

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

Ассистент

преподавателя

Р.А. Прокудин

М.В. Водяницкий

Задание

Выполнить задания на Python и оформить отчет по стандартам 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 должна вводиться пользователем с консоли.

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.

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 приравнять 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

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.

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.

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()`.

Задание 8. В магазине проводится акция. Акция работает по следующим правилам, изображенным в таблице 1:

Таблица 1

Сумма покупки	Скида
до 1000	0%
1000-5000	5%
5000-10000	10%
более 10000	15%

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

Задание 9. Написать программу, которая определяет время суток по введенному часу (целое число от 0 до 23). См. Таблицу 2

Таблица 2

Время	Период
0-5	Ночь
6-11	Утро
12-17	День
18-23	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.

Содержание

1 Выполнение работ

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

1 Выполнение работы

1.1 Задание 1

В данном задании была создана переменная, которая хранит ввод пользователя о температуре помещения. Это было сделано для определения состояния кондиционера. После с помощью команды `print()`, `if` и `else` были выведен на экран состояния кондиционера. На рисунке 1 представлен код полученной программы.

```
1 a = int(input("Введите температуру в помещении: "))
2 print("Кондиционер включен" if a < 20 else "Кондиционер выключен")
```

Рисунок 1 — Листинг программы для задания 1

Пояснение работы программы:

1) 1 строка кода запрашивает у пользователя температуру помещения и преобразовывает её в целочисленные значения, после сохраняет это значение в переменную `a`.

2) 2 строка кода выводит сообщение “Кондиционер включен” или “Кондиционер выключен” соответствующая условия находящееся в функции `print()`, которое выводит “Кондиционер включен” если `a < 20` иначе “Кондиционер выключен”.

1.2 Задание 2

В данном задании была создана переменная, которая хранит ввод пользователя о номере месяца. А также был создан массив, который хранил в себе времена года. И после с помощью команды `print()` было выведено название месяца соответствующему номеру, который был введён пользователем до этого. Ниже на рисунке 2 представлен код полученной программы.

```
1 num = int(input("Введите номер месяца: "))
2 m = ["зима", "весна", "лето", "осень", "зима"]
3 print(m[num // 3])
```

Рисунок 2 — Листинг программы для задания 2

Пояснение работы программы:

1) 1 строка кода запрашивает у пользователя номер месяца и после преобразовывает в целочисленное значение и сохраняет это значение в переменную `num`

2) 2 строка кода создаёт массив в переменную `m`, который хранил в себе названия времён года и при этом зима дублируется. Пояснение на следующем шаге.

3) 3 строка выводит значение через индекс `num` делённая на цело на 3 с помощью функции `print()`. Именно из-за этого в массиве `m` две зимы, так как `12 // 3` равно 4, и как раз по 4 индексу в массиве `m` это зима.

1.3 Задание 3

В данном задании была создана функция, которая переводит возраст собаки по человеческим меркам. После был создан бесконечный цикл для обработки ввода пользователя, чтобы регулировать не корректный ввод пользователя. На рисунке 3 представлен следующий код полученной программы.

```

1  ✓ def OldD(o):
2      |     if o <= 2: return 10.5 * o
3      |     if o > 2: return 21 + ((o - 2) * 4)
4
5  ✓ while True:
6      |     otvet = input("Введите возраст собаки в годах:")
7  ✓      |     if otvet.isdigit():
8  ✓          |         if int(otvet) < 1 or int(otvet) > 22:
9              |             print("Ошибка: нужно ввести возраст от 1 до 22 включительно")
10 ✓          |         else:
11              |             print(OldD(int(otvet)))
12              |             break
13 ✓      |     else:
14          |         print("Ошибка: Введите возраст используя цифры (:")

```

Рисунок 3 — Листинг программы задания 3

Пояснение работы программы:

1) 1-3 строка кода это создание функции OldD, с помощью функции def(), которая переводит возраст собаки на человеческие мерки. И после возвращает число, а именно возраст собаки в человеческих мерках. Используя if и return.

2) 5 строка создаёт бесконечный цикл с помощью цикла while и значения True в условии, что делает этот цикл бесконечным.

