Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Дисциплина: Современные языки программирования

ЛАБОРАТОРНАЯ РАБОТА № 1

Вариант № 2

Выполнил:

Атаев И.М. гр. 910101

Проверила:

Василькова А.Н.

Минск 2022

**Задание:** Визуализация данных. Matpotlib. Разработать диаграммы, графики, обосновать вид представления данных (источник <http://dataportal.belstat.gov.by/>)

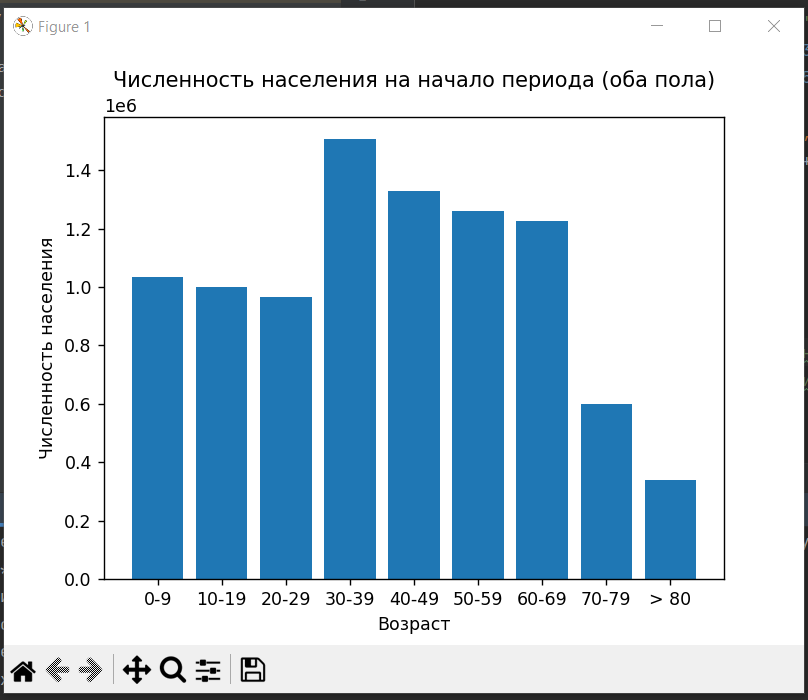
**Задание №1.1 по варианту:** Демографическая и социальная статистика. Статистика населения. Показатели: Численность населения на начало периода, Среднегодовая численность населения

**Листинг кода:**

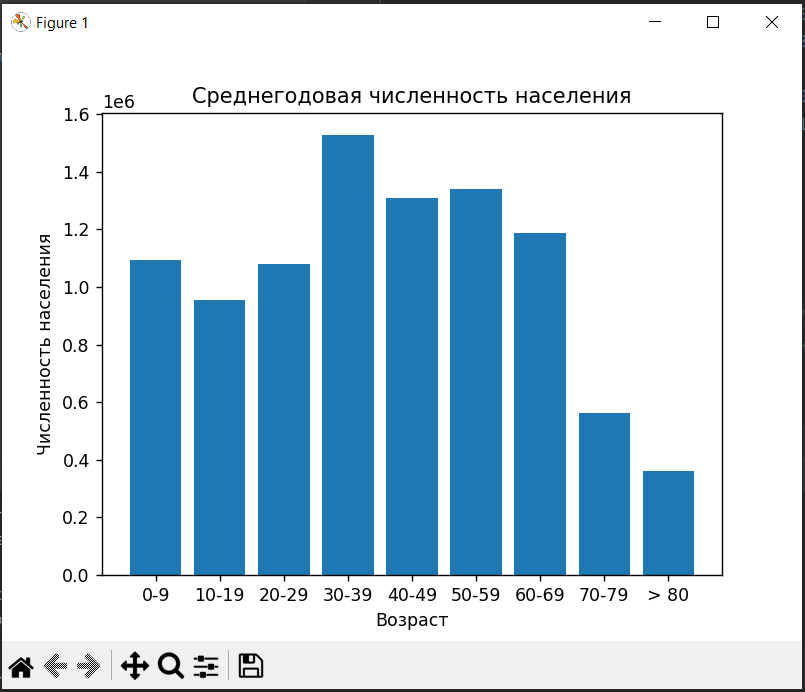
**1 график**

import matplotlib.pyplot as plt  
  
age = ['0-9', '10-19', '20-29', '30-39', '40-49', '50-59', '60-69', '70-79', '> 80']  
people\_new\_year = [444851 + 588347, 533786 + 464731, 445229 + 520539, 724861 + 781956, 689143 +  
 639636, 615008 + 645018, 685683 + 540113, 407613 + 190090, 338920]  
  
people\_average\_year = [535311 + 559328, 501231 + 455275, 453187 + 627098, 783689 + 744104, 668849 +  
 639488, 628396 + 711447, 672803 + 514700, 329546 + 233908, 361398]  
  
  
def answer(choose):  
 if choose == 1:  
 plt.bar(age, people\_new\_year)  
 plt.xlabel('Возраст')  
 plt.ylabel('Численность населения')  
 plt.title('Численность населения на начало периода (оба пола)')  
 plt.show()  
 return 1  
  
 if choose == 2:  
 plt.bar(age, people\_average\_year)  
 plt.xlabel('Возраст')  
 plt.ylabel('Численность населения')  
 plt.title('Среднегодовая численность населения')  
 plt.show()  
 return 1  
  
 if choose == 3:  
 return 0  
  
 if choose not in (1, 2, 3):  
 raise Exception('Неправильно')  
  
  
def start():  
 flag = True  
 while flag:  
 choose = int(input(  
 '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\nВыберите график:'  
 '\n1. Численность населения на начало периода 2022 года. '  
 '\n2. Среднегодовая численность населения'  
 '\n3. Выход'  
 '\nВаш выбор: '))  
 flag = answer(choose)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 start()

