Министерство образования Республики Беларусь

Белорусский Национальный Технический Университет

Факультет информационных технологий и робототехники

Кафедра «Программное обеспечение информационных систем

и технологий»

**Отчёт**

по лабораторной работе №02

по дисциплине ***«Языки программирования»***

тема: **«Переменные, простые типы данных и операции над ними»**

|  |  |  |
| --- | --- | --- |
| Исполнитель: |  | студент группы 10702119  Старостин Павел Алексеевич |
| Преподаватель: |  | Белова Светлана Владимировна |

2019-2020 учебный год

**ЛАБОРАТОРНАЯ РАБОТА №02**

**«Переменные, простые типы данных и операции над ними»**

Цель работы:

Цель работы: освоить базовый синтаксис языка Python, простые типы данных, приобрести навыки создания интерактивных программ с использованием линейных алгоритмов.

**Задание на лабораторную работу:**

**1**. Разработать интерактивную программу, демонстрирующую работу с простыми типами данных:

1.1 Программа должна запрашивать имя пользователя (строка) и почтовый индекс (целое число) и записывать соответственно в переменные Chameleon1 и Chameleon2. Вывести значения и типы переменных на экран.

1.2 Преобразовать переменную Chameleon2 к строковому типу с использованием функции str(). Умножить полученную строку на число даты рождения пользователя и вывести результат.

1.3 Сложить переменные Chameleon1 и Chameleon2, результат записать в Chameleon3. Вывести результат суммирования и тип переменной.

1.4 Преобразовать переменную Chameleon3 к списку с использованием функции list(). Вывести результат и тип переменной на экран.

1.5 Напечатать начальный и конечный элементы списка Chameleon3. Первое значение списка Chameleon3 заменить возрастом пользователя.

1.6 Прибавить к списку Chameleon3 список, вводимый пользователем с клавиатуры. Результат вывести на экран, а затем умножить на 2 и снова вывести на экран.

1.7 Преобразовать переменную Chameleon3 к кортежу с использованием функции tuple(). Вывести результат и тип переменной на экран.

1.8 Попытаться изменить любое значение кортежа Chameleon3.

1.9 Продемонстрировать операцию сложения кортежей и умножения кортежа на число.

1.10 Преобразовать переменную Chameleon3 к множеству с использованием функции set(). Вывести результат и тип переменной на экран. Сделать вывод о назначении множеств.

1.11 Создать словарь My\_Dictionary из пяти пар, осуществляющий перевод слов с русского на английский. Продемонстрировать обращение к элементам словаря по ключу. Вывести словарь несколько раз на экран. Сделать вывод.

1.12 Сделать вывод о простых типах данных языка Python, универсальности команд и о динамической типизации.

