

МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВВГУ»)  
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ  
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

ОТЧЕТ  
ПО ЛАБОРАТОРНОЙ РАБОТЕ №4  
по дисциплине  
«Информатика и программирование»

Студент  
гр. БИН-25-2 \_\_\_\_\_ Н.Д. Ананко  
Ассистент  
преподавателя \_\_\_\_\_ М.В. Водяницкий

## Задание

Выполнить задания и оформить отчет по СТО ВВГУ.

**Задание 1.** Написать программу, которая определяет, как будет вести себя кондиционер. Если температура в помещении 20 градусов и выше, то кондиционер выключается, если меньше - включается. Температура вводится пользователем с консоли.

Пример:

Введите температуру: 18

Кондиционер включен

**Задание 2.** Год делится на четыре сезона: зима, весна, лето и осень. Написать программу, которая запрашивает у пользователя номер месяца и выводит к какому сезону этот месяц относится.

Пример:

Введите номер месяца: 4

Это весна

**Задание 3.** Считается, что один год, прожитый собакой, эквивалентен семи человеческим годам. При этом зачастую не учитывается, что собаки становятся абсолютно взрослыми уже к двум годам. Таким образом, многие предпочитают каждый из первых двух лет жизни собаки приравнивать к 10.5 годам человеческой жизни, а все последующие к 4.

Написать программу, которая будет переводить собачий возраст в человеческий. Программа должна корректно обрабатывать входные данные и выводить соответствующие сообщения об ошибках:

- Если вводится не число
- Если вводится число меньше 1
- Если вводится число большее 22

Пример:

Введите возраст собаки (в годах): 5

Возраст собаки в человеческих годах: 33.0

Пример:

Введите возраст собаки (в годах): 0

Ошибка: возраст должен быть не меньше 1

**Задание 4.** Число делиться на 6 только в случае соблюдения двух условий:

- Последняя цифра четная

- Сумма всех цифр делиться на 3

Написать программу, которая выведет делиться ли введенное число на 6 или нет.

**Задание 5.** Написать программу, которая будет проверять пароль на надежность.

Пароль считается надежным, если его длина не менее 8 символов и если он содержит:

- Заглавные буквы латиницы
- Строчные буквы латиницы
- Числа
- Специальные знаки

В случае, если пароль не проходит по одному из условий, необходимо сообщить пользователю каким именно условиям он не удовлетворяет.

Пример:

Введите пароль: qwerty

Пароль ненадежный: отсутствуют заглавные буквы, числа и специальные символы

**Задание 6.** Написать программу, которая определяет, является ли введенный пользователем год високосным. Год считается високосным, если он делится на 4, но не делится на 100, либо если он делится на 400.

Пример:

Введите год: 2024

2024 - високосный год

**Задание 7.** Написать программу, которая запрашивает у пользователя три числа и выводит на экран наименьшее из них. При решении нельзя использовать встроенные функции min() и max().

Пример:

Введите три числа: 8 3 5

Наименьшее число: 3

**Задание 8.** В магазине проводится акция. Акция работает по следующим правилам:

- Сумма < 1000 => скидка - 0%
- Сумма < 5000 => скидка - 5%
- Сумма < 10000 => скидка - 10%
- Сумма > 10000 => скидка - 15%

Напишите программу, которая запрашивает сумму покупки и выводит размер скидки и итоговую сумму к оплате.

Пример:

Введите сумму покупки: 7500

Ваша скидка: 10% оплате : 6750.0

**Задание 9.** Написать программу, которая определяет время суток по введенному часу (целое число от 0 до 23).

- С 0 до 5 часов - ночь
- С 6 до 11 часов - утро
- С 12 до 17 часов - день
- С 18 до 23 часов - вечер

Пример:

Введите час (0–23): 20

Сейчас вечер

**Задание 10.** Написать программу, которая определяет, является ли введенное число простым. Число называется простым, если оно больше 1 и делится только на 1 и само себя. Программа должна корректно обрабатывать некорректный ввод и выводить соответствующие сообщения об ошибках.

