|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |
| Институт Информационных Технологий | | |
| Кафедра Вычислительной техники | | |

|  |  |
| --- | --- |
| **ОТЧЕТ О ВЫПОЛНЕНИИ ПРАКТИЧЕСКОЙ РАБОТЫ №2**  «Простой калькулятор алгебраических выражений в обратной польской записи с целыми числами на Python» | |
|  | |
| **по дисциплине** | |
| **«**Теория формальных языков**»** | |
|  | |
| Выполнил студент группы ИКБО-04-20 | *Основин А.И.* |
| Принял ассистент | *Боронников А.С.* |

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | «20»\_сентября\_2023 г. |  |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2023 г. |  |

**1 Постановка задачи**

*Задание*: на любом языке программирования реализовать простой калькулятор алгебраических выражений в обратной польской записи с целыми числами. Поддерживаемые операции: сложение, вычитание, умножение, деление.

*Пример работы программы*:

ввод: 10 2 + 2 \*

вывод: 24

**2 Реализация программы**

Для реализации поставленной задачи использовался алгоритм перевода алгебраических выражений в *обратную польскую запись*, реализованный на языке программирования Python. Для корректного разделения введённых данных на порции используется функция main.

*Листинг 1 – Функция main*

|  |
| --- |
| def main(string: str) -> list:  substr = ""  in\_list = list()   for symbol in string:  if symbol in ["\*", "/", "+", "-", "(", ")"]:  if len(substr) != 0:  in\_list.append(substr)   in\_list.append(symbol)  substr = ""   else:  substr += symbol   if len(substr) != 0:  in\_list.append(substr)   return in\_list |

Код реализованного алгоритма состоит из функции transform, которая представлена в Листинге 2.

*Листинг 2 – Функция count*

|  |
| --- |
| def count(input: list, params: dict = {}) -> float:  stack = list()  for symbol in input:  if symbol.isdigit() or symbol in params.keys():  stack.append(symbol)  elif symbol in ["\*", "/", "+", "-"]:  a = stack.pop()  b = stack.pop()  if a.isalpha():  a = params[a]  if b.isalpha():  b = params[b]  stack.append(str(eval(b + symbol + a)))  else:  print(f"Can't parse symbol while counting: {symbol}.")  exit(0)  return float(stack.pop()) |

**3 Тестирования программы**

Ниже приведены примеры ввода данных в программу, вывод программы и правильный результат работы алгоритма

Таблица 1 – *тест программы*

|  |  |  |
| --- | --- | --- |
| Ввод | Вывод программы | Правильный ответ |
| 10 2 2 \* + | 14 | 14 |
| 6 8 2 / - | 2 | 2 |

**4 Вывод**

В ходе выполнения данной практической работы был изучен алгоритм обработки простейших алгебраических выражений в обратном польской записи. В результате проделанной работы была разработана программа, которая полностью соответствует приведенному алгоритму и отвечает поставленным требованиям.

**5 Список литературы**

1. Свердлов С. З. Языки программирования и методы трансляции: учебное пособие. – Санкт-Петербург: Лань, 2019. – 564 с.
2. Малявко А. А. Формальные языки и компиляторы: учебное пособие для вузов. – М.: Юрайт, 2018. – 429 с.
3. Миронов С. В. Формальные языки и грамматики: учебное пособие для студентов факультета компьютерных наук и информационных технологий. – Саратов: СГУ, 2019. – 80 с.
4. Антик М. И., Казанцева Л. В. Теория формальных языков в проектировании трансляторов: учебное пособие. – М.: МИРЭА, 2020. – 57 с.
5. Ахо А. В., Лам М. С., Сети Р., Ульман Дж. Д. Компиляторы: принципы, технологии и инструментарий. – М.: Вильямс, 2008. – 1184 с.
6. Ишакова Е.Н. Теория языков программирования и методов трансляции: учебное пособие. – Оренбург: ИПК ГОУ ОГУ, 2007. – 137 с.
7. Гинзбург С. Математическая теория контекстно-свободных языков. – М.: Мир, 1970. – 328 с.
8. Хантер Р. Основные концепции компиляторов. – М.: Вильямс, 2002. – 252 с.
9. Рейуорд-Смит В. Теория формальных языков. Вводный курс. – М.: Радио и связь, 1988. – 128 с.
10. Пратт Т. Языки программирования: разработка и реализация. – СПб.: Питер принт, 2002. – 688 с.