

Инструкция по использованию программой Insighter

1. Введение

Insighter - это программа на Python, позволяющая автоматически проверять работы учеников, давать по ним статистику и строить графики. Программа предназначена в основном для упрощения работы учителям - она автоматизирует рутинную проверку тестов формата "цифра - буква".

Программа не требует установки Python и работает в среде Windows 10/11. Для запуска программы необходимо запустить файл Insighter.exe, который находится в одноименном архиве в репозитории GitHub. Для работы программы вам необходим только файл .exe, все остальные файлы в репозитории - это дополнительные файлы с исходным кодом, примерами и прочим.

2. Структура файлов и папок

Во время работы программы использует множество различных папок и файлов.

Для работы программы необходимы:

- Системный файл sys.json
- Системный лог-файл syslog.json

Важно, если вы не разработчик, то не трогайте файлы с расширением json!

Все действия происходят в папке с программой, т.е. в папке с этими двумя json-файлами. Они создаются автоматически при первом запуске или после сброса. Рекомендуется создать основную рабочую папку, например, на рабочем столе, куда положить Insighter.exe. Все файлы и папки программа будет создавать внутри папки, в которой находится. Не перемещайте ни в коем случае никакие папки и файлы, которые создает вам программа. Основное хранилище программы - папка archive, которая создается самой программой при выборе режима "проверки работ". В ней хранятся отдельные папки классов, в которых, в свою очередь, сами файлы с результатами проверки. Не трогайте и не редактируйте файлы в этой папке. В ней же, в папке каждого класса, будут храниться и системные json-файлы для подробной статистики и режима сравнения. Программа поддерживает работу с txt и csv файлами. (подробнее о каждом в режиме проверки работ).

Для проверки работ используются файлы с правильными ответами, с критериями оценивания, отсутствующими учениками; для статистики создается одноразовый файл statistics.txt или results.csv в зависимости от режима. Подробнее о каждом далее.

3. Режимы программы

3.1 Первичный запуск

При первом запуске программы будет появляться окно ввода данных. Его необходимо заполнить, все данные будут браться программой оттуда.

Необходимо заполнить семь полей: класс(числоБуква), название работы(любое название без символов _ и латинских букв, только кириллица), дату работы(дд.мм.гггг), названия файлов с ответами, с критериями оценивания, со списком учеников (не забудьте расширение .txt), название папки с работами учеников, и, наконец, названия файла с отсутствующими(в это поле также можно вписать “auto”, об этом далее). О критерии к формату каждого файла далее. После заполнения и перезапуска программы вам будет доступно меню с режимами. Привожу таблицу с описанием каждого режима.

Название режима	Описание
Проверка работ	Проверяет работы по введенным данным и записывает в файл в архив
Перезапись данных	Открывает окно для ввода данных
Поиск по работам	Выводит все оценки введенного ученика в указанном файле или папке
Статистика	Дает краткую и подробную статистику по выбранной работе, строит графики
Генерация	Генерирует папки и файлы по введенным данным
Случайный вызов	Вызывает учеников “к доске” в случайном порядке
Сброс данных	Частичный или полный сброс файлов и папок программы
Сравнение работ	По введенным работам или периоду

	строит графики динамики успеваемости класса или ученика
Помощь	Открывает инструкцию или дает ссылку на нее
Импорт данных	Импортирует данные из Яндекс Форм по json-файлу
Вывод по очереди	Поочередно выводит на экран отсортированные по фамилии и имени оценки учеников

3.2 Проверка работ

Для работы использует введенные данные в форму заполнения данных.

При запуске режима программа спросит вас, нужна ли вам подробная статистика для работы. При положительном ответе будет создан json-файл, станет доступна подробная статистика и сравнение этой работы с другими, так что рекомендуется всегда включать эту опцию. Затем программа спросит вас о формате файла, который она создаст: csv или txt. Csv-файлы доступны для работы в excel. В остальном, их функциональность не отличается. Последнее, о чем вас спросит программа, это сортировка. Можно выбрать один из четырех режимов(по именам, по оценкам(по убыванию и по возрастанию) и по умолчанию). После этого программа создаст файл (txt или csv), назвав его в формате: класс_название работы_дата.расширение и положит его в архив в папку соответствующего класса.

Для работы необходимы, как говорилось ранее, 4 основных файла и папка с работами:

- со списком учеников - текстовый файл, в котором в столбик написаны имена и фамилии учащихся;
- с верными ответами - текстовый файл, в котором в столбик с номерами записаны верные ответы. Пример содержимого такого файла в папке templates в репозитории GitHub. (answers.txt);
- с критериям оценивания - текстовый файл, в котором n-1 строк, где n - система оценивания(по умолчанию - пятибалльная, т.е. 4 строчки).