Примеры:

Введите число: 17

17 - простое число

Введите число: 12

12 - составное число

## Содержание

1 Выполнение работы .....	3
1.1 Задание 1 .....	3
1.2 Задание 2 .....	4
1.3 Задание 3 .....	5
1.4 Задание 4 .....	6
1.5 Задание 5 .....	7
1.6 Задание 6 .....	9
1.7 Задание 7 .....	10
1.8 Задание 8 .....	11
1.9 Задание 9 .....	12
1.10 Задание 10.....	13

## 1 Выполнение работы

### 1.1 Задание 1

Код :

```
t = int(input("введите температуру"))
if t >= 20:
    print("кондиционер выключен!")
else:
    print("кондиционер включен!")
```

**Алгоритм работы:**

1. Пользователь вводит текущую температуру как целое число
2. Если температура  $\geq 20^{\circ}\text{C}$  - выводится сообщение о выключенном кондиционере
3. Если температура  $< 20^{\circ}\text{C}$  - выводится сообщение о включенном кондиционере

**Ключевые элементы кода:**

- `input("введите температуру")` - запрос данных с текстовой подсказкой
- `int()` - преобразование ввода в целое число
- `if t >= 20:` - условие сравнения температуры с пороговым значением

**Логика работы:**

Программа реализует пороговую систему управления: при достижении температуры  $20^{\circ}\text{C}$  и выше кондиционер считается выключенным, при более низких температурах - включенным.

## 1.2 Задание 2

Код :

```
mesiac = int(input("Введите номер месяца (от 1 до 12):"))
if mesiac == 12 or 1 <= mesiac <= 2:
    print("Зима")
elif 3 <= mesiac <= 5:
    print("Весна")
elif 6 <= mesiac <= 8:
    print("Лето")
elif 9 <= mesiac <= 11:
    print("Осень")
else:
    12 < mesiac
    mesiac < 1
    print("ты ввел ерунду")
```

### Назначение программы:

Определение времени года по номеру месяца.

### Алгоритм работы:

1. Пользователь вводит номер месяца (1-12)
2. Программа проверяет принадлежность месяца к сезонам:
  - 12, 1, 2 → "Зима"
  - 3, 4, 5 → "Весна"
  - 6, 7, 8 → "Лето"
  - 9, 10, 11 → "Осень"
3. При вводе неверного числа выводится сообщение об ошибке

### Ключевые элементы кода:

- `input()` - ввод номера месяца
- `int()` - преобразование в целое число
- Цепочка условий `if-elif-else`
- Логические операторы `or` и диапазоны `<=`
- Обработка некорректного ввода

### Особенности:

- Зима включает декабрь (12) и январь-февраль (1-2)
- Используется понятная логика диапазонов месяцев
- Есть проверка на ошибочный ввод

### 1.3 Задание 3

Код :

```

try:
    dog_age = float(input("Введите возраст собаки (в
годах):"))

    if dog_age < 1:
        print("ошибка! Возраст должен быть не меньше 1!")
    elif dog_age > 22:
        print("ошибка! Возраст должен быть не больше 22!")
    else:
        if dog_age <= 2:
            human_age = dog_age * 10.5
        else:
            human_age = 2 * 10.5 + (dog_age - 2) * 4

        print(f"Возраст собаки в человеческих годах:
{human_age}")
except ValueError:
    print("Ошибка! Введите число!")

```

#### **Назначение программы:**

Конвертация возраста собаки в человеческий эквивалент с обработкой ошибок.