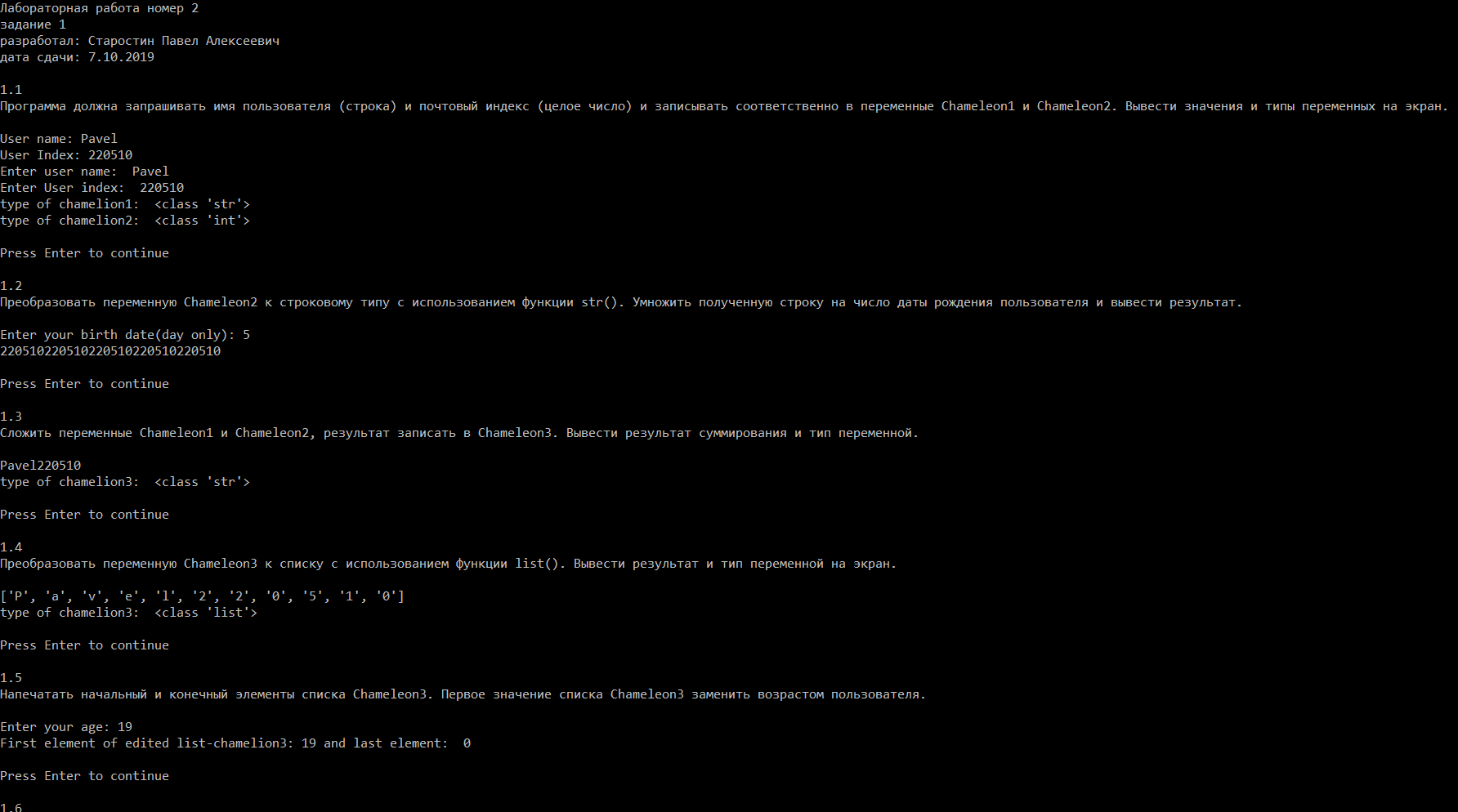
**2.** Разработать интерактивную программу «What is My Age in Seconds» («Каков мой возраст в секундах»), которая на входе принимает дату рождения пользователя, рассчитывает количество прожитых пользователем секунд и выводит результат на экран монитора.

