Инструкция:

- Два дикшинари:

- questions {type:{0:0, 1:3}}

- students {type:{"Name Surname": StID}}

- Кнопка Generate берёт дикшинари students и создаёт ворд документ отдельно для каждого студента

- Документы объединяються в один PDF и печатаються

- Листы раздаються

- Экзамен написан, листы собраны

- Листы сканируются

- Запускается приложение, вводятся нужные параметры

- Grading (main.py)

- TypeIdentifier (type\_recognition\_3\_0.py)

- Проверка Type происходит в 3 этапа:

- Проверяется верхний Type с помощью doctr модели

- Проверяется нижний Type с помощью OpenCV

- Сверяется, если равны то значение тайп дальше отправляется, если нет, то берётся верхнее/нижнее значение (надо провести тесты в пятницу)

- ImageSplitter (split\_image\_1\_2.py)

- Скан разделяется на 3 части

- TestGrader (Grading\_2\_1.py)

- Оценка теста

- combine\_images (main.py)

- Объединение 3 частей обратно

- numbers\_recognition (dig\_rec.py)

- Распознавание St ID

- создание дикшинари result\_dict {(st\_id, type): grade}

- put\_percent (perc\_show.py)

- Пишутся Type и Grade в центре изображения \_TOTAL\_RESULT

- StudentGrades (excel\_results.py)

- 3 колонки St ID, Name Surname, Grade (всё из student и result\_dict дикшинарис)

Новый План:

Переместить распознавание (numbers\_recognition) St ID после Image Spliter

Поставить условие сравнения полученных двух Type сверху и снизу

Добавить вывод Type, Name и Surname в numbers\_recognition

Поменять разархивирование дикшинари в файле формирования бланков и эксель