Каждая строка в формате: {оценка _ от _ до _ баллов}. Иной формат

программа не поддерживает. Обе границы включаются. Пример содержимого такого файла с комментариями находится в папке templates в репозитории GitHub. (marks.txt);

- файл с отсутствующими учениками. По формату такой же, как и со списком учеников, только там вписаны ученики, которых программа будет расценивать как отсутствующих. Если указать путь к такому файлу, то в независимости от того, есть или нет работа такого ученика в папке с работами, он будет записан как отсутствующий. Если вместо пути к файлу указать "auto", то программа запишет в отсутствующих тех учеников, которые есть в полном списке, но чьих работ нет в папке. Если работа есть, и даже если она пустая, программа проверит ее и выставит оценку. Вероятно, минимальную из возможных;
- папка с работами - это папка с текстовыми файлами учеников. Каждый файл должен быть в формате: имя_фамилия.txt. Их неудобно создавать вручную, поэтому для этого есть отдельный режим - генерации(см. пункт 3.6). Содержимое такого файла должно быть в том же формате, что и в файле с ответами. В противном случае ответ будет засчитан как неверный. Примеры файлов и папки в папке templates в репозитории GitHub. (puples.txt и workname)

3.3 Перезапись данных

При запуске режима проверка на миссклик и затем удаляется системный файл. При перезапуске программы откроется окно для ввода данных для проверки работ и других режимов.

3.4 Поиск по работам

При запуске режима программа попросит ввести имя и фамилию интересующего вас ученика. Затем нужно будет выбрать класс из архива. После выбора класса программа спросит вас о режиме - по целой папке, т.е. по всем работам, или по одной работе. Во втором случае программа попросит выбрать файл и выведет результат поиска в формате: название работы - имя ученика: оценка. В первом случае(при выборе поиска по папке) будут выведены подобные сообщения для каждой работы. Если ученик

будет не найден, то программа сообщит и об этом в режиме поиска по файлу. В режиме же поиска по папке файл, где ученика нет, будет проигнорирован.

3.5 Статистика по работам

При запуске этого режима программа попросит выбрать вас папку, а затем и конкретную работу для получения статистики. Рядом с теми работами, по которым невозможно получить подробную статистику будет написано: <только краткая статистика>. При выборе такой работы программа сразу создаст одноразовый(при следующем запуске его содержимое перезапишется) файл statistics.txt. Пример такого файла находится в папке templates(brief_statistics.txt) в репозитории на GitHub. Если вы выберите файл, для которого доступна подробная статистика, программа спросит вас, какую конкретно статистику вы хотите получить: краткую, подробную по классу, подробную по ученику или подробную по заданиям. При выборе краткой статистике произойдет то же самое, что и в первом случае.

При выборе подробной статистики программа запишет ее в одноразовый файл statistics.txt(она охватывает гораздо больше данных и дает рекомендации по каждому заданию, поэтому создание json-файлов на стадии проверки рекомендуется). Пример такого файла находится в папке templates(deep_statistics.txt) в репозитории на GitHub. Затем вас спросят о том, нужны ли вам графики. Если да, то программа построит столбчатые диаграммы зависимости оценок от их количества и зависимости номера задания от процента правильных ответов на него. (это просто графическое представление того, что и так уже есть в файле statistics.txt). Пример диаграмм находится в папке templates/img (deep_stat_class.png) в репозитории на GitHub. После того, как вы закроете окно с графиками автоматически откроется файл statistics.txt.

При выборе режима подробной статистики по ученику программа попросит вас имя и фамилию ученика с учетом регистра и затем сгенерирует статистику по нему в statistics.txt. Пример такого файла находится в папке templates(pupil_statistics.txt) в репозитории на GitHub. Если ученик не будет найден, программа выведет соответствующее сообщение.

При выборе режима подробной статистики по заданиям, программа сначала спросит вас об отметке правильности ответов. (по умолчанию стоит “1” для верного ответа и “0” для неверного. Вы можете указать свои отметки через пробел, причем сначала для верных ответов, затем для неправильных.) А затем

создаст одноразовый файл results.csv. В этом файле - таблица, которая показывает как конкретно каждый ученик выполнил задание. По сути, это более подробная развертка части подробной статистики по классу. Пример такого файла находится в папке templates(results.csv).