3) 6 строка кода запрашивает у пользователя возраст собаки и сохраняет возраст в строковое значение в переменную otvet.

4) 7 строка кода это условие, которое проверяет является ли переменная otvet числом или не является числом.

5) 8-11 строка кода делает проверки на корректность числа и выводит ошибку в случае если число меньше 1 или больше 22 и в обратном случае выводит возраст собаки в человеческих мерках.

6) 12 строка кода прерывает бесконечный цикл, с помощью команды break, ведь пользователь получил результат всей программы.

7) 13-14 строка выводят ошибку, ведь пользователь не ввел число, а что-то другое. Срабатывают эти строки когда 7 строка возвращает False, то есть когда пользователь не ввёл число.

1.4 Задание 4

В данном задании была создана переменная, которая хранит число, в строковом значении, которое ввел пользователь. После была рассчитана сумма числа и в конце проводилась проверка делиться ли введенное число на шесть или нет. Если число делиться на шесть то сумма цифр числа делиться на три и если последнее число четное иначе число не делиться на шесть. На рисунки 4 представлен код полученной программы для задания номер 4.

```

1  a = input("Введите число для проверки делимости на 6: ")
2  sm = 0
3  √ for i in a:
4  |      sm += int(i)
5
6  √ if sm % 3 == 0 and int(a[-1]) % 2 == 0:
7  |     print("Делиться;")
8  √ else:
9  |     print("Не делиться-_-")

```

Рисунок 4 — Листинг программы задания 4

Пояснение работы программы:

- 1) 1 строка кода запрашивает у пользователя число и сохраняет это число в переменную a.
- 2) 2 строка кода создает переменную sm и присваивает ей число 0.
- 3) 3-4 строка кода считает сумму цифр в числе, которое хранится в переменной a. Для этого используется цикл for с переменной i, где i это цифры числа a.
- 4) 6-9 строка кода проверяет делиться ли число на 6 или нет и выводит соответствующие результаты. Делиться если сумма цифр в числе делиться на 3 и если последняя цифра четная. Не делиться если условие не подходит к числу.

1.5 Задание 5

В данном задании была создана переменная, которая хранит пароль пользователя в строковом значении. Также была создана переменная для маски пароля, чтобы узнать какие виды переменных находиться в пароле. После каждый символ пароля обрабатывался и добавлялся вид этого символа в маску. Далее обрабатывалась маска и выводился ответ, который выводит является ли пароль надежным или нет. На рисунке 5 представлен код полученной программы.

```

1  p = input("Введите пароль: ")
2  mask = ""
3  if len(p) >= 8:
4      for i in range(len(p)):
5          if 96 < ord(p[i]) < 123:
6              mask += "a"
7          elif 64 < ord(p[i]) < 91:
8              mask += "A"
9          elif p[i].isdigit():
10             mask += "0"
11         else:
12             mask += "_"
13
14  otvet = []
15
16  if "a" not in mask:
17      otvet.append("строчные буквы")
18  if "A" not in mask:
19      otvet.append("заглавные буквы")
20  if "0" not in mask:
21      otvet.append("числа")
22  if "_" not in mask:
23      otvet.append("специальные символы")
24
25  print("Надежный пароль" if len(otvet) == 0 else "Ненадежный пароль отсутствуют: " + ", ".join(otvet))

```

Рисунок 5 — Листинг программы задания 5

Пояснение работы программы:

- 1) 1 строка кода запрашивает и сохраняет пароль пользователя.
- 2) 2 строка создает переменную `mask`, которая на данный момент хранит в себе пустую строку.
- 3) 3 строка проверяет больше или равна длина пароля 8 или нет.
- 4) 4-12 строка кода перебирает каждый символ пароля и проверяя его вид а именно является ли символ заглавной буквой, строчной буквой, числом или специальным символом. После соответствующем символом добавляется в маску.
- 5) 14 строка кода создает пустой список.
- 6) 15-23 проверяют не содержится ли какой либо вид символов в маске или нет. Если не содержится то в список под названием `otvet` добавляется названия вида символа, который отсутствует в данной маске.
- 7) 25 строка выводит пароль надежный если длина списка `otvet` равна 0 иначе выводит, что пароль не надежный и какие виды символов отсутствуют в пароле.

