**Модуль 1**

* 1. **Динамическая типизация**

Цель: Написать программу на языке Python, используя Pycharm, для демонстрации динамической типизации.

В проекте, где вы решаете домашние задания, создайте модуль 'module\_1\_3.py' и напишите весь код в нём.

Создайте переменные разных типов данных:

Создайте переменную name и присвойте ей значение вашего имени (строка).

Выведите значение переменной name на экран.

Создайте переменную age и присвойте ей значение вашего возраста (целое число).

Выведите значение переменной age на экран.

Перезапишите в age текущее значение переменной age + новое.

Как неверно (просто перезапись на новое число):

a = 15

a = 17

Выведите измененное значение переменной age на экран.

Создайте переменную is\_student и присвойте ей значение True (логическое значение).

Выведите значение переменной is\_student на экран.

Примечания:

Для вывода значений на экран используйте функцию print().

Обратите внимание на использование разных типов данных и возможности их изменения.

Пример результата выполнения программы:

Name: John

Age: 25

New Age: 26

Is Student: True

**1.2 Переменные**

Практическое задание по работе в Pycharm - "Переменные".

Цель: научиться правильно называть переменные и взаимодействовать с ними.

Задача:

Напишите 4 переменных которые буду обозначать следующие данные:

Количество выполненных ДЗ (запишите значение 12)

Количество затраченных часов (запишите значение 1.5)

Название курса (запишите значение 'Python')

Время на одно задание (вычислить используя *1* и *2* переменные)

Выведите на экран(в консоль), используя переменные, следующую строку:

*Курс: Python, всего задач:12, затрачено часов: 1.5, среднее время выполнения 0.125 часа.*

Пример написанного кода:

first\_name = 'Vasya'

apple\_count = 10

second\_name = 'Petya'

orange\_count = 15

print(first\_name, 'дал', second\_name, apple\_count, 'яблок и', orange\_count, 'апельсинов')

Результат выполнения кода:Vasya дал Petya 10 яблок и 15 апельсинов

Примечания:

Для вывода значений на экран используйте функцию print(a, b, c), где a, b, c - данные которые выводятся на экран.

Для названия переменных используйте только: латинские буквы, не начинайте запись с числа, пишите в змеином регистре(разделяя слова символом '\_')

Пробелы между словами и символами ':' ',' можно ставить на своё усмотрение.

**1.3 Строки и индексация строк**

Практическое задание по уроку "Строки и индексация строк"

Цель: Научиться работать со строками и индексацией строк в Python.

Задача:

Выполните следующие действия в PyCharm:

В переменную ***example*** запишите любую строку.

Выведите на экран(в консоль) первый символ этой строки.

Выведите на экран(в консоль) последний символ этой строки (используя отрицательный индекс).

Выведите на экран(в консоль) вторую половину этой строки (С нечётным количеством символов: 'Urban' -> 'ban').

Выведите на экран(в консоль) это слово наоборот.

Выведите на экран(в консоль) каждый второй символ этой строки. (Пример: 'Топинамбур'->'оиабр')

Вводные данные:

example = 'Топинамбур'

Вывод на экран(в консоль):

Т

р

амбур

рубманипоТ

оиабр

Примечания:

Для вывода значений на экран используйте функцию **print()**

Для доступа к символу строки по индексу используйте квадратные скобки и индекс символа ('Urban'[3] -  четвёртый символ строки 'Urban' - 'a')

Индексация начинается с нуля.

**1.4 Организация программ и методы строк**

Практическая работа по уроку "Организация программ и методы строк"

Цель: Написать программу на языке Python с использованием Pycharm для работы с методами строк и организации программ.

1. В проекте, где вы решаете домашние задания, создайте модуль '**module\_1\_4.py**' и напишите весь код в нём.

2. Организуйте программу:

Создайте переменную my\_string и присвойте ей значение строки с произвольным текстом (функция input()).

Вывести количество символов введённого текста

3. Работа с методами строк:

Используя методы строк, выполните следующие действия:

Выведите строку my\_string в верхнем регистре.

Выведите строку my\_string в нижнем регистре.

Измените строку my\_string, удалив все пробелы.

Выведите первый символ строки my\_string.

Выведите последний символ строки my\_string.

Примечания:

Для вывода значений на экран используйте функцию print().

Для вызова методов строк используйте операцию точки '.'.

Дополнительно о всех методах строк можно узнать [здесь](https://docs.python.org/3/library/stdtypes.html#string-methods).

Пример результата выполнения программы:

Код:

name = input()

print(name.upper())

Ввод в консоль:

Urban

Вывод на консоль:

URBAN

**1.5 "Неизменяемые и изменяемые объекты. Кортежи и списки"**

Цель: Написать программу на языке Python, используя Pycharm, для работы с неизменяемыми и изменяемыми объектами.

