|  |  |  |
| --- | --- | --- |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное автономное  образовательное учреждение высшего образования  «Пермский государственный национальный  исследовательский университет» | | |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по индивидуальной работе №2  по дисциплине «Язык программирования Python»  Вариант 9 | | |
|  | | Работу выполнил  студент группы ИТ-7,8-2023 1 курса  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Рогозин Е.А.  «\_\_\_» \_\_\_\_\_\_\_\_2025 г. |
| Работу проверил  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Рубцова М.Б.  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2025 г. |
| Пермь 2025 | | |

СОДЕРЖАНИЕ

[Постановка задачи 3](#_Toc153130027)

[Алгоритм решения 3](#_Toc153130028)

[Тестирование 3](#_Toc153130029)

[Код программы 3](#_Toc153130030)

[Инструкция по применению стилей и оформлению работы 4](#_Toc153130031)

# Постановка задачи

|  |
| --- |
| Реализуйте структуру данных "стек". Напишите программу, содержащую описание стека и моделирующую работу стека, реализовав все указанные здесь методы. Программа считывает последовательность команд и в зависимости от команды выполняет ту или иную операцию. После выполнения каждой команды программа должна вывести одну строчку. Возможные команды для программы:  push n - Добавить в стек число n (значение n задается после команды). Программа должна вывести ok.  pop - Удалить из стека последний элемент. Программа должна вывести его значение.  back - Программа должна вывести значение последнего элемента, не удаляя его из стека.  size - Программа должна вывести количество элементов в стеке.  clear - Программа должна очистить стек и вывести ok.  exit - Программа должна вывести bye и завершить работу.  Входные данные: команды управления стеком вводятся в описанном ранее формате по 1 на строке. Гарантируется, что набор входных команд удовлетворяет следующим требованиям: максимальное количество элементов в стеке в любой момент не превосходит100, все команды pop и back корректны, то есть при их исполнении в стеке содержится хотя бы один элемент.  Выходные данные: протокол работы со стеком, по 1 сообщению в строке.  Примеры:  push 3  ok  push 14  ok  size  2  clear  ok  push 1  ok  back  1  push 2  ok  back  2  pop  2  size  1  pop  1  size  0  exit  bye |

# Алгоритм решения

По заданию нужно реализовать стек, и первое что стоило сделать - выбрать с помощью каких структур он будет реализовываться. Для реализации стека можно использовать связанные списки и динамические массивы. В этом случае использовались списки, они были выбраны в основном из-за схожести списка и стека(функций которые нужно будет реализовать) и из-за того, что количество элементов в стеке постоянно будет изменяться.

Ход создания программы:

Сначала был написан класс, в котором содержались конструктор по умолчанию (\_\_init\_\_) и методы для работы со стеком (push, pop, clear, size, back). После этого вне класса была создана функция, которая отвечает за выполнение этих команд. Принцип работы: функция принимает строку и через классовые методы выполняет команды, которые были вызваны пользователем.

После создания класса и функции было написано само тело программы, где создаётся элемент класса, а также циклически производится ввод команд с клавиатуры и вызов функции.

Далее проводилось тестирование программы, доработки в методах класса и функции для обработки команд. В том числе была добавлена обработка исключений.

Исключения:

Программа могла выдать 2 исключения в процессе работы программы: первое - ValueError, которое выводилось, когда в стек пытались добавить нечисловой элемент(например букву/слово/символ), и второе - InputError, когда либо ввод не производился(если была введена команда “push” без аргумента), либо пытались удалить или вывести последний элемент с помощью команд “back” и “pop”, когда стек был пуст.

После исправления ошибок и всех доработок был разработан дружественный интерфейс, который представляет из себя подсказки при вводе и выводе и команду “help”, которая показывает возможные команды.

# Тестирование

Попробуем добавить в стек цифру

>Input a command: push 1

>ok

Добавим число из двух и более цифр

>Input a command: push 103

>ok

Попробуем добавить нечисловой элемент

>Input a command: push a

>ValueError: an element " a " is not a number

Нечисловой элемент с цифрой спереди

>Input a command: push 1a

>ValueError: an element " 1a " is not a number

Посмотрим размер стека

>Input a command: size

>size = 2

Посмотрим значение последнего элемента

>Input a command: back

>Last element = 103

Проверим что элемент не удалился

>Input a command: size

>size = 2

Теперь удалим его и посмотрим размер

>Input a command: pop

>Element deleted: 103

>Input a command: size

>size = 1

Добавим ещё элемент и очистим стек

>Input a command: push 2

>ok

>Input a command: clear

>ok

Проверим что стек пуст

>Input a command: size

>size = 0

Введем несуществующую команду

>Input a command: egor

>Command " egor " does not exist

>Type "help" to see available commands

Введем команду help

>Input a command: help

>push n - adds new element(n) to the stack

>size - shows a size of the stack

>back - shows a last element of the stack

>pop - deletes a last element of the stack

>clear - deletes all elements in the stack

>exit - ends the program

Введем push без аргумента

>Input a command: push

>IndexError: input an element

Очистим стек и попробуем удалить последний элемент

>Input a command: clear

>ok

>Input a command: pop

>IndexError: a stack has no elements

Завершим программу

>Input a command: exit

>bye

# Код программы

[Код](https://github.com/chavasse97/python_ikm_sem2.git)(гиперссылка на репозиторий)

# Инструкция по применению стилей и оформлению работы

Для оформления частей отчёта следует использовать заранее созданные стили. Все стили, которые могут пригодиться, начинаются с «ЛР. 1 …».



**ЛР. 1 Обычный** – для оформления текста задания и алгоритма решения.

**ЛР. 1 Текст программы** – для оформления кода программы.

**ЛР. 1 Заголовок 1** – заголовок первого уровня (для того, чтобы озаглавить основные разделы отчета).

**ЛР. 1 Заголовок 2** – заголовок второго уровня (для того, чтобы озаглавить подразделы).

Для того, чтобы перенести текст следующего блока на другую страницу, необходимо воспользоваться инструментов «Разрыв страницы» в разделе «Вставка».