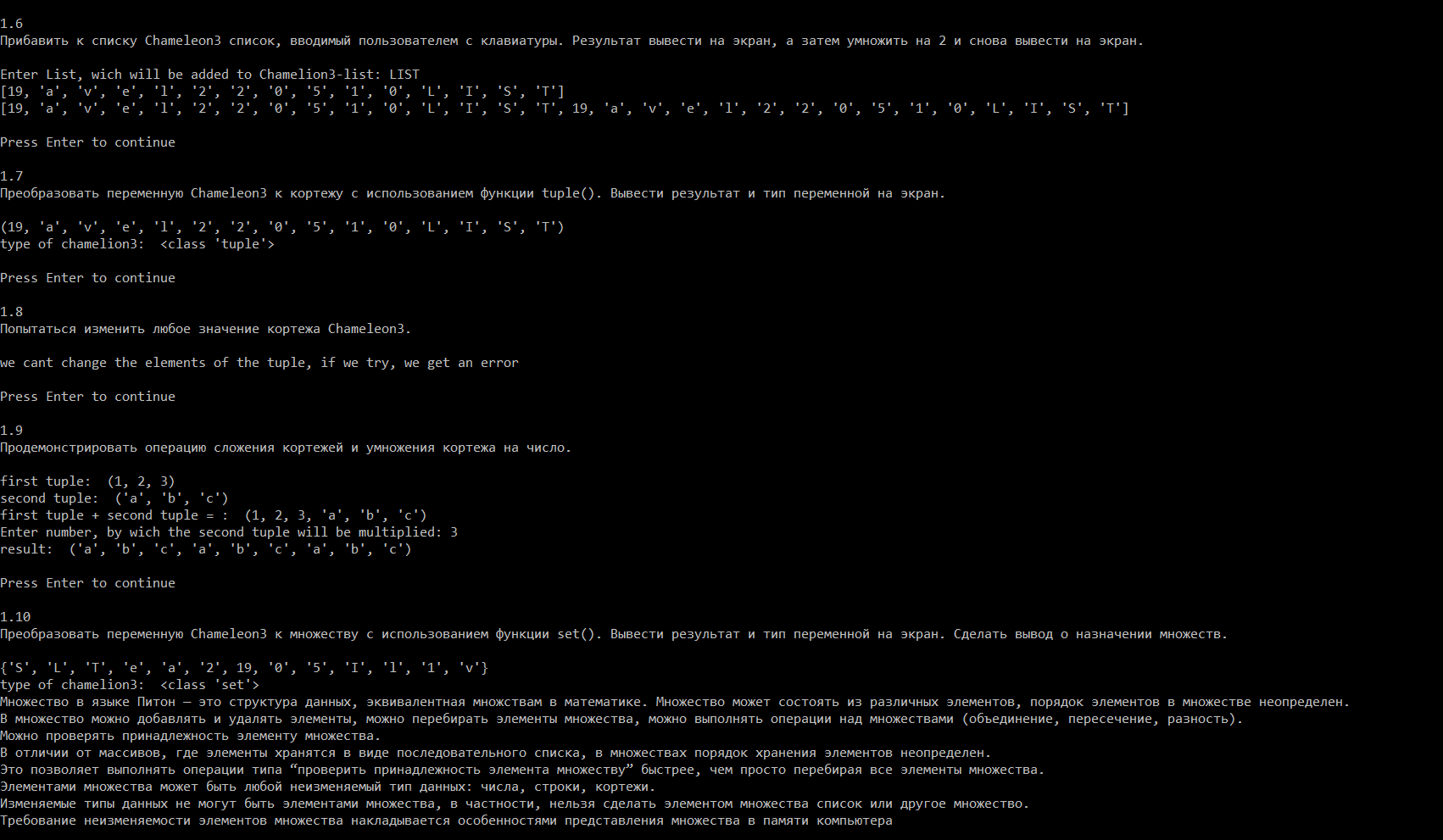
**Дополнительное задание\*** Согласно официальному сайту Европейского Центрального Банка в настоящий момент наши западные соседи используют в обращении 7 номиналов банкнот евро – 5, 10, 20, 50, 100, 200 и 500 евро. Монетный ряд включает 8 монет достоинством 1, 2, 5, 10, 20, 50 евроцентов, 1 и 2 евро (1 евро = 100 евроцентов). Вариант 1. Необходимо написать программу для специального банкомата (ATM). Банкомат должен оптимальным способом выдавать любую ввдённую пользователем сумму: вначале выдаются крупные банкноты, потом меньше и так вплоть до одного евроцента. К примеру, пользователь запрашивает в банкомате сумму в 587 евро и 99 центов. Банкомат ему должен выдать следующее: банкноты номиналом 500, 50, 20, 10 и 5 евро, затем монеты: 50, 20 × 2, 5, 2 × 2 и 1 евроцент.

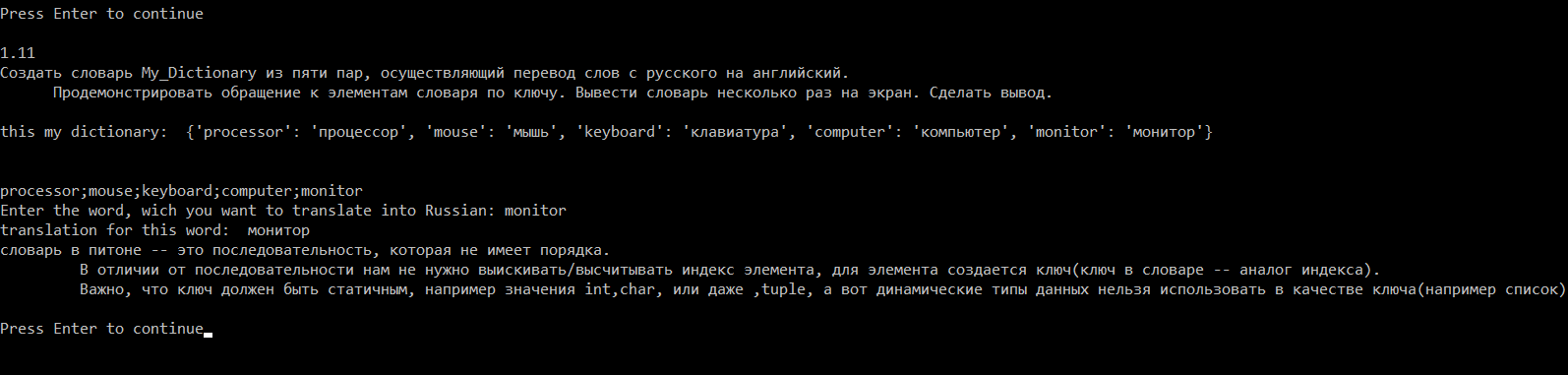
**Вариант 1**

Результаты выполнения основных заданий:

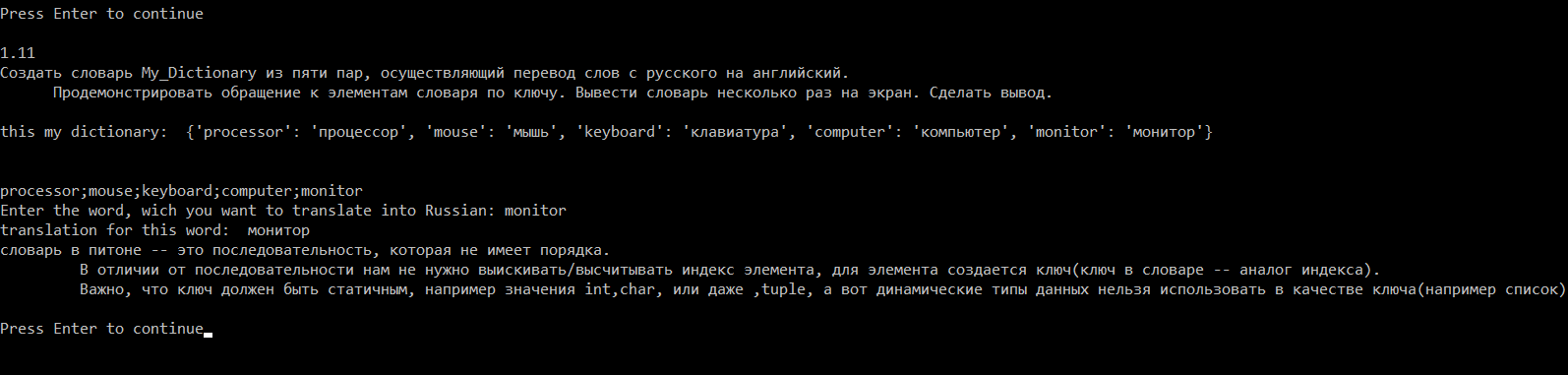
1)



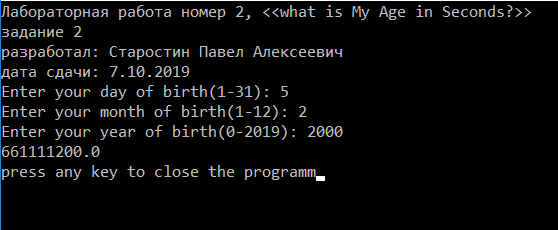




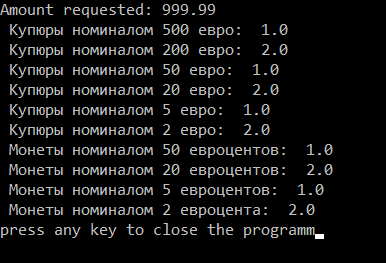
2)



2)



Дополнительное задание)



Ответы на контрольные вопросы:

1. Описать архитектуру и основные элементы компьютера. Какой элемент является центральным при построении любой вычислительной систем:

*-Архитектурой* компьютера называется его описание на некотором общем уровне, включающее описание пользовательских возможностей программирования, системы команд, системы адресации, организации памяти и т.д. Архитектура определяет принципы действия, информационные связи и взаимное соединение основных логических узлов компьютера: процессора, оперативного ЗУ, внешних ЗУ и периферийных устройств. Общность архитектуры разных компьютеров обеспечивает их совместимость с точки зрения пользователя.

*Структура* компьютера — это совокупность его функциональных элементов и связей между ними. Элементами могут быть самые различные устройства — от основных логических узлов компьютера до простейших схем. Структура компьютера графически представляется в виде структурных схем, с помощью которых можно дать описание компьютера на любом уровне детализации.

Центральный процессор (ЦП) - это центральный блок персонального компьютера, предназначенный для управления работой всех остальных блоков и выполнения арифметических и логических операций над информацией.

Основные элементы компьютера:

-корпус,

-материнская плата,

-процессор,

-кулер,

-оперативка,

-жесткий диск,

-видеокарта,

-блок питания

2. Какие типы памяти доступны при разработке программ?

Виды памяти:

Существует 3 типа памяти: статический, автоматический и динамический.

**Статический** — выделение памяти до начала исполнения программы. Такая память доступна на протяжении всего времени выполнения программы. Во многих языках для размещения объекта в статической памяти достаточно задекларировать его в глобальной области видимости.

**Автоматический**, также известный как «размещение на стеке», — самый основной, автоматически выделяет аргументы и локальные переменные функции, а также прочую метаинформацию при вызове функции и освобождает память при выходе из неё.