1.6 Задание 6

В данном задании была создана переменная, которая хранит ввод пользователя о номере года. И после была проверка является ли введенный год високосным или нет и после выводился соответствующий ответ. На рисунки 6 представлен следующий код программы.


```

1  g = int(input("Введите год:"))
2  if ((g % 4 == 0) and (g % 100 != 0)) or (g % 400 == 0):
3      print(f"{g} - високосный год")
4  else:
5      print(f"{g} - невисокосный год")

```

Рисунок 6 — Листинг программы задания 6

Пояснение работы программы:

1) 1 строка кода запрашивает у пользователя номер года и преобразовывает этот номер в целочисленное значение и сохраняет его в переменную `g`.

2) 2-5 строка кода делает проверку является ли год високосным или нет. Если год делиться на четыре и при этом год не делиться на 100 или если год делиться на 400, то год високосный иначе год невисокосный. После выводиться соответствующий ответ.

1.7 Задание 7

В данном задании была создана переменная, которая хранит в себе 3 числа от пользователя. После был создан и сортирован список. И в конце выводился последний элемент данного списка. На рисунке 7 представлен код программы.

```

1  ch = input("Введите три числа через пробел: ")
2  a = sorted([int(i) for i in ch.split()])
3  print(a[0])

```

Рисунок 7 — Листинг программы задания 7

Пояснение работы программы:

1) 1 строка кода запрашивает у пользователя три числа через пробел и сохраняет это строковое значение в переменную `ch`.

2) 2 строка кода создает список с помощью цикла `for`, где `i` это числа из списка, где список это `ch` с командой `split()`, которая создает из строкового значения список, где элементы этого списка числа, которые были разделены пробелами. И каждый элемент преобразовывался в целочисленное значение с помощью команды `int()` и добавлялся в первоначальный список. Далее этот список сортировался по возрастанию с помощью команды `sorted()` и после этот список сохранялся в переменную `a`.

3) 3 строка кода выводит первый элемент списка `a`, с помощью команды `print()`. И так как список сортирован по возрастанию, то первый элемент является самым наименьшим.

1.8 Задание 8

В данном задании была создана переменная, которая хранит в себе сумму покупки, которую ввел пользователь. После был создан список, который содержит все виды скидок. И затем была создана переменная, которая содержит индекс предоставляемой скидкой. После выводиться предоставляемая скидка и сумму к оплате с учетом скидки. На рисунке 8 представлен код программы.

```

1  smp = float(input("Введите сумму покупки:"))
2  sk = [0, 5, 10, 15]
3  pr = [smp < 1000, 1000 <= smp <= 5000, 5000 <= smp <= 10000, smp > 10000].index(True)
4  print(f"Ваша скидка: {sk[pr]}")
5  print(f"К оплате: {smp - (smp / 100 * sk[pr])}")

```

Рисунок 8 — Листинг программы задания 8

Пояснение работы программы:

1) 1 строка кода запрашивает у пользователя сумму покупки и после сохраняет это числовое значение с плавающей точкой в переменную `smp`.

2) 2 строка кода создает список, который хранит в себе целочисленные размеры скидок. И сохраняет этот список в переменную `sk`.

3) 3 строка кода создает и сохраняет индекс предоставляемой скидки с помощью списка, который содержит все условия скидок и при этом индекс каждого условия совпадает с индексом предоставляемой скидки из списка `sk`. После с помощью функции `index(True)`, которая возвращает индекс `True`, которая находится в списке условий. И уже этот индекс сохраняется в переменную `pr`. Пояснение: данный список будет состоять из 1 `True` и 3 `False` так, как каждое значение в списке это условие связанная с переменной `smp`. И условия поставлены так, как и в условии задачи.

4) 4 строка кода выводит на экран предоставляемую скидку с помощью команды `print()`.

5) 5 строка кода выводит сумму оплаты с учетом скидки, так же с помощью команды `print()`.