1. В проекте, где вы решаете домашние задания, создайте модуль '**module\_1\_5.py**' и напишите весь код в нём.

2. Задайте переменные разных типов данных:

  - Создайте переменную immutable\_var и присвойте ей кортеж из нескольких элементов разных типов данных.

  - Выполните операции вывода кортежа immutable\_var на экран.

3. Изменение значений переменных:

  - Попытайтесь изменить элементы кортежа immutable\_var. Объясните, почему нельзя изменить значения элементов кортежа.

4. Создание изменяемых структур данных:

  - Создайте переменную mutable\_list и присвойте ей список из нескольких элементов.

  - Измените элементы списка mutable\_list.

  - Выведите на экран измененный список mutable\_list.

Примечания:

- Для вывода значений на экран используйте функцию print().

- Обратите внимание на особенности изменяемых и неизменяемых объектов в Python.

Пример результата выполнения программы:

Immutable tuple: (1, 2, 'a', 'b')

Mutable list: [1, 2, 'a', 'b', 'Modified']

**1.6 Словари и множества**

Практическое задание по теме: "Словари и множества"

Цель: Написать программу на языке Python, используя Pycharm, для работы со словарями и множествами.

1. В проекте, где вы решаете домашние задания, создайте модуль '**module\_1\_6.py**' и напишите весь код в нём.

2. Работа со словарями:

  - Создайте переменную my\_dict и присвойте ей словарь из нескольких пар ключ-значение.

Например: Имя(str)-Год рождения(int).

  - Выведите на экран словарь my\_dict.

  - Выведите на экран одно значение по существующему ключу, одно по отсутствующему из словаря my\_dict без ошибки.

  - Добавьте ещё две произвольные пары того же формата в словарь my\_dict.

 - Удалите одну из пар в словаре по существующему ключу из словаря my\_dict и выведите значение из этой пары на экран.

  - Выведите на экран словарь my\_dict.

3. Работа с множествами:

  - Создайте переменную my\_set и присвойте ей множество, состоящее из разных типов данных с повторяющимися значениями.

  - Выведите на экран множество my\_set (должны отобразиться только уникальные значения).

  - Добавьте 2 произвольных элемента в множество my\_set, которых ещё нет.

  - Удалите один любой элемент из множества my\_set.

  - Выведите на экран измененное множество my\_set.

Примечания:

- Для вывода значений на экран используйте функцию print().

- Помните, что словарь и множество - неупорядоченные коллекции.

- Больше о методах словарей [тут](https://docs.python.org/3/library/stdtypes.html#dict), множеств [тут](https://docs.python.org/3/library/stdtypes.html#set).

Пример результата выполнения программы:

Dict: {'Vasya': 1975, 'Egor': 1999, 'Masha': 2002}

Existing value: 2002

Not existing value: None

Deleted value: 1999

Modified dictionary: {'Vasya': 1975, 'Kamila': 1981, 'Artem': 1915, 'Masha': 2002}

Set: {1, 'Яблоко', 42.314}

Modified set: {'Яблоко', 42.314, 13, (5, 6, 1.6)}

**1.7 Дополнительное практическое задание по модулю: "Базовые структуры данных."**

Цель: Применить знания полученные в модуле, решив задачу повышенного уровня сложности

Предисловие:

Сложность подобных задач заключается в:

Отсутствии чёткого алгоритма решения. Его вы должны придумать сами на основе полученных ранее знаний (синтаксиса и инструментов).

Объединении большинства тем изученного модуля.

Предполагаемом большом объёме решения.

Задание "Средний балл":

Вам необходимо решить задачу из реальной жизни: "школьные учителя устали подсчитывать вручную **средний балл каждого ученика**, поэтому вам предстоит автоматизировать этот процесс":

На вход даны следующие данные:

Список: **grades = [[5, 3, 3, 5, 4], [2, 2, 2, 3], [4, 5, 5, 2], [4, 4, 3], [5, 5, 5, 4, 5]]**

Множество: **students = {'Johnny', 'Bilbo', 'Steve', 'Khendrik', 'Aaron'}**

Список ***grades*** содержит списки оценок для каждого ученика в алфавитном порядке.

Например:**'Aaron' - [5, 3, 3, 5, 4]**

Множество ***students*** содержит неупорядоченную последовательность имён всех учеников в классе.

Напишите программу, которая составляет словарь, используя grades и students, где ключом будет имя ученика, а значением - его средний балл.

Вывод в консоль:

{'Aaron': 4.0, 'Bilbo': 2.25, 'Johhny': 4.0, 'Khendrik': 3.6666666666666665, 'Steve': 4.8}

Примечания:

Самостоятельно составлять (вручную) словарь не нужно (только изначально пустой).

Для решения задачи нужно вспомнить функции sum, len и др. (подумать самому).

Помните, что множество не является упорядоченной последовательностью. (нужен перевод в другой тип).