3.6 Генерация директорий и файлов

Этот режим предназначен для автоматической генерации файлов и папок для работ. Всего есть два режима - быстрая генерация, без ручной настройки, и с точной ручной настройкой. В режиме без настройки программа спросит вас название работы(по факту - папка с работами), название файла с именами учеников(по именам оттуда программа будет генерировать соответствующие txt файлы), количество полей для ответов(оно одинаково в answers и marks). После этого программа сгенерирует шаблонные файлы answers.txt, marks.txt и пустой missings.txt. Создаст папку с уже подготовленными работами файлов для учеников. Их останется заполнить ответами и можно проверять при помощи режима проверки работ(см. 3.2). Если в процессе генерации возникнут ошибки, то программа остановит генерацию и сообщит вам о них(например, если вы ввели имя несуществующего файла).

При выборе ручной настройки генерации программа расспросит вас о каждом файле. Это удобно, когда нужно сгенерировать только, например, папку с работами, а портить, т.е. перезаписывать файлы с критериями оценивания не нужно. Полный вывод представлен в файле manset_generation.txt в папке templates в репозитории на GitHub.

3.7 Случайный вызов

Этот режим позволяет вызывать “к доске” учеников в случайному порядке. При запуске вам предложат ввести число, файл или пропустить ввод. Если вы введете число, то сообщения будут выводится в формате:

Ученик N идет к доске

Если вы введете названия файла, он должен быть с именами и фамилиями учеников, то вывод будет в формате:

Ученик Имя Фамилия идет к доске

Если же вы пропустите ввод, то будет взят последний файл, который вы вводили в поле файла с учениками при вводе данных(см. 3.2 и 3.1) и имена и фамилии учеников будут браться оттуда в том же формате.

Когда все ученики будут спрошены, будет выведено сообщение: "Вы всех спросили!"

Чтобы остановить вывод учеников, вместо нажатия enter впишите stop или просто остановите программу вручную через редактор кода.

3.8 Сброс данных

Режим, позволяющий вернуть программу в изначальное состояние или сбросить некоторые файлы. При запуске режима вам надо будет выбрать, какой конкретно режим удаления вас интересует: сброс файлов, сброс папок и файлов или сброс до начальной конфигурации.

При выборе режима сброса файлов, будет проверка на миссклик - нужно ввести любое натуральное число. После ввода будут удалены: системные файлы sys.json и syslog.json, файлы, с названиями по умолчанию(answers.txt, marks.txt, missings.txt) и их аналоги, имена которых вы указывали в последний раз при записи данных. Программа сообщит о всех удаленных файлах. Такой режим полезен для очистки после блока работ, где, например, у разных классов из одной параллели на работу были одни и те же ответы.

При выборе режима сброса файлов и папок, также будет проверка на миссклик. Программа создаст файл delete.txt, откуда нужно будет ввести четырехзначный пароль. После корректного ввода пароля, будут удалены все те же файлы, что и в предыдущем режиме, но так же и папка archive. Такое удаление необходимо периодически выполнять, чтобы не забивать архив. Если вам нужны файлы из архива, вы можете просто перенести папку archive в другую директорию, а при надобности заменить ее в рабочей, если, например, нужно будет посмотреть отчет за давнюю работу.

Есть и последний режим - он удаляет абсолютно все из рабочей директории, все файлы и папки, кроме файлов .ru и папки modules, необходимой для работы программы. Он доступен только в режиме разработчика. Подробнее о нем смотреть в пункте 5

3.9 Сравнение работ

Режим, позволяющий сравнивать несколько работ. При запуске режима программа попросит выбрать нужную папку. После выбора папки необходимо будет выбрать режим сравнения: по ученику или по всему классу. После программы спросит вас о выборе работ: вы можете сравнить работы по периоду или конкретные работы. Сравнение по периоду поддерживает форматы:

- дд.мм.гггг - дд.мм.гггг(промежуток между двумя датами)
- мм(номер одного месяца)
- мм-мм(промежуток от начала первого месяца, до конца второго)

Сравнение по конкретным работам поддерживает всего два формата: ввод номеров нужных работ через пробел или можно ввести “все”/“all”. Тогда будут сравнены все доступные работы. Если вы выбираете режим сравнение по периоду, то после его ввода программа выведет вам работы, которые будут проверены.

После выбора работ программа сгенерирует графики: два графика по ученику: зависимость работ от количества правильных ответов за эту работу и от оценки за каждую из работ. И четыре графика для класса: зависимость работ от средней оценки по классу, количества отсутствующих, среднего процента правильных ответов и среднего количества правильных ответов. Все примеры графиков приведены в папке templates/img в репозитории на GitHub(comapre_graphics_n.png).