#### **Алгоритм работы:**

- Программа запрашивает возраст собаки и преобразует в число
- Проверяет корректность введенного возраста:
- Если меньше 1 года - ошибка
- Если больше 22 лет - ошибка (но есть логическая ошибка в условии)
- Вычисляет человеческий возраст по формуле:
- До 2 лет: каждый год = 10.5 человеческих лет
- После 2 лет: первые 2 года = 21 год + остальные годы × 4

#### **Ключевые элементы кода:**

- try-except - обработка исключений при вводе
- float() - преобразование в вещественное число
- Вложенные условия if-elif-else
- Форматированная строка f"{{variable}}"
- Формула пересчета собачьего возраста в человеческий

#### **Особенности:**

- Защита от некорректного ввода (не числа)
- Две разные формулы расчета для разных возрастных периодов

- Проверка на минимальный возраст

- **Обнаруженная ошибка:**

В коде дублируется условие `dog_age < 1` в двух первых `if`, что делает проверку максимального возраста (22 года) недостижимой.

**Формула расчета:**

- Для собак до 2 лет: Человеческий\_возраст = Возраст\_собаки  $\times 10.5$

## 1.4 Задание 4

Код:

```
number = input("Введите число: ")

if not number.isdigit():
    print("Ошибка: введите целое положительное число")
else:
    last_digit = int(number[-1])
    digit_sum = sum(int(digit) for digit in number)

    if last_digit % 2 == 0 and digit_sum % 3 == 0:
        print(f"Число {number} делится на 6")
    else:
        print(f"Число {number} не делится на 6")
```

**Назначение программы:**

Проверка делимости числа на 6 без использования операции деления.

**Алгоритм работы:**

- Программа получает ввод от пользователя как строку
- Проверяет, состоит ли строка только из цифр
- Если ввод корректен:
- Определяет последнюю цифру числа
- Вычисляет сумму всех цифр числа
- Проверяет условия делимости на 6:
  - Четность (делимость на 2) - по последней цифре
  - Делимость на 3 - по сумме цифр
- Выводит результат проверки

**Ключевые элементы кода:**

- `isdigit()` - проверка, что строка содержит только цифры
- `number[-1]` - получение последнего символа строки
- Генератор списка `int(digit) for digit in number`
- `sum()` - вычисление суммы цифр
- Комбинация условий `% 2 == 0` и `% 3 == 0`

### **Математическая логика:**

Число делится на 6, если оно делится одновременно на 2 и на 3:

**На 2:** последняя цифра четная (0, 2, 4, 6, 8)

**На 3:** сумма всех цифр числа делится на 3

### **Особенности:**

- Работает с числами любой длины
- Обработка некорректного ввода
- Не использует математическое деление на 6
- Эффективный алгоритм проверки

## 1.5 Задание 5

Код :

```
import string

password = input("Введите пароль: ")
errors = []

if len(password) < 8:
    errors.append("длина менее 8 символов")

if not any(char.isupper() for char in password):
    errors.append("отсутствуют заглавные буквы")

if not any(char.islower() for char in password):
    errors.append("отсутствуют строчные буквы")

if not any(char.isdigit() for char in password):
    errors.append("отсутствуют числа")

special_chars = set(string.punctuation) #
if not any(char in special_chars for char in password):
    errors.append("отсутствуют специальные символы")

if errors:
    error_list = ", ".join(errors)
    print(f"Пароль ненадежный: {error_list}")
else:
    print("Пароль надежный")
```

### **Назначение программы:**

Комплексная проверка надежности пароля по нескольким критериям безопасности.

### **Алгоритм работы:**

- Программа получает пароль от пользователя
- Последовательно проверяет 5 критериев надежности:
  - Длина не менее 8 символов
  - Наличие заглавных букв
  - Наличие строчных букв
  - Наличие цифр
  - Наличие специальных символов
- Собирает все найденные ошибки в список
- Выводит либо перечень ошибок, либо сообщение о надежности пароля

### **Ключевые элементы кода:**

- `string.punctuation` - встроенный набор специальных символов
- `errors = []` - список для накопления ошибок
- `any()` с генераторами для проверки условий
- Методы строк: `islower()`, `isupper()`, `isdigit()`
- `",".join(errors)` - форматированный вывод ошибок

### **Критерии надежности пароля:**

- **Длина:**  $\geq 8$  символов
- **Регистр:** есть и заглавные, и строчные буквы
- **Цифры:** минимум одна цифра
- **Спецсимволы:** !"#\$%&'()\*,-./;:<=>?@[{}]^\_`{|}~

### **Особенности:**

- Детализированный отчет об ошибках
- Модульная структура проверок
- Использование встроенных функций Python для эффективности
- Четкое разделение логики проверки и вывода результатов
- `_возраст = 21 + (Возраст_собаки - 2) * 4`

## 1.6 Задание 6

Код :

```
year = int(input("Введите год: "))

if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
    print(f"{year} - високосный год")
else:
    print(f"{year} - не високосный год")
```

**Назначение программы:**

Определение, является ли введенный год високосным.

