|  |  |  |
| --- | --- | --- |
| 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 10 | | |
|  | | Работу выполнил  студент группы ИТ-5-2024 1 курса  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Захаров П.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

# Очередь. Реализуйте структуру данных "очередь". Напишите программу, содержащую описание очереди и моделирующую работу очереди, 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 очередь число n (значение n задается после команды). Программа должна вывести ok. pop Удалить из очереди первый элемент. Программа должна вывести его значение. front Программа должна вывести значение первого элемента, не удаляя его из очереди. size Программа должна вывести количество элементов в очереди. clear Программа должна очистить очередь и вывести ok. exit Программа должна вывести bye и завершить работу. Перед исполнением операций front и pop программа должна проверять, содержится ли в очереди хотя бы один элемент. Если во входных данных встречается операция front или pop, и при этом очередь пуста, то программа должна вместо числового значения вывести строку error.

# Алгоритм решения

Для решения этой задачи я реализовал класс структуры данных очередь и вспомогательный класс узлов. В классе очереди реализованы методы для добавления элемента в конец очереди - push(), удаления и возврат элемента из начала очереди – pop(), возврат элемента из начала очереди без удаления – front(), возврат длины очереди – length(), удаления очереди – clear(), и метод выхода из программы – exit(). Функционал взаимодействия с очередью реализован через цикл while True с вводом команд через терминал.

# Тестирование

Тестирование задачи.

# Код программы

Код

# 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».