ВАЖНО, что режим сравнения работ доступен только работ, для которых доступна подробная статистика. Иные работы не отображаются при выборе. Также, если у вас некорректно отображаются графики, не видно например, часть текста, то необходимо настроить показ графика в окне графика. Как это делать показано в файлах в папке templates/img в репозитории на GitHub(graphset.png)

3.10 Помощь

Режим открывает файл instruction.pdf или дает ссылку на него.

3.11 Импорт данных

Режим предназначен для импорта данных из Яндекс Форм. При запуске режима программа запросит путь к json-файлу. Получить его несложно. После создания собственного теста(инструкция по тому, как конкретно создать корректный тест будет ниже) необходимо перейти во вкладку “ответы”. Затем в левом верхнем углу в выпадающем меню выбрать “параметры файла: json” и нажать кнопку скачать. После скачивания файла перенесите его в рабочую директорию(это, впрочем, необязательно, но так вам не придется возится с абсолютным путем к файлу). И вот уже после этого скопируйте имя файла(или путь к нему) и вставьте при запросе в программу. Все вышеописанные шаги в картинках показаны в папке templates (import_step_1, import_step_2). После ввода файла от вас потребуется название работы - оно же название папки, в которой программа будет создавать файлы учеников. Программа автоматически нормализует данные и приведет их к рабочему формату, т.е. создаст папку с работами учеников(как будто бы вы воспользовались режимом генерации и затем вручную загрузили туда файлы).

Чтобы создать корректный тест, следуйте следующим правилам:

1. Обязательно сделайте вопрос “Фамилия Имя”
2. Вопросы должны быть названы как “Задание n”, где n - уникальный номер задания
3. Программа обрабатывает только ручной ввод. Например, если вопрос будет подразумевать ссылку на файл, например, то такой вопрос помимо того, что не будет обработан, так он еще может сломать программу

В будущем у этого режима, вероятно, появится большая вариативность, но пока важно соблюдать именно эти правила, т.к. именно по ним программа отслеживает наличие того или иного вопроса в teste.

После импорта вы можете работать с этой папкой как с обычной папкой с заданиями. Написать к ней файл с ответами и файл с критериями оценивания, запустить режим проверки работ, затем статистики или сравнения. После импорта она будет такой же, как и созданные вручную папки.

3.12 Вывод по очереди

Этот весьма незамысловатый режим предназначен для удобного выставления оценок в журнал, электронный или бумажный. При запуске программа попросит вас выбрать папку класса, а затем и работу. После будет один за одним выводится ученик и его оценка, отсортированные по фамилии и имени. После каждого ученика необходимо ввести что-то, проще всего просто нажать `enter`, затем программа выведет следующего ученика, очистив экран от остальных. Это может быть полезно, чтобы не отвлекаться и не теряться в журнале при большом количестве учеников.

4. Исправление ошибок

В процессе работы программы могут возникать ошибки. Большую часть из них программа обрабатывает и выводит поясняющий текст, что и как нужно делать. Однако, может быть и такое, что программа просто выдаст ошибку, без всяких пояснений.

Ни в коем случае не трогайте исходный код. Если вы случайно что-то удалили или сломали, просто переустановите программу с GitHub. Если в процессе выполнения возникла ошибка `JSONEncoderError`, там так и будет написано, то, вероятно, что-то случилось с json-файлами. Вам следует удалить оба(и `sys.json`, и `syslog.json`). Программа будет работать корректно после их удаления, если вы случайно изменили один из этих файлов. Если в процессе выполнения любого из режимов происходит ошибка, перепроверьте соответствие данных инструкции. Возможно, вы упустили где нибудь знак, или случайно удалили заголовок файла. Подробные шаги по запуску программы описаны в `README.md` в репозитории на GitHub. Если исправить ошибку никак не удается, то напишите на почту с проблемой(см. пункт 7).

5. Для разработчиков и о репозитории GitHub

В репозитории на GitHub достаточно большое количество различных папок и файлов. Папка Old versions - это папка с более старыми версиями программы. В папке others находятся два файла - для тестов программы и первый прототип этой программы. В папке templates, к которой я уже неоднократно обращался, находятся примеры работы программы. В папке The last version code хранятся .ру файлы, которые были использованы для генерации .exe. В архиве Insighter.zip хранится как раз сам .exe-файл с программой. Соответственно, файл instruction.pdf вы сейчас читаете.