**Алгоритм работы:**

- Программа получает год от пользователя и преобразует в целое число
- Проверяет условия високосности по григорианскому календарю:
  - Год делится на 4 И НЕ делится на 100
  - ИЛИ год делится на 400
- Выводит соответствующее сообщение с указанием года

**Ключевые элементы кода:**

- `int(input())` - получение и преобразование ввода
- Сложное логическое условие с `and`, `or`, `!=`
- Операторы остатка от деления `%`
- Форматированный вывод с `f`-строкой

**Математические правила високосности:**

- **Високосный:**  $(\text{год} \div 4 = \text{целое}) \text{ И } (\text{год} \div 100 \neq \text{целое})$
- **Високосный:**  $\text{год} \div 400 = \text{целое}$
- **Не високосный:** все остальные случаи

**Примеры:**

$2024 \div 4 = 506$  (целое),  $2024 \div 100 \neq$  целое  $\rightarrow$  **високосный**

$1900 \div 4 = 475$  (целое), но  $1900 \div 100 = 19$  (целое)  $\rightarrow$  **не високосный**

$2000 \div 400 = 5$  (целое)  $\rightarrow$  **високосный**

## 1.7 Задание 7

Код :

```
a, b, c = map(int, input("Введите три числа: ").split())
if a < b:
    smallest = a
else:
    smallest = b
if c < smallest:
    smallest = c
print(f"Наименьшее число: {smallest}")
```

**Назначение программы:**

Нахождение наименьшего числа среди трех введенных значений.

**Алгоритм работы:**

- Программа получает три числа в одной строке, разделенные пробелами
- Преобразует введенные строки в целые числа
- Последовательно сравнивает числа:
- Сначала находит меньшее между a и b
- Затем сравнивает полученный результат с c
- Выводит наименьшее из трех чисел

**Ключевые элементы кода:**

- `input().split()` - разбиение ввода на список строк
- `map(int, ...)` - преобразование всех элементов в целые числа
- Множественное присваивание `a, b, c =`
- Последовательные сравнения через временную переменную `smallest`
- Поэтапный подход к нахождению минимума

**Логика сравнения:**

- **Шаг 1:** `smallest = min(a, b)`
- **Шаг 2:** `smallest = min(smallest, c)`
- **Результат:** наименьшее из трех чисел

**Особенности:**

- Эффективный алгоритм без использования встроенной функции `min()`
- Работа с одним вводом для трех чисел
- Четкая последовательность операций
- Использование промежуточной переменной для хранения текущего минимума

## 1.8 Задание 8

Код :

```

amount = float(input("Введите сумму покупки: "))
if amount < 1000:
    discount_percent = 0
    discount_name = "0%"
elif 1000 <= amount <= 5000:
    discount_percent = 5
    discount_name = "5%"
elif 5000 < amount <= 10000:
    discount_percent = 10
    discount_name = "10%"
else:
    discount_percent = 15
    discount_name = "15%"

discount_amount = amount * discount_percent / 100
final_amount = amount - discount_amount

print(f"Сумма покупки: {amount:.2f} руб.")
print(f"Ваша скидка: {discount_name}")
    ({discount_amount:.2f} руб.)")
print(f"К оплате: {final_amount:.2f} руб.")

```

**Назначение программы:**

Расчет скидки на основе суммы покупки и вывод итогового чека.