Динамическая — выделение памяти из ОС по требованию приложения.

Автоматическая и статическая память выделяются единоразово перед запуском программы. При их нехватке, либо если модель LIFO не совсем подходит, используется динамическая память.

Приложение при необходимости может запросить у ОС дополнительную память через аллокатор или напрямую через системный вызов.

1. Что такое машинный код?

**Машинный код** или **машинный язык** представляет собой набор инструкций, выполняемых непосредственно центральным процессором компьютера ([CPU](https://ru.bmstu.wiki/CPU_(Central_Processing_Unit))). Каждая команда выполняет очень конкретную задачу, например, загрузки (load), перехода (jump) или элементарной арифметической или логической операции для единицы данных в регистре процессора или памяти. Каждая программа выполняется непосредственно процессором и состоит из ряда таких инструкций.

Машинный код можно рассматривать как самое низкоуровневое представление скомпилированной или собранной компьютерной программы или в качестве примитивного и аппаратно-зависимого [языка программирования](https://ru.bmstu.wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). Писать программы непосредственно в машинном коде возможно, однако это утомительно и подвержено ошибкам, так как необходимо управлять отдельными битами и вычислять числовые адреса и константы вручную. По этой причине машинный код практически не используется для написания программ.

1. В чём отличия языка высокого уровня от языка низкого уровня?

Высокоуровневый язык программирования — язык программирования, разработанный для быстроты и удобства использования программистом. Основная черта высокоуровневых языков — это абстракция, то есть введение смысловых конструкций, кратко описывающих такие структуры данных и операции над ними, описания которых на машинном коде (или другом низкоуровневом языке программирования) очень длинны и сложны для понимания.  
  
Низкоуровневый язык программирования — язык программирования, близкий к программированию непосредственно в машинных кодах.  
Как правило, использует особенности конкретного семейства процессоров.

1. Какой язык понимает и обрабатывает центральный процессор (Central Process Unit, CPU)?

Машинный язык преобразованный в электрические сигналы.

1. Что такое транслятор и что он делает?

**Трансля́тор** — программа или техническое средство, которая принимает на вход программу на одном языке (он в этом случае называется исходный язык, а программа — исходный код) , и преобразует её в программу, написанную на другом языке (соответственно, целевой язык и объектный код) .  
  
В качестве целевого языка наиболее часто выступают машинный код, Ассемблер и байт-код, так как они наиболее удобны (с точки зрения производительности) для последующего исполнения.  
  
7) **Что общего между компилятором и интерпретатором и чем они отличаются?**

Наиболее часто встречаются две разновидности трансляторов:  
  
Компиляторы — выдают результат в виде исполняемого файла (в данном случае считаем, что компоновка входит в компиляцию) . Этот файл:   
транслируется один раз — может быть запущен самостоятельно   
не требует для работы наличия на машине создавшего его транслятора   
Интерпретаторы — исполняют программу после разбора (в этом случае в роли объектного кода выступает внутреннее представление программы интерпретатором) . Исполняется она построчно. В данном случае программа   
транслируется (интерпретируется) при каждом запуске (если объектный код кешируется, возможны варианты)   
требует для исполнения наличия на машине интерпретатора и исходного кода.

Ниже перечислены главные отличия между компилятором и интерпретатором:

* Интерпретатор берет одну инструкцию, транслирует и выполняет ее, а затем берет следующую инструкцию. Компилятор же транслирует всю программу сразу, а потом выполняет ее.
* Компилятор генерирует отчет об ошибках после трансляции всего, в то время как интерпретатор прекратит трансляцию после первой найденной ошибки.
* Компилятор по сравнению с интерпретатором требует больше времени для анализа и обработки языка высокого уровня.
* Помимо времени на обработку и анализ, общее время выполнения кода компилятора быстрее в сравнении с интерпретатором.
  1. Что такое переменная и зачем она нужна в программе?

Так же, как в алгебре. Для хранения данных и промежуточных результатов вычислений, которые могут изменяться.

9) Что такое «соглашение по присваиванию имён»?

Это когда какой-то переменной присваивается какое-то значение

1. Что такое константа?

Это постоянное значение переменной

1. Что такое hardcode («тяжёлый код»)?

Тяжелочитаемый код

1. Описать стандартные простые типы данных в Python?