В программе существует режим для дебага программы. Запустить режим разработчика достаточно просто, в файле syslog.json значение debug поменять на True. После каждого перезапуска программы файл очищается и режим разработчика выключается. Туда записываются значения почти всех переменных, которые использует программа для работы в том или ином режиме. Как говорилось ранее, в режиме разработчика доступен режим сброса до начальной конфигурации программы. Пароль для него генерируется в файле syslog.json прям во время работы программы. Его нужно будет посмотреть, не останавливая программу. После ввода пароля будет таймер и по его истечению все файлы и папки директории будут удалены.

Программа построена на принципе ООП. Конкретные классы разбиты по файлам и находятся в папке modules. Почти к каждому классу и методу даны комментарии и докстринги, в самом коде также есть комментарии. Если вы разработчик и хотите протестировать программу, то в папке The last version code вы увидите два основных .ру файла - [insighter.py](#) и [run.py](#). [Run.py](#) необходим для корректной работы исключительно .exe, при запуске в IDE лучше запускать первый файл.

6. Для кого эта программа?

Данная программа ориентирована на преподавателей, которым необходимо автоматизировать процесс проверки письменных работ, а также получить подробную статистику, анализ результатов и наглядные визуализации. Она может быть полезна в следующих случаях: при проведении контрольных,

тестовых, диагностических и пробных экзаменационных работ в текстовом формате; при необходимости систематической проверки большого количества файлов от разных учащихся; при анализе успеваемости, типичных ошибок и прогресса обучающихся; в условиях дистанционного обучения, когда работы передаются в виде файлов; при желании повысить точность и снизить нагрузку на преподавателя при ручной проверке. Программа подходит пользователям без опыта программирования: взаимодействие происходит в интерактивном режиме через простые текстовые команды, все действия сопровождаются пояснениями, а возможные ошибки обработки вводов — перехватываются и разъясняются.

7. Обратная связь

Несмотря на то, что программа уже была протестирована, в ней могут быть ошибки, неточности в формулировках и отсутствие некоторых полезных функций. Все предложения и вопросы присылайте на почту: andr.udalov10@gmail.com или в телеграм: [@dostoievsky_1821](https://t.me/dostoievsky_1821).

8. Рекомендации по использованию и некоторые дополнения

Программа часто спрашивает у вас вопросы, на которые ответ либо да, либо нет. Чтобы ответить положительно, вы можете написать: 1, lf, да. Любой из этих трех вариантов программа расценит как утвердительный ответ. Любые остальные символы - как отрицательный(рекомендуется просто нажимать enter для экономии времени). Программой почти полностью можно управлять вводом чисел. В начальном меню вы можете написать не только число, выбор режима, но и полноценно написать название режима(в этом нет никакого смысла, но такая возможность есть). Хорошая рекомендация - не трогайте названия файлов, если в этом нет необходимости, в таком случае у вас меньше шанс ввести не то. Если у вас несколько классов, то рекомендуется создать папки для ответов и критерий оценивания, где будут файлы для разных классов. Если ответы будут совпадать, что редко, или критерий оценивания, что чаще, просто выбираете нужный файл и не нужно ничего переделывать. Если вам нужна подробная статистика по ученику или по классу, то сохраняйте этот файл! Также рекомендуется иногда чистить архив, копируя его в другую директорию, а потом удаляя через режим сброса. Рекомендуется для всех работ создавать json-файлы для

подробной статистики и сравнения работ. Как уже говорилось ранее, лучше работать в одной папке с программой, чтобы файлы не разбрасывались по всему компьютеру, это значительно упростит вам задачу. И, последнее, если вы вводите куда-то название файла и он находится не в рабочей директории, то нужно указывать полный путь к нему. Но лучше все файлы хранить в рабочей директории.

9. Дальнейшие версии

Это уже вторая версия этой программы, уже в релизе, так как она была достаточно протестирована. В этой версии, наконец, удалось собрать .exe, чтобы не было необходимости в IDE для запуска программы. В дальнейшем планируется сделать полноценный графический интерфейс для нее.

На базе Flask, вместо работы с файлами и папками - подключить к ней базу данных(планируется версия 2.0 в конце лета 2026 года), сделать чат-бота в телеграмме для моментальной передачи результатов ученикам, а может даже и полноценное приложение для учеников по типу Яндекс Форм.

P.S. в этой версии инструкции могут быть небольшие опечатки по поводу запуска программы, потому что только с этой версии появилась возможность запускать программу не в IDE, а как полноценное приложение.

Спасибо за пользование программой *Insighter*. Любые предложения и пожелания отправляйте на почту или в телеграмм.