**Результаты работы программы:**

****

*Рис. 1 – Гистограмма численность населения на начало периода.*

****

*Рис. 2 – Гистограмма Среднегодовая численность населения.*

**Задание №1.2 по варианту:**  Создать класс, который проверяет число и определяет что это – простое число, число фибоначчи, комплексное число, целое, вещественное.

**Листинг кода:**

from tkinter import \*  
  
  
def clicked():  
 number = txt.get()  
 c = Checker(number)  
 c.str\_to\_type()  
  
 is\_int = Label(window, text=str(c.is\_int), fg='#0f0' if c.is\_int else '#f00')  
 is\_int.grid(column=1, row=1)  
  
 is\_float = Label(window, text=str(c.is\_float), fg='#0f0' if c.is\_float else '#f00')  
 is\_float.grid(column=1, row=2)  
  
 is\_fib = Label(window, text=str(c.is\_fibonacci()), fg='#0f0' if c.is\_fibonacci() else '#f00')  
 is\_fib.grid(column=1, row=3)  
  
 is\_complex = Label(window, text=str(c.is\_complex), fg='#0f0' if c.is\_complex else '#f00')  
 is\_complex.grid(column=1, row=4)  
  
 is\_prime = Label(window, text=str(c.is\_prime()), fg='#0f0' if c.is\_prime() else '#f00')  
 is\_prime.grid(column=1, row=5)  
  
  
def fibonacci\_of(n):  
  
 if n in (0, 1):  
 return n  
  
 return fibonacci\_of(n - 1) + fibonacci\_of(n - 2)  
  
  
def fibonacci\_list(n) -> list:  
 return [fibonacci\_of(i + 1) for i in range(n)]  
  
  
class Checker:  
 num = 0  
 is\_int = False  
 is\_float = False  
 is\_complex = False  
  
 def \_\_init\_\_(self, num):  
 self.num = num  
  
 def str\_to\_type(self) -> None:  
  
 self.num = ''.join(self.num.split())  
  
 try:  
 self.num = int(self.num)  
 self.is\_int = True  
 except ValueError:  
  
 try:  
 self.num = float(self.num)  
 self.is\_float = True  
 except ValueError:  
  
 try:  
  
 if 'j' in self.num and ('+' in self.num or '-' in self.num):  
  
 self.num = complex(self.num)  
 self.is\_complex = True  
  
 except ValueError:  
 pass  
  
 def is\_prime(self) -> bool:  
  
 if not isinstance(self.num, int):  
 return False  
 else:  
 if self.num < 2:  
 return False  
 else:  
 for i in range(2, self.num):  
 if (self.num % i) == 0:  
 return False  
 return True  
  
 def is\_fibonacci(self) -> bool:  
  
 if not isinstance(self.num, int):  
 return False  
 else:  
  
 if self.num > 0:  
  
 fib\_lst = fibonacci\_list(30)  
  
 for f in fib\_lst:  
 if self.num == f:  
 return True  
 if f > self.num:  
 return False  
 else:  
 return False  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 window = Tk()  
 window.title("Number checker")  
 window.geometry('300x150')  
 lbl = Label(window, text="Введите число:")  
 lbl.grid(column=0, row=0)  
 int\_lbl = Label(window, text='Целочисленное:')  
 int\_lbl.grid(column=0, row=1)  
 float\_lbl = Label(window, text='Вещественное:')  
 float\_lbl.grid(column=0, row=2)  
 fib\_lbl = Label(window, text='Число Фибоначчи:')  
 fib\_lbl.grid(column=0, row=3)  
 complex\_lbl = Label(window, text='Комплексное:')  
 complex\_lbl.grid(column=0, row=4)  
 prime\_lbl = Label(window, text='Простое число:')  
 prime\_lbl.grid(column=0, row=5)  
 btn = Button(window, text="Проверить число", command=clicked)  
 btn.grid(column=2, row=0)  
 txt = Entry(window, width=10)  
 txt.grid(column=1, row=0)  
  
 window.mainloop()

**1 график**

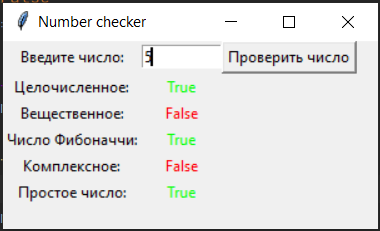
****

Рисунок 1.2.1

**Вывод:** В ходе выполнения лабораторной работы ознакомились с библиотекой Matpotlib для визуализации данных, а именно работа с гистограммой. Повторили работу с классами в Python.