

[Finuniversity](#) / **3\_assignment-boikova2604** Privategenerated from [Finuniversity/3\\_workshop\\_python\\_1y\\_1s](#)[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) main ▾**3\_assignment-boikova2604 /**  
**3\_Assignment.md**[Go to file](#)

...

**github-classroom** Initial com...

Latest commit d54bf14 44 minutes ago

[History](#) 1 contributor 66 lines (50 sloc) | 11.7 KB[Raw](#)[Blame](#)

## 🔗 3 Задание. Практикума по программированию.

**Тема задания: «Реализация собственного пакета модулей по манипулированию табличными данными».**

Количество баллов – 10 (требуется защита работы).

Базовая часть – 3 баллы.

Дополнительная часть – до 7 баллов.

Отсутствие защиты штрафует 5 баллами.

Нарушение дедлайна штрафует 5 балами (минимальный количество баллов 1).

**Дедлайн: 24.11.2022**

**Методические рекомендации:.*****Выполняется в jupyter notebook.***

Оформление блокнота должна присутствовать следующая информация: предмет, номер задания, ФИО, группа, формулировка базового задания, формулировка дополнительных заданий, код и результат выполнения кода.

**Базовая часть:.**

На базе модулей: csv, pickle и прямой работы с файлами реализовать следующий базовый функционал:

1. функций load\_table, save\_table по загрузке/сохранению табличных данных во внутреннее представление модуля/из внутреннего представления модуля:
  - файла формата csv (отдельный модуль с load\_table, save\_table в рамках общего пакета).
  - файла формата pickle (отдельный модуль с load\_table, save\_table в рамках общего пакета), модуль использует структуру данных для представления таблицы, удобную автору работы.
  - текстового файла (только функция save\_table сохраняющая в текстовом файле представление таблицы, аналогичное выводу на печать с помощью функции print\_table()).

**Примечание:** внутреннее представление может базироваться на словаре, где по разным ключам хранятся ключевые «атрибуты» таблицы, а значения таблицы хранятся в виде вложенных списков. Студент может выбрать другое внутреннее представление таблицы (согласовав его с преподавателем), в том числе, студенты знакомые с ООП на Python, могут реализовать собственный класс для таблицы. При определении api модулей максимально полно использовать возможности сигнатур функций на Python (значения по умолчанию, упаковка/распаковка, в т.ч. именованных параметров, возвращение множественных значений), интенсивно выполнять проверки и возбуждать исключительные ситуации.

2. модуля с базовыми операциями над таблицами:

- get\_rows\_by\_number(start, [stop], copy\_table=False) – получение

таблицы из одной строки или из строк из интервала по номеру строки. Функция либо копирует исходные данные, либо создает новое представление таблицы, работающее с исходным набором данных (`copy_table=False`), таким образом изменения, внесенные через это представления будут наблюдаться и в исходной таблице.

- `get_rows_by_index(val1, ... , copy_table=False)` – получение новой таблицы из одной строки или из строк со значениями в первом столбце, совпадающими с переданными аргументами `val1, ... , valN`. Функция либо копирует исходные данные, либо создает новое представление таблицы, работающее с исходным набором данных (`copy_table=False`), таким образом изменения, внесенные через это представления будут наблюдаться и в исходной таблице.
- `get_column_types(by_number=True)` – получение словаря вида столбец:тип\_значений. Тип значения: `int`, `float`, `bool`, `str` (по умолчанию для всех столбцов). Параметр `by_number` определяет вид значения столбец – целочисленный индекс столбца или его строковое представление.
- `set_column_types(types_dict, by_number=True)` – задание словаря вида столбец:тип\_значений. Тип значения: `int`, `float`, `bool`, `str` (по умолчанию для всех столбцов). Параметр `by_number` определяет вид значения столбец – целочисленный индекс столбца или его строковое представление.
- `get_values(column=0)` – получение списка значений (типизированных согласно типу столбца) таблицы из столбца либо по номеру столбца (целое число, значение по умолчанию 0, либо по имени столбца).
- `get_value(column=0)` – аналог `get_values(column=0)` для представления таблицы с одной строкой, возвращает не список, а одно значение (типизированное согласно типу столбца).
- `set_values(values, column=0)` – задание списка значений `values` для столбца таблицы (типизированных согласно типу столбца) либо по номеру столбца (целое число, значение по умолчанию 0, либо по имени столбца).
- `set_value(column=0)` – аналог `set_values(value, column=0)` для представления таблицы с одной строкой, устанавливает не список значений, а одно значение (типизированное согласно типу столбца).

- `print_table()` – вывод таблицы на печать.

3. Для каждой функции должно быть реализована генерация не менее одного вида исключительных ситуаций.

### Дополнительные задания:

1. В `load_table` реализовать `load_table(file1, ...)` – поддержку загрузки таблицы, разбитой на несколько файлов (произвольное количество файлов) (для форматов `csv` и `pickle`). В случае несоответствия структуры столбцов файлов вызывать исключительную ситуацию. Баллы – 1
2. Расширение задания 1. В `save_table` реализовать поддержку сохранения таблицы в разбитой на несколько файлов (произвольное количество файлов) по параметру `max_rows`, определяющему максимальное количество строк в файле. Файлы `csv` и `pickle`, полученные с помощью `save_table` должны быть совместимы с `load_table` из задания 1. Баллы – 1
3. Реализовать функцию `concat(table1, table2)` и `split(row_number)` склеивающую две таблицы или разбивающую одну таблицу на 2 по номеру строки. Баллы – 1
4. Реализовать автоматическое определение типа столбцов по хранящимся в таблице значениям. Оформить как отдельную функцию и встроить этот функционал как опцию работы функции `load_table`. Баллы – 2
5. Реализовать поддержку дополнительного типа значений «дата и время» на основе модуля `datetime`. Баллы – 2
6. Добавить набор функций `add`, `sub`, `mul`, `div`, которые обеспечат выполнение арифметических операций для столбцов типа `int`, `float`, `bool`. Продумать сигнатуру функций и изменения в другие функции, которые позволят удобно выполнять арифметические операции со столбцами и присваивать результаты выч. Реализовать реагирование на некорректные значения с помощью генерации исключительных ситуаций. Баллы – 2

7. По аналогии с п. 6 реализовать функции `eq (==)`, `gr (>)`, `ls (<)`, `ge (>=)`, `le (<=)`, `ne (==)`, которые возвращают список булевских значений длиной в количество строк сравниваемых столбцов. Реализовать функцию `filter_rows (bool_list, copy_table=False)` – получение новой таблицы из строк для которых в `bool_list` (длиной в количество строк в таблице) находится значение `True`. Баллы - 3
8. Реализовать функцию `merge_tables(table1, table2, by_number=True)`: в результате слияния создается таблица с набором столбцов, соответствующих объединенному набору столбцов исходных таблиц. Соответствие строк ищется либо по их номеру (`by_number=True`) либо по значению индекса (1й столбец). При выполнении слияния возможно множество конфликтных ситуаций. Автор должен их описать и определить допустимый способ реакции на них (в т.ч. через дополнительные параметры функции и инициацию исключительных ситуаций). Баллы - 2
9. Реализовать полноценную поддержку значения `None` в незаполненных ячейках таблицы. Должно работать при загрузке ячеек с пропусками значений, при операциях приводящих к появлению пустых ячеек, при работе с `get` и `set` операциями. Баллы - 2