

**Российская Федерация
КОМИТЕТ ПО ОБРАЗОВАНИЮ АДМИНИСТРАЦИИ
ГОРОДСКОГО ОКРУГА «ГОРОД КАЛИНИНГРАД»
МУНИЦИПАЛЬНОЕ АВТОНОМНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
города Калининграда ЛИЦЕЙ №49**

**Тема проекта: «Программа распознавания
лиц для биометрического турникета»**

Выполнил: ученик 10 «Б» класса
Мыльников Евгений Алексеевич

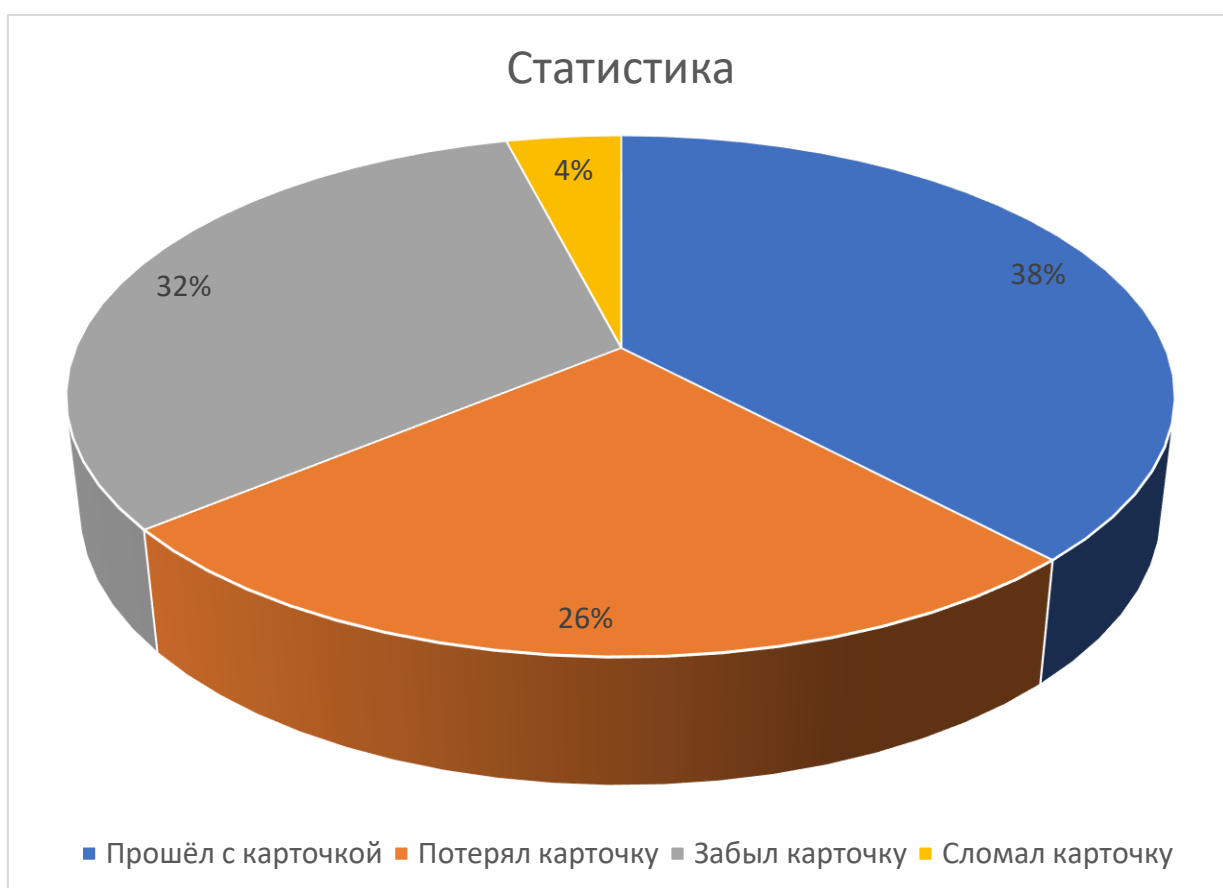
Руководитель проекта:
Бугрышева Татьяна Николаевна

Калининград
2025 г.

Актуальность проекта:

Для доказательства актуальности своего проекта я провёл наблюдение, в ходе которого выявил:

- многие ученики нашей школы забывают/теряют/ломают свои ключ-карты, что приводит к трудностям при входе в школу;
- посторонний/злоумышленник может найти/выкрасть ключ-карту у ученика и с её помощью проникнуть в школу.

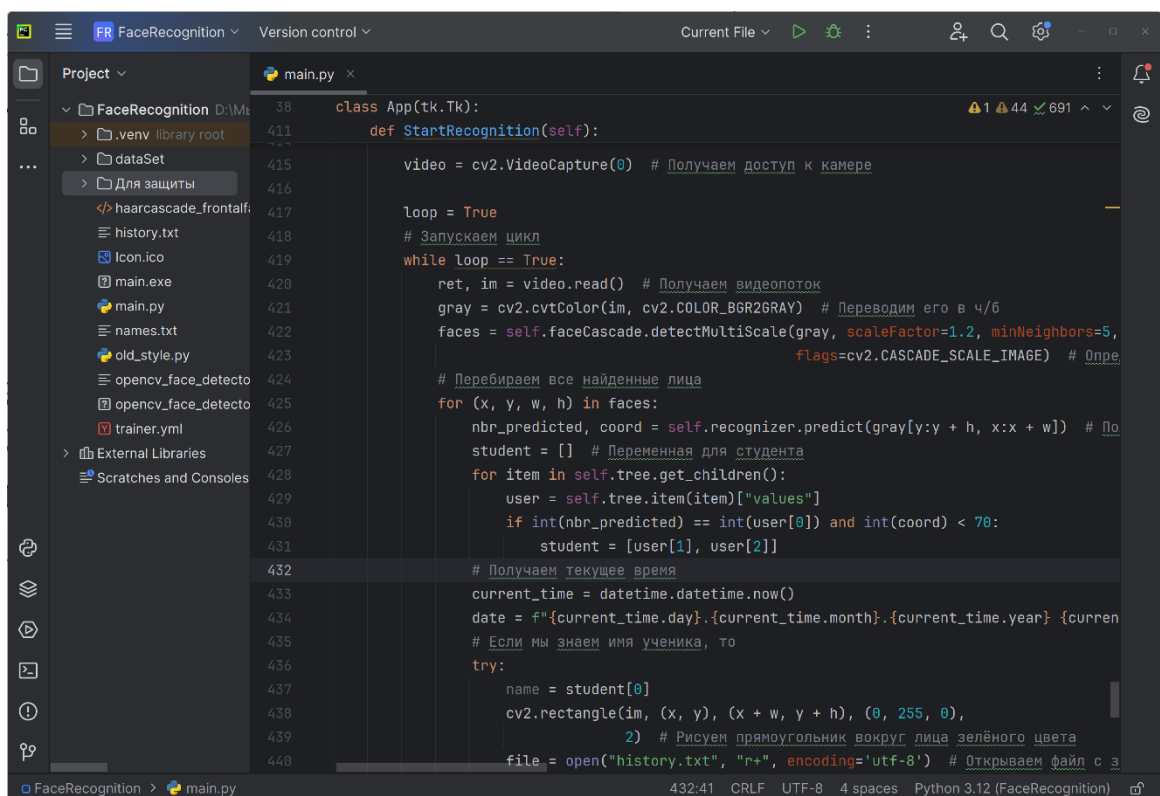


Проблема, положенная в основу проекта: Турникеты с ключ-картой не всегда безопасны и удобны для использования.

Цель: Разработать программу распознавания лиц для биометрического турникета до защиты проекта.

Этапы проекта:

ЭТАПЫ	СРОК
Изучить, как работает распознавание лиц.	До 03.11.24
Изучить весь список имеющихся фреймворков для распознавания лиц и их возможностей и выбрать наиболее подходящий.	До 15.11.24
Изучить документацию к выбранному фреймворку.	До 30.11.24
Создаём метод для распознавания лица с камеры.	До 20.12.24
Обучаем нейросеть запоминать лица и отличать их друг от друга.	До 01.02.25
Создаём базу данных для хранения моделей лиц и информации об учениках.	До 05.02.25
Создаём базу данных для хранения истории пропусков.	До 10.02.25
Добавляем в программу способы для загрузки модели лица.	До 30.03.25
Оптимизация кода.	До 15.04.25
Сборка и компиляция программы для демонстрации её работоспособности на защите.	До 18.04.25



```
class App(tk.Tk):
    def StartRecognition(self):
        video = cv2.VideoCapture(0) # Получаем доступ к камере

        loop = True
        # Запускаем цикл
        while loop == True:
            ret, im = video.read() # Получаем видеопоток
            gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY) # Переводим его в ч/б
            faces = self.faceCascade.detectMultiScale(gray, scaleFactor=1.2, minNeighbors=5,
                                                        flags=cv2.CASCADE_SCALE_IMAGE) # Определяем лица

            # Перебираем все найденные лица
            for (x, y, w, h) in faces:
                nbr_predicted, coord = self.recognizer.predict(gray[y:y + h, x:x + w]) # По
                student = [] # Переменная для студента
                for item in self.tree.get_children():
                    user = self.tree.item(item)["values"]
                    if int(nbr_predicted) == int(user[0]) and int(coord) < 70:
                        student = [user[1], user[2]]

            # Получаем текущее время
            current_time = datetime.datetime.now()
            date = f"{current_time.day}.{current_time.month}.{current_time.year} {current_time.hour}:{current_time.minute}:{current_time.second}"
            # Если мы знаем имя ученика, то
            try:
                name = student[0]
                cv2.rectangle(im, (x, y), (x + w, y + h), (0, 255, 0),
                               2) # Рисуем прямоугольник вокруг лица зелёного цвета
                file = open("history.txt", "a+", encoding='utf-8') # Открываем файл с з
```

Продукт проекта:

