Федеральное государственное бюджетное образовательное учреждение высшего образования «Байкальский государственный университет»

Прикладная информатика

Объектно-ориентированное программирование

Лабораторная работа №1.2

Вариант: 3

Выполнил студент

Вайкус Яков Михайлович

Группа № ИС-23-1

Преподаватель: Федотов Никита Андреевич

г. Иркутск

2025

Оглавление

[**Задание** 3](#_Toc207876137)

[**Код программы** 3](#_Toc207876138)

[Вывод программы: 4](#_Toc207876139)

# **Задание**

Вариант 29

Задание:

Напишите программу в Colab, которая выполняет следующие шаги:

1. Пользователь вводит текстовую строку, содержащую несколько слов.

2. Программа должна:

a. Разделить строку на слова и создать список слов.

b. Создать новый список, где каждое слово записано в виде: каждая

буква в верхнем регистре через пробел (например, "Python" → "P

Y T H O N").

3. Подсчитайте:

a. количество слов в исходной строке,

b. количество символов в новом списке слов (включая пробелы между

буквами).

4. Создайте строку, объединяя все слова нового списка через |.

5. Создайте кортеж, содержащий исходный список слов, новый список слов,

количество слов и количество символов.

6. Выведите тип всех переменных, список, строку и кортеж.

7. Проверка: если количество символов нового списка > 20, выведите True,

иначе False.

Требования:

• Использовать str, list, tuple, int, bool.

• Обязательны: строковые методы, циклы, списки, кортежи, логические

выражения.

• Студент должен самостоятельно реализовать преобразование регистра и

вставку пробелов между буквами.

# **Код программы**

a = '5.5'

b = 3

c = '2'

res = (float(a) \*\* b) + complex(c)

real\_str = str(res.real)

real\_str = real\_str.replace('.', ':')

substr = real\_str[1:-1]

print(substr)

print(len(substr))

s = "АбстракцияПолиморфизмИнкапсуляцияНаследование"

w1 = s[0:10]

w2 = s[10:21]

w3 = s[21:34]

w4 = s[34:]

s\_with\_sep = w1 + ";" + w2 + ";" + w3 + ";" + w4

print(s\_with\_sep)

def fmt(word):

    return word[0].upper() + word[1:].lower()

new\_s = fmt(w3) + fmt(w1) + fmt(w4) + fmt(w2)

print(new\_s)

import string

user\_input = input("Введите три значения через ;: ")

values = user\_input.split(";")

parsed = []

types = []

for v in values:

    v = v.strip()

    try:

        if '.' in v:

            parsed.append(float(v))

            types.append(float)

        else:

            parsed.append(int(v))

            types.append(int)

    except ValueError:

        parsed.append(v)

        types.append(str)

if all(t == str for t in types):

    # Все строки

    combined = "".join(reversed(parsed))

    vowels = "aeiouаеёиоуыэюяAEIOUАЕЁИОУЫЭЮЯ"

    result = "".join(ch for ch in combined if ch not in vowels)

    print(result)

elif types.count(int) + types.count(float) == 2 and types.count(str) == 1:

    # Два числа и строка

    nums = [p for p in parsed if isinstance(p, (int,float))]

    txt = [p for p in parsed if isinstance(p, str)][0]

    geom\_mean = (nums[0]\*nums[1]) \*\* 0.5

    result = f"{geom\_mean}-{txt}".upper()

    print(result)

elif set(types) != {str} and not (types.count(int) + types.count(float) == 2 and types.count(str) == 1):

    # Смешанные иначе

    s = "&".join(str(p) for p in parsed)

    s = "".join(ch for ch in s if ch not in string.punctuation)

    result = s[::-1]

    print(result)

else:

    # Ни одно условие не подошло

    total\_chars = sum(len(str(p)) for p in parsed)

    print(total\_chars)

d, e, f = 18, 7, 13

expr = (d ^ e) << (f % 3)

print("Значение выражения:", expr)

check\_val = (d & e) \* (f | 5)

is\_power\_of\_two = (expr & (expr - 1) == 0) and expr > 0

cond = (is\_power\_of\_two or expr > check\_val) and expr % 5 != 0

print("Результат проверки:", cond)

# **Вывод программы:**

68:37

5

Абстракция;Полиморфизм;ИнкапсуляцияН;аследование

ИнкапсуляциянАбстракцияАследованиеПолиморфизм

Введите три значения через ;: rgerhtr;5;4

4.47213595499958-RGERHTR

Значение выражения: 42

Результат проверки: True