**Алгоритм работы:**

- Программа получает сумму покупки от пользователя
- Определяет размер скидки по прогрессивной шкале:
- До 1000 руб. → скидка 0%
- 1000-5000 руб. → скидка 5%
- 5001-10000 руб. → скидка 10%
- Свыше 10000 руб. → скидка 15%
- Рассчитывает сумму скидки и итоговую сумму к оплате
- Форматирует и выводит детализированный чек

**Ключевые элементы кода:**

- `float(input())` - получение суммы с возможностью дробных значений
- Цепочка условий `if-elif-else` для определения скидки
- Параллельное хранение процента и названия скидки
- Математические расчеты: скидка = сумма × процент ÷ 100
- Форматирование чисел `:.2f` для вывода с двумя decimal places\*

**Decimal places\*** (десятичные знаки) - это цифры, которые находятся после десятичной точки в дробном числе.

### Шкала скидок:

- 0-999 руб.: Без скидки (0%)
- 1000-5000 руб.: Базовая скидка (5%)
- 5001-10000 руб.: Средняя скидка (10%)
- 10000+ руб.: Максимальная скидка (15%)

### 1.9 Задание 9

Код :

```
hour = int(input("Введите час (0-23): "))
if hour < 0 or hour > 23:
    print("Ошибка: введите число от 0 до 23")
elif hour <= 5:
    print("Сейчас ночь")
elif hour <= 11:
    print("Сейчас утро")
elif hour <= 17:
    print("Сейчас день")
else:
    print("Сейчас вечер")
```

### Назначение программы:

Определение времени суток по введенному часу.

### Алгоритм работы:

- Программа получает час от пользователя (0-23)
- Проверяет корректность ввода (число от 0 до 23)
- Определяет время суток по диапазонам:
- 0-5 часов → Ночь
- 6-11 часов → Утро
- 12-17 часов → День
- 18-23 часов → Вечер
- Выводит соответствующее время суток

### Ключевые элементы кода:

- `int(input())` - получение и преобразование часа
- Проверка допустимого диапазона `0 <= hour <= 23`
- Цепочка условий `if-elif-else` с последовательными проверками

- Логика "накопления" условий (каждое следующее условие предполагает, что предыдущие ложны)

### **Распределение времени суток:**

- **Ночь:** 00:00 - 05:59
- **Утро:** 06:00 - 11:59
- **День:** 12:00 - 17:59
- **Вечер:** 18:00 - 23:59

### **Особенности:**

- Обработка некорректного ввода
- Использование 24-часового формата
- Последовательная проверка условий от меньшего к большему
- Четкое разделение суток на 4 равных периода

## **1.10 Задание 10**

Код :

```
try:
    number = int(input("Введите число: "))

    if number < 2:
        print(f"{number} - не является простым числом")
    else:
        is_prime = True
        for i in range(2, number):
            if number % i == 0:
                is_prime = False
                break

        if is_prime:
            print(f"{number} - простое число")
        else:
            print(f"{number} - составное число")

except ValueError:
    print("Ошибка: введите целое число!")
```

### **Назначение программы:**

Проверка, является ли введенное число простым или составным.

### **Алгоритм работы:**

- Программа получает число от пользователя и преобразует в целое
- Обрабатывает возможные ошибки ввода (не числа)
- Проверяет базовый случай: числа меньше 2 не являются простыми

- для чисел  $\geq 2$  проверяет делимость на все числа от 2 до (number-1)
- Если найден делитель - число составное, иначе - простое

### Ключевые элементы кода:

- try-except - обработка некорректного ввода
- for i in range(2, number) - перебор возможных делителей
- number % i == 0 - проверка делимости
- Флаг is\_prime - отслеживание состояния числа
- break - оптимизация (прекращение проверки при найденном делителе)

### Математическая логика:

- **Простое число:** делится только на 1 и на само себя
- **Составное число:** имеет делители кроме 1 и самого себя
- **Особые случаи:** 0 и 1 не являются простыми числами

### Критерии проверки:

- Числа  $< 2$ : не простые
- Числа  $\geq 2$ : проверка делимости на диапазоне от 2 до (n-1)