|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное автономное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)

КАФЕДРА «Информационная безопасность» (ИУ8)

Отчёт

по лабораторной работе № 1

по дисциплине «Программирование на Python»

**Тема: «Введение в Python»**

Выполнил: Бородин Глеб Викторович,

студент группы ИУ8-13М

Проверил: Зотов М.

г. Москва, 2025 г.

1. Цель работы

ЦЕЛЬ РАБОТЫ:

* Познакомиться с базовыми типами данных
* Познакомиться с базовыми операторами

2. Постановка задачи

**ЗАДАЧА 1:** Калькулятор среднего балла

Напишите программу, которая запрашивает у пользователя три оценки (целые числа), вычисляет их среднее арифметическое и выводит результат с точностью до двух знаков после запятой.

**ЗАДАЧА 2:** Обработка строки

Попросите пользователя ввести строку. Выведите:

* Длину строки
* Первый и последний символ
* Строку в верхнем регистре
* Строку, повторённую 3 раза

****ЗАДАЧА 3:**** Работа со списком

Создайте список из пяти произвольных элементов (чисел и строк). Выведите:

* Второй элемент списка
* Длину списка
* Последние три элемента
* Измените третий элемент на слово "Python"

****ЗАДАЧА 4:**** Уникальные элементы

Попросите пользователя ввести несколько чисел через пробел. Преобразуйте ввод в список, затем в множество. Выведите оригинальный список и множество (чтобы показать удаление дубликатов).

****ЗАДАЧА 5:**** Информация о студенте

Создайте словарь с ключами: имя, фамилия, возраст, курс. Заполните его данными (можно вручную или через input). Выведите все данные в виде:

Студент: Бородин Глеб

Возраст: 21

Курс: 1

****ЗАДАЧА 6:**** Проверка чётности

Программа запрашивает целое число. Определите, чётное оно или нет, и выведите True или False.

****ЗАДАЧА 7:**** Простой калькулятор

Запросите два числа и операцию (+, -, \*, /). Выполните операцию и выведите результат. Если пользователь ввёл неизвестный оператор — выведите ошибку.

3. Ход работы

ИСПОЛЬЗУЕТСЯ В ДАННОЙ РАБОТЕ: в данной работе реализуются функции для каждого задания и меню для выбора выполнения определенного задания. Функции вызываются в зависимости от введенного числа 1-7, чтоб выйти из меню без ошибки введите exit. Используются условные конструкции и циклы. Библиотеки не импортированы.

### ****МОЙ ХОД РАБОТЫ:****

### ****ШАГ 1. ОСОЗНАНИЕ ЗАДАЧИ 1****

****Задача:**** Необходимо создать программу, которая принимает от пользователя три целых числа (оценки), вычисляет их среднее арифметическое и выводит результат, округленный до двух знаков после запятой.

### ****ШАГ 2. ОПИСАНИЕ ДЕЙСТВИЙ (Задача 1)****

1. Запросить у пользователя ввод трех чисел.
2. Преобразовать введенные строки в целые числа.
3. Вычислить среднее арифметическое: сложить числа и разделить на их количество (3).
4. Отформатировать результат до двух десятичных знаков.
5. Вывести результат.

### ****ШАГ 3. РЕАЛИЗАЦИЯ, РЕЗУЛЬТАТ (Задача 1)****

****Реализация:**** Функция calculate\_sr(numbers) принимает список чисел, проверяет, не пустой ли он, и выводит среднее значение. В основном цикле ввод от пользователя разбивается на список и преобразуется в целые числа.  
****Результат:**** Программа корректно вычисляет и выводит средний балл. Пример: для ввода 5 4 5 выводится 4.67.

### ****ШАГ 1. ОСОЗНАНИЕ ЗАДАЧИ 2****

****Задача:**** Требуется принять строку от пользователя и выполнить с ней ряд операций: определить длину, вывести первый и последний символ, преобразовать к верхнему регистру и повторить строку три раза.

### ****ШАГ 2. ОПИСАНИЕ ДЕЙСТВИЙ (Задача 2)****

1. Запросить у пользователя ввод строки.
2. Используя встроенные функции Python:
   * len() для получения длины.
   * Индексацию [0] и [-1] для получения первого и последнего символа.
   * str.upper() для преобразования в верхний регистр.
   * Умножение строки \* 3 для повторения.
3. Вывести все результаты в удобочитаемом формате.

### ****ШАГ 3. РЕАЛИЗАЦИЯ, РЕЗУЛЬТАТ (Задача 2)****

****Реализация:**** Функция stringProcessing(input\_string) принимает строку, проверяет ее на пустоту и выводит все требуемые параметры, используя f-строку.  
****Результат:**** Программа полностью обрабатывает строку. Пример: для ввода hello выводится длина 5, символы h и o, HELLO и hellohellohello.

### ****ШАГ 1. ОСОЗНАНИЕ ЗАДАЧИ 3****

****Задача:**** Нужно создать список из 5 элементов (чисел и строк), а затем выполнить над ним операции: вывод второго элемента, длины, последних трех элементов, а также изменение третьего элемента на слово "Python".

### ****ШАГ 2. ОПИСАНИЕ ДЕЙСТВИЙ (Задача 3)****

1. Создать список из 5 произвольных элементов (например, [10, "apple", 30, "banana", 50]).
2. Вывести элемент с индексом 1 (второй по счету).
3. Вывести длину списка с помощью len().
4. Используя срез [-3:], вывести последние три элемента.
5. Присвоить элементу с индексом 2 (третий по счету) новое значение "Python" и вывести измененный список.

### ****ШАГ 3. РЕАЛИЗАЦИЯ, РЕЗУЛЬТАТ (Задача 3)****

****Реализация:**** Функция listProcessing(input\_list) принимает список, проверяет его и выводит требуемые данные. Для демонстрации изменения третьего элемента создается новый список через конкатенацию срезов.  
****Результат:**** Премамма успешно манипулирует списком. Пример: для списка [1, 2, 3, 4, 5] второй элемент — 2, длина — 5, последние три — [3, 4, 5], а после изменения список становится [1, 2, 'Python', 4, 5].

### ****ШАГ 1. ОСОЗНАНИЕ ЗАДАЧИ 4****

****Задача:**** Получить от пользователя последовательность чисел, разделенных пробелами, преобразовать ее в список, а затем в множество (которое автоматически удалит дубликаты), и показать оба варианта.

### ****ШАГ 2. ОПИСАНИЕ ДЕЙСТВИЙ (Задача 4)****

1. Запросить ввод чисел через пробел.
2. Разделить строку с помощью .split(), чтобы получить список строк.
3. Преобразовать каждый элемент полученного списка в целое число (или оставить строкой, если возможны и строки).
4. Преобразовать список в множество с помощью set().
5. Вывести исходный список и полученное множество.

### ****ШАГ 3. РЕАЛИЗАЦИЯ, РЕЗУЛЬТАТ (Задача 4)****

****Реализация:**** Функция uniqueElements(input\_list) выводит оригинальный список и множество, созданное из него.  
****Результат:**** Наглядно демонстрируется удаление дубликатов. Пример: для ввода 1 2 2 3 4 список будет ['1', '2', '2', '3', '4'], а множество — {'4', '1', '3', '2'} (порядок в множестве не сохраняется).

### ****ШАГ 1. ОСОЗНАНИЕ ЗАДАЧИ 5****

****Задача:**** Создать словарь с информацией о студенте (ключи: имя, фамилия, возраст, курс) и вывести эти данные в заданном формате.

### ****ШАГ 2. ОПИСАНИЕ ДЕЙСТВИЙ (Задача 5)****

1. Создать пустой словарь.
2. Заполнить словарь данными, запрашивая их у пользователя или задавая вручную.
3. Используя f-строку, вывести значения по соответствующим ключам в требуемом формате.

### ****ШАГ 3. РЕАЛИЗАЦИЯ, РЕЗУЛЬТАТ (Задача 5)****

****Реализация:**** Функция informationAboutStudents(input\_dict) ожидает готовый словарь и форматирует его вывод. В основном коде есть небольшая логическая ошибка: цикл предназначен для заполнения словаря несколькими студентами, но функция ожидает словарь с одним студентом. Корректнее было бы заполнить один словарь для одного студента.  
****Результат:**** При корректном заполнении словарь выводит аккуратную анкету. Пример: Студент: Иван Иванов\nВозраст: 20\nКурс: 2.

### ****ШАГ 1. ОСОЗНАНИЕ ЗАДАЧИ 6****

****Задача:**** Написать программу, которая определяет, является ли введенное пользователем целое число четным.

### ****ШАГ 2. ОПИСАНИЕ ДЕЙСТВИЙ (Задача 6)****

1. Запросить у пользователя целое число.
2. Использовать оператор деления по модулю % для проверки остатка от деления числа на 2.
3. Если остаток равен 0, число четное — вывести True. В противном случае вывести False.

### ****ШАГ 3. РЕАЛИЗАЦИЯ, РЕЗУЛЬТАТ (Задача 6)****

****Реализация:**** Функция checkEven(input\_number) реализует проверку через input\_number % 2 == 0 и выводит булево значение.  
****Результат:**** Программа корректно определяет четность. Пример: для 10 выводится True, для 7 — False.

### ****ШАГ 1. ОСОЗНАНИЕ ЗАДАЧИ 7****

****Задача:**** Реализовать простой калькулятор, который принимает два числа и оператор (+, -, \*, /), выполняет вычисление и обрабатывает ошибки (деление на ноль, неверный оператор).

