

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ имени  
И.С.ТУРГЕНЕВА»**

**ИНСТИТУТ ПРИБОРОСТРОЕНИЯ, АВТОМАТИЗАЦИИ И  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Кафедра информационных систем и цифровых технологий

**Захарова О.В.**

**Методические указания к лабораторным работам  
по дисциплине «Программирование на языке  
Python»**

*для направлений подготовки: 09.03.01, 09.03.02,  
09.03.03, 09.03.04*

**Орел 2021**

## Содержание

<b>Лабораторная работа № 1</b>	
<b>Основные типы данных. Управляющие конструкции</b>	<b>3</b>
<b>Лабораторная работа № 2</b>	
<b>Функциональное программирование</b>	<b>17</b>
<b>Лабораторная работа № 3</b>	
<b>Объектно-ориентированное программирование</b>	<b>18</b>
<b>Лабораторная работа № 4</b>	
<b>Разработка графического интерфейса</b>	<b>24</b>
<b>Лабораторная работа № 5</b>	
<b>Работа с файлами</b>	<b>25</b>
<b>Лабораторная работа № 6</b>	
<b>Сетевое программирование</b>	<b>26</b>
<b>Лабораторная работа № 7</b>	
<b>Разработка приложения, взаимодействующего с базой данных</b>	<b>31</b>

В методических указаниях к лабораторным работам задания представлены по вариантам. Номер варианта задания выдает преподаватель каждому студенту.

## Лабораторная работа № 1

### Основные типы данных. Управляющие конструкции

#### Вариант 1

1. Написать программу, вычисляющую выражение:  $\frac{3 \cos^n \left( x - \frac{\pi}{6} \right)}{\frac{1}{2} + \sin y}$ . Значения всех

переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль `math` и управляющие конструкции (не использовать `lambda`).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в указанную пользователем позицию списка (добавлять элементы разных типов);
- 3) удаление указанного пользователем элемента из списка;
- 4) сформировать кортеж, состоящий из элементов, стоящих на четных позициях списка; вывести содержимое кортежа на экран;
- 5) найти сумму всех целочисленных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать количество слов в строке;
- 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран множество, полученное путем объединения множеств M1 и M2;
- 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с ключом больше 5.

Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «R» (площадь прямоугольника), «T» (площадь прямоугольного треугольника), «M» (площадь многоугольника), «E» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать `lambda`.

#### Вариант 2

1. Написать программу, вычисляющую выражение:  $\sqrt{a_0 + a_1 x^x + a_2 \sqrt[3]{|\sin x|}} + \frac{5a_0}{e^{x^5}}$ .

Значения всех переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль `math` и управляющие конструкции (не использовать `lambda`).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в конец списка (добавлять элементы разных типов);
- 3) удаление указанного пользователем элемента из списка;
- 4) сформировать кортеж, состоящий из элементов, стоящих на нечетных позициях списка; вывести содержимое кортежа на экран;

- 5) найти произведение всех вещественных элементов списка;
  - 6) сформировать строку из значений элементов списка и посчитать количество знаков препинания в строке;
  - 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран множество, полученное путем пересечения множеств M1 и M2;
  - 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с ключом меньше 5.
- Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «S» (площадь квадрата), «T» (площадь треугольника), «R» (площадь ромба), «E» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать lambda.

### Вариант 3

1. Написать программу, вычисляющую выражение: 
$$\frac{2}{\sqrt{x}} \cdot \frac{x - \frac{m}{2}}{\sqrt[3]{\left(x - \frac{m}{2}\right)^n + x^{m^n}}}$$
. Значения всех

переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль math и управляющие конструкции (не использовать lambda).

2. Создать список с элементами разных типов. Функционал программы:
  - 1) показать значения списка на экране;
  - 2) добавление нового элементов в начало списка (добавлять элементы разных типов);
  - 3) удаление последнего элемента списка;
  - 4) сформировать кортеж, состоящий из вещественных элементов списка; вывести содержимое кортежа на экран;
  - 5) найти сумму всех целочисленных положительных элементов списка;
  - 6) сформировать строку из значений элементов списка и посчитать количество цифр в строке;
  - 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран разницу множеств M1 и M2;
  - 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с четными значениями ключа.

Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «P» (площадь параллелограмма), «E» (площадь эллипса), «M» (площадь многоугольника), «Q» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать lambda.

**Вариант 4**

1. Написать программу, вычисляющую выражение:  $\sqrt[n]{y}(\sin y^2 + \cos^2 y) + \frac{x^{n^m}}{m}$ . Значения всех переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль `math` и управляющие конструкции (не использовать `lambda`).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в указанную пользователем позицию списка (добавлять элементы разных типов);
- 3) удаление последнего элемента списка;
- 4) сформировать кортеж, состоящий из строковых элементов списка; вывести содержимое кортежа на экран;
- 5) найти сумму всех целочисленных отрицательных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать количество букв в строке;
- 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран разницу множеств M2 и M1;
- 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с нечетными значениями ключа.

Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «R» (площадь прямоугольника), «T» (площадь треугольника), «E» (площадь эллипса), «Q» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать `lambda`.

**Вариант 5**

1. Написать программу, вычисляющую выражение:  $\sqrt{|\cos x|^n + \frac{e^{n^3}}{\ln x} + \sqrt[n]{|\sin x|}}$ . Значения всех переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль `math` и управляющие конструкции (не использовать `lambda`).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в конец списка (добавлять элементы разных типов);
- 3) удаление указанного пользователем элемента из списка;
- 4) сформировать кортеж, состоящий из вещественных положительных элементов списка; вывести содержимое кортежа на экран;
- 5) найти произведение всех целочисленных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать сколько раз встречается в строке указанное пользователем слово;

- 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран симметричную разницу множеств M1 и M2;
- 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с нечетными значениями ключа.

Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «S» (площадь квадрата), «Т» (площадь трапеции), «Р» (площадь параллелограмма), «Е» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать lambda.

### Вариант 6

1. Написать программу, вычисляющую выражение: 
$$\frac{4-m}{\sqrt[n]{x-2}} \cdot \frac{m^{2^n}}{2 \sqrt[4]{\left(x-\frac{m}{2}\right)^n + m}}$$
. Значения всех

переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль math и управляющие конструкции (не использовать lambda).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в указанную пользователем позицию списка (добавлять элементы разных типов);
- 3) удаление последнего элемента списка;
- 4) сформировать кортеж, состоящий из элементов, стоящих на четных позициях списка; вывести содержимое кортежа на экран;
- 5) найти сумму всех вещественных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать