероятность мошенничества“

Задача 3.

`import re`

`def sensitive_data(text):`

`patterns = {`

`„Паспорт“: r"1b1d{4}1s1d{6}1b",`

`„СНИЛС“: r"1b1d{33}-1d{33}-1d{33}1s1d{22}1b",`

`„Панель карты“: r"1b(?:1d{4}[-1s]?){33}1d{4}1b",`

`„CVC/CVV“: r"1b1d{3}1b",`

`„Телефон“: r"1b(?:1+718)[-1s]?(?:1d{33})?[-1s]?`

`„Email“: r"1b[А-За-ао-9._%+-]+@([А-За-ао-9.-]+`

`+1.[А-За-а-9]{2,7}1b"`

`}`

```
extracted_data = {}
```

```
for key, pattern in pattern.items():
```

```
    matches = re.findall(pattern, text)
```

```
    if matches:
```

```
        extracted_data[key] = matches
```

```
return extracted_data
```

```
text = "Добрый день, Ваня Банковская  
карта дома заблокирована, куда..."
```

```
result = ex_sensitive_data(text)
```

```
for category, values in result.items():
```

```
    print(f"{category}: {' '.join(values)}")
```

Для более корректной работы можно
добавить nlp модели для понимания
смыслов и контекстов заданного
текста.

Структура текста - сплошной текст,
содержащий речь человека и шумы.