1.9 Задание 9

В данном задании была создана переменная, которая хранит в себе время в часах, которую ввел пользователь. После был создан список, который хранит в себе времена суток. И затем была создана переменная, которая хранит в себе нужный индекс времени года. На рисунки 9 предоставлен код программы.

```

1  tm = float(input("Введи час: "))
2  vr = ["ночь", "утро", "день", "вечер"]
3  p = [0 <= tm <= 5, 6 <= tm <= 11, 12 <= tm <= 17, 18 <= tm <= 23].index(True)
4  print(f"Сейчас {vr[p]}")

```

Рисунок 9 — Листинг программы задания 9

Пояснение работы программы:

1) 1 строка кода запрашивает у пользователя время в часах и после сохраняет это числовое значение с плавающей точкой в переменную `tm`.

2) 2 строка кода создает список, который хранит в себе времена суток и сохраняет этот список в переменную `vr`.

3) 3 строка кода создает и сохраняет индекс текущего времени суток с помощью списка, где каждый элемент это условия, индексы которых совпадают с соответствующим индексом времени в списке `vr`. После с помощью функции `index(True)`, которая возвращает индекс `True`, которая

находиться в списке условий. И уже этот индекс сохраняется в переменную `p`. Пояснение: данный список будет состоять из 1 True и 3 False так, как каждое значение в списке это условие связанная с переменной `tm`. И условия поставлены так, как и в условии задачи.

4) 4 строка кода выводит на экран ответ, а именно время суток из списка `vr` по индексу `p` с помощью команды `print()`.

1.10 Задание 10

В данном задании была создана функция для вычисления является ли число составным или простым. После был создан бесконечный цикл, чтобы обработать и проверить запрос пользователя на корректность ввода. В случае не корректного ввода пользователю говорится, что не так и просят ввести ещё раз. В обратном случае пользователь получает желанный ответ. На рисунке 10 предоставлен код программы.

```

1  def isps(ch):
2      p = ["составное число", "простое число"]
3      f = 0
4      for i in range(1, int(ch)+ 1):
5          if int(ch) % i == 0:
6              f += 1
7      return p[(f == 2)]
8
9  while True:
10     ch = input("Введите неотрицательное число: ")
11     if ch.isdigit():
12         print(isps(ch))
13         break
14     elif ch[0] == "-" and ch[1:].isdigit():
15         print("Введите неотрицательное число (':")
16     else:
17         print("Введите число -_-")
18

```

Рисунок 10 — Листинг программы задания 10

Пояснение работы программы:

- 1) 1 строка кода это создание функции `isps` с переменной `ch`, с помощью функции `def()`.
- 2) 2 строка кода создает внутри функции список, который содержит 2 строки (“составное число”, “простое число”) и сохраняет этот список в локальную переменную `p`.
- 3) 3 строка кода создает переменную `f` и присваивает ей числовое значение 0.
- 4) 4-6 строка кода считает сколько есть делителей у числа. Создается цикл `for` с переменной `i` от 1 до числа (локальной переменной `ch`), с которой вызывается функция. После если число делиться на `i` то я прибавляю 1 в переменную `f`, чтобы посчитать количество делителей.

5)7 строка кода возвращает элемент из списка `r` по индексу, который определяется истинностью условия. Где условия звучит так: равняется ли количество делителей 2. Ведь если у числа 2 делителя, то это означает, что число простое. Иначе составное.

6)9 строка кода создает бесконечный цикл с помощью цикла `while` и с помощью значения этого цикла `True`.

7)10 строка кода запрашивает у пользователя не отрицательное число и после в строковом значении сохраняет в переменную `ch`.

8)11-13 строка кода проверяет является ли ввод пользователя неотрицательным числом с помощью функции `isdigit()`. После если это истина выводится результат программы с помощью команды `print()` вызывая от этого числа вышеописанную функцию `isps(ch)`. И затем идет прерывание бесконечного цикла с помощью команды `break`. Это происходит так, как пользователь получил результат программы.

9)14-15 строка кода проверяет является ли введенное число отрицательным(это проверяется в случае если выше перечисленная проверка для числа пользователя не прошла) и в случаи положительного результата выводится сообщение, что нужно ввести неотрицательное число. С помощью команды `print()`

10)16-17 строка кода срабатывает если все выше перечисленные условия не выполнились. То тогда выводится сообщение, что нужно ввести число. И вывод происходит с помощью очередной команды `print()`.