### ****ШАГ 2. ОПИСАНИЕ ДЕЙСТВИЙ (Задача 7)****

1. Запросить у пользователя два числа (преобразовать в float) и символ операции.
2. Использовать условные операторы (if/elif/else) для выбора арифметического действия.
3. Для операции деления / добавить проверку, что второе число не равно нулю.
4. Если введен неизвестный оператор, вывести сообщение об ошибке.

### ****ШАГ 3. РЕАЛИЗАЦИЯ, РЕЗУЛЬТАТ (Задача 7)****

****Реализация:**** Функция simpleCalculator(num1, num2, operation) проверяет оператор и выполняет соответствующее действие, включая проверку деления на ноль.  
****Результат:**** Калькулятор работает корректно для основных операций и обрабатывает ошибки ввода. Пример: для 5 \* 3 выводится 15, для 10 / 0 — сообщение об ошибке.

### ****ШАГ 4. ВЕРНУТЬСЯ К ШАГУ 1 (ЗАДАЧИ ЗАВЕРШЕНЫ)****

Все 7 задач успешно реализованы в рамках одной интерактивной программы. Пользователь может выбирать задачу для выполнения через главное меню.

4. Заключение

В ходе работы были изучены типы данных: списки, строки, целые числа, с плавающей точкой, словари, булевые значения, функции, множества, map. Применялись функции: sum, len, if, else elseif, for, while, print, input и реализованные функции, преобразование типов, методы словарей. Было реализовано интерактивное меню, и функция под каждую задачу.

5. ПРИЛОЖЕНИЕ

Реализованный код располагается по следующей ссылке: ССЫЛКА НА РЕПОЗИТАРИЙ (ПРИ НАЛИЧИИ).

Листинг 1 – Код на Python с интерактивным меню и решением каждой задачи из 7.

|  |
| --- |
| def calculate\_sr(numbers):  if not numbers:  return print("No numbers provided")  return print(f"{sum(numbers) / len(numbers):.2f}")  def stringProcessing(input\_string):  if not input\_string:  return print("No input provided")  return print(f"""Processed string:\n  Length: {len(input\_string)}\n  First+Last characters: {input\_string[0]}{input\_string[-1]}\n  StringUp: {input\_string.upper()}\n  RepeatString3: {input\_string \* 3}""")  def listProcessing(input\_list):  if not input\_list:  return print("No input provided")  return print(f"""Processed list:  Second element: {input\_list[1]}  Length: {len(input\_list)}  Last 3 elements: {input\_list[-3:]}  Change 3rd element: {input\_list[:2] + ['Python'] + input\_list[3:]}""")  def uniqueElements(input\_list):  if not input\_list:  return print("No input provided")  return print(f"""Processed list:  List: {input\_list}  Set : {set(input\_list)}""")  def informationAboutStudents(input\_dict):  if not input\_dict:  return print("No input provided")  print(f"""Студент: {input\_dict['Имя'].strip()} {input\_dict['Фамилия'].strip()}  Возраст: {input\_dict['Возраст'].strip()}  Курс: {input\_dict['Курс'].strip()}""")  return  def checkEven(input\_number):  if input\_number % 2 == 0:  return print(f"True")  else:  return print(f"False")  def simpleCalculator(num1, num2, operation):  if operation == "+":  return print(f"Result: {num1 + num2}")  elif operation == "-":  return print(f"Result: {num1 - num2}")  elif operation == "\*":  return print(f"Result: {num1 \* num2}")  elif operation == "/":  if num2 != 0:  return print(f"Result: {num1 / num2}")  else:  return print("Error: Division by zero is not allowed.")  else:  return print("Error: Invalid operation.")  while True:  task = input("Enter task (1-7) or 'exit' to quit: ")  if task == "1":  numbers = list(map(int, input("Enter numbers separated by spaces: ").split()))  calculate\_sr(numbers)  elif task == "2":  input\_string = input("Enter a string: ")  stringProcessing(input\_string)  elif task == "3":  input\_list = list(input("Enter a list of elements separated by spaces: ").split())  listProcessing(input\_list)  elif task == "4":  input\_list = list(input("Enter a list of elements separated by spaces: ").split())  uniqueElements(input\_list)  elif task == "5":  input\_dict = {}  while True:  entry = input("Enter student name and grade (or 'done' to finish): ")  if entry == "done":  break  name, grade = entry.split()  input\_dict[name] = grade  informationAboutStudents(input\_dict)  elif task == "6":  input\_number = int(input("Enter a number: "))  checkEven(input\_number)  elif task == "7":  num1 = float(input("Enter first number: "))  num2 = float(input("Enter second number: "))  operation = input("Enter operation (+, -, \*, /): ")  simpleCalculator(num1, num2, operation)  elif task == "exit":  break  else:  print("Invalid task. Please try again.") |