[*Числа: целые, вещественные, комплексные*](https://pythonworld.ru/tipy-dannyx-v-python/chisla-int-float-complex.html)

[*None (null)*](https://pythonworld.ru/tipy-dannyx-v-python/none.html)

[*Множества (set*](https://pythonworld.ru/tipy-dannyx-v-python/mnozhestva-set-i-frozenset.html)

[*Функции и их аргументы*](https://pythonworld.ru/tipy-dannyx-v-python/vse-o-funkciyax-i-ix-argumentax.html)

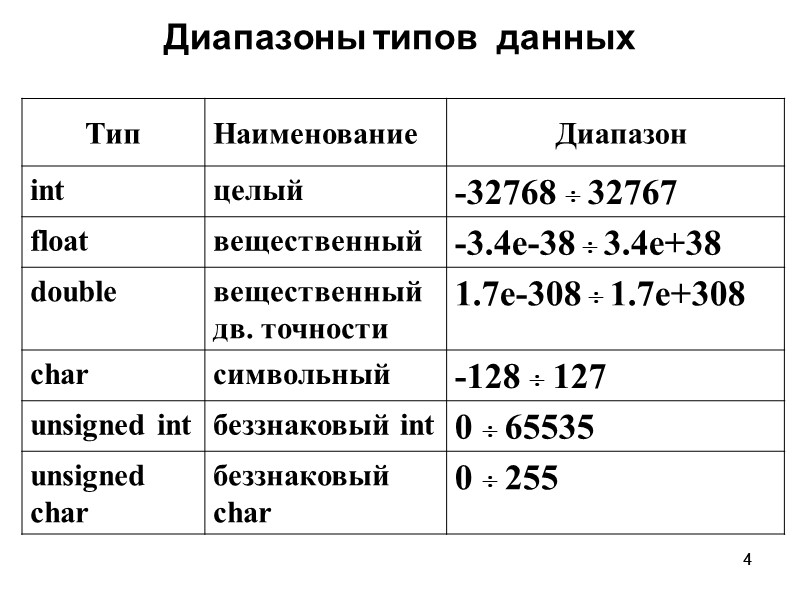
[*Кортежи (tuple)*](https://pythonworld.ru/tipy-dannyx-v-python/kortezhi-tuple.html)

[*Словари (dict)*](https://pythonworld.ru/tipy-dannyx-v-python/slovari-dict-funkcii-i-metody-slovarej.html)

[*Списки (list)*](https://pythonworld.ru/tipy-dannyx-v-python/spiski-list-funkcii-i-metody-spiskov.html)

[*Строки.*](https://pythonworld.ru/tipy-dannyx-v-python/stroki-funkcii-i-metody-strok.html)

1. Что измеряет диапазон типа данных?



1. Что такое выражение?

**Выражение** — это любое сочетание чисел, букв и знаков операций.

1. Что такое инициализация?

Инициализация — создание, активация, подготовка к работе, определение параметров. Приведение программы или устройства в состояние готовности к использованию. Термин употребляется как для программных, так и для аппаратных средств.

1. Можно ли обратиться к переменной, если инициализировать её чуть позже?

Нет, переменная изначально должна быть инициализирована, а потом используема

1. Что произойдёт, если присвоить строковой переменой целочисленное значение (или наоборот)?

Переменная поменяет свой тип при корректности трансформации, иначе выдаст ошибку

1. Какой из арифметических операторов не может иметь дробного

%

1. Какой арифметический оператор не может иметь в качестве второго операнда 0?

%; /; //

1. Описать все группы операций, доступные в Рython

В данном языке программирования доступны все математические операции

Сложение «+»

Вычитание «-»

Умножение «\*»

Деление «/»

Возведение в степень «\*\*»

При подключении библиотек доступно логарифмирование и прочее.

Что я освоил в процессе выполнения лабораторной работы (выводы):

1) Базовые основы разработки программ на языке программирования Python.

2) Процесс создания простейшей программы на Python c использованием среды разработки IDLE.

3) Два режима (интерактивный и сценарный) разработки приложений.

4) Некоторые горячие клавиши среды IDLE для быстрой разработки приложений

5) Изучили raw-строки и escape-последовательности

**6)** Узнали про *б*олее мощное средство разработки программ на *Python - jetbrains pycharm Community Edition* .

Листинг исходных кодов программ:

1)

print("Лабораторная работа номер 2")

print("задание 1")

print("разработал: Старостин Павел Алексеевич")

print("дата сдачи: 7.10.2019")

print("\n1.1\nПрограмма должна запрашивать имя пользователя (строка) и почтовый индекс (целое число) и записывать соответственно в переменные Chameleon1 и Chameleon2. Вывести значения и типы переменных на экран. \n")

chamelion1 = input("User name: ")

chamelion2 = int(input("User Index: "))

print("Enter user name: ",chamelion1)

print("Enter User index: ",chamelion2)

print("type of chamelion1: ",type(chamelion1))

print("type of chamelion2: ",type(chamelion2))

input("\nPress Enter to continue")

print("\n1.2\nПреобразовать переменную Chameleon2 к строковому типу с использованием функции str(). Умножить полученную строку на число даты рождения пользователя и вывести результат.\n")

chamelion2 = str(chamelion2)

date = int(input("Enter your birth date(day only): "))

print(chamelion2\*date)

input("\nPress Enter to continue")

print("\n1.3\nСложить переменные Chameleon1 и Chameleon2, результат записать в Chameleon3. Вывести результат суммирования и тип переменной. \n")

chamelion3 = (chamelion1+chamelion2)

print(chamelion3)

print("type of chamelion3: ",type(chamelion3))

input("\nPress Enter to continue")

print("\n1.4\nПреобразовать переменную Chameleon3 к списку с использованием функции list(). Вывести результат и тип переменной на экран. \n")

chamelion3 = list(chamelion3)

print(chamelion3)

print("type of chamelion3: ",type(chamelion3))

input("\nPress Enter to continue")

print("\n1.5\nНапечатать начальный и конечный элементы списка Chameleon3. Первое значение списка Chameleon3 заменить возрастом пользователя.\n")

age = int(input("Enter your age: "))

chamelion3[0] = age

print("First element of edited list-chamelion3:",chamelion3[0],"and last element: ",chamelion3[-1])

'''

менее быстрый способ нахождения последнего элемента списка:

int(lastElement)=len(chamelion3)

print("First and last element of list-chamelion3:",chamelion3[0],chamelion3[lastElement-1])

'''

input("\nPress Enter to continue")

print("\n1.6\nПрибавить к списку Chameleon3 список, вводимый пользователем с клавиатуры. Результат вывести на экран, а затем умножить на 2 и снова вывести на экран.\n")

listAdd = input("Enter List, wich will be added to Chamelion3-list: ")

listAdd=list(listAdd)

chamelion3 = (chamelion3+listAdd)

print(chamelion3)

print(chamelion3\*2)

input("\nPress Enter to continue")

print("\n1.7\nПреобразовать переменную Chameleon3 к кортежу с использованием функции tuple(). Вывести результат и тип переменной на экран.\n")

chamelion3 = tuple(chamelion3)

print(chamelion3)

print("type of chamelion3: ",type(chamelion3))

input("\nPress Enter to continue")

print("\n1.8\nПопытаться изменить любое значение кортежа Chameleon3.\n")

print("we cant change the elements of the tuple, if we try, we get an error")

input("\nPress Enter to continue")

print("\n1.9\nПродемонстрировать операцию сложения кортежей и умножения кортежа на число.\n")

firstTuple = (1,2,3);

secondTuple = ('a','b','c');

print("first tuple: ", firstTuple)

print("second tuple: ", secondTuple)

print("first tuple + second tuple = : ", firstTuple + secondTuple)

x = int(input("Enter number, by wich the second tuple will be multiplied: "))

print("result: ",secondTuple\*x)

input("\nPress Enter to continue")

print("\n1.10\nПреобразовать переменную Chameleon3 к множеству с использованием функции set(). Вывести результат и тип переменной на экран. Сделать вывод о назначении множеств.\n")

chamelion3 = set(chamelion3)

print(chamelion3)

print("type of chamelion3: ",type(chamelion3))

print('''Множество в языке Питон — это структура данных, эквивалентная множствам в математике. Множество может состоять из различных элементов, порядок элементов в множестве неопределен.

В множество можно добавлять и удалять элементы, можно перебирать элементы множества, можно выполнять операции над множествами (объединение, пересечение, разность).

Можно проверять принадлежность элементу множества.

В отличии от массивов, где элементы хранятся в виде последовательного списка, в множествах порядок хранения элементов неопределен.

Это позволяет выполнять операции типа “проверить принадлежность элемента множеству” быстрее, чем просто перебирая все элементы множества.

Элементами множества может быть любой неизменяемый тип данных: числа, строки, кортежи.

Изменяемые типы данных не могут быть элементами множества, в частности, нельзя сделать элементом множества список или другое множество.

Требование неизменяемости элементов множества накладывается особенностями представления множества в памяти компьютерa''')

input("\nPress Enter to continue")

print('''\n1.11\nСоздать словарь My\_Dictionary из пяти пар, осуществляющий перевод слов с русского на английский.

Продемонстрировать обращение к элементам словаря по ключу. Вывести словарь несколько раз на экран. Сделать вывод.\n''')

My\_Dictionary = {'processor':'процессор', 'mouse':'мышь', 'keyboard':'клавиатура', 'computer':'компьютер', 'monitor':'монитор'}

print("this my dictionary: ",My\_Dictionary)

print('\n')

print("processor;mouse;keyboard;computer;monitor")

TranslatedWord = input("Enter the word, wich you want to translate into Russian: ")

print("translation for this word: ",My\_Dictionary[TranslatedWord])

print('''словарь в питоне -- это последовательность, которая не имеет порядка.

В отличии от последовательности нам не нужно выискивать/высчитывать индекс элемента, для элемента создается ключ(ключ в словаре -- аналог индекса).

Важно, что ключ должен быть статичным, например значения int,char, или даже ,tuple, а вот динамические типы данных нельзя использовать в качестве ключа(например список) ''')

input("\nPress Enter to continue")

print("\n1.12\nСделать вывод о простых типах данных языка Python, универсальности команд и о динамической типизации.\n")

2)

print("Лабораторная работа номер 2, <<what is My Age in Seconds?>>");

print("задание 2");

print("разработал: Старостин Павел Алексеевич");

print("дата сдачи: 7.10.2019");

secInDay = 60\*60\*24;

secInMonth = secInDay\*31

secInYear = secInMonth\*12-9\*secInDay

day = 7;

month = 10;

year = 2019;

Bday = int(input("Enter your day of birth(1-31): "));

Bmonth = int(input("Enter your month of birth(1-12): "));

Byear = int(input("Enter your year of birth(0-2019): "));

ageInSeconds = secInDay\*day+secInMonth\*month+secInYear\*year-secInDay\*Bday-secInMonth\*Bmonth-secInYear\*Byear+year/4\*secInDay;

print (ageInSeconds);

input("press any key to close the programm");

3)

balance = float(input("Amount requested: "))

numOf500 = (balance) // 500

if (numOf500>0):

print (" Купюры номиналом 500 евро: ", numOf500);

balance = balance - 500 \* numOf500;

numOf200 = (balance) // 200

if (numOf200>0):

print (" Купюры номиналом 200 евро: ", numOf200);

balance = balance - 200 \* numOf200;

numOf100 = (balance) // 100

if (numOf100>0):

print (" Купюры номиналом 100 евро: ", numOf100);

balance = balance - 100 \* numOf100;

numOf50 = (balance) // 50

if (numOf50>0):

print (" Купюры номиналом 50 евро: ", numOf50);

balance = balance - 50 \* numOf50;

numOf20 = (balance) // 20

if (numOf20>0):

print (" Купюры номиналом 20 евро: ", numOf20);

balance = balance - 20 \* numOf20;

numOf10 = (balance) // 10

if (numOf10>0):

print (" Купюры номиналом 10 евро: ", numOf10);

balance = balance - 10 \* numOf10;

numOf5 = (balance) // 5

if (numOf5>0):

print (" Купюры номиналом 5 евро: ", numOf5);

balance = balance - 5 \* numOf5;

numOf2 = (balance) // 2

if (numOf2>0):

print (" Купюры номиналом 2 евро: ", numOf2);

balance = balance - 2 \* numOf2;

numOf1 = (balance) // 1

if (numOf1>0):

print (" Купюры номиналом 1 евро: ", numOf1);

balance = balance - numOf1;

numOf50cent = (balance) // 0.5

if (numOf50cent>0):

print (" Монеты номиналом 50 евроцентов: ", numOf50cent);

balance = balance - 0.5 \* numOf50cent;

numOf20cent = (balance) // 0.2

if (numOf20cent>0):

print (" Монеты номиналом 20 евроцентов: ", numOf20cent);

balance = balance - 0.2 \* numOf20cent;

numOf10cent = (balance) // 0.1

if (numOf10cent>0):

print (" Монеты номиналом 10 евроцентов: ", numOf10cent);

balance = balance - 0.1 \* numOf10cent;

numOf5cent = (balance) // 0.05

if (numOf5cent>0):

print (" Монеты номиналом 5 евроцентов: ", numOf5cent);

balance = balance - 0.05 \* numOf5cent;

numOf2cent = (balance) // 0.02

if (numOf2cent>0):

print (" Монеты номиналом 2 евроцентa: ", numOf2cent);

balance = balance - 0.02 \* numOf2cent;

numOf1cent = (balance) // 0.01

if (numOf1cent>0):

print (" Монеты номиналом 1 евроцент: ", numOf1cent);

input("press any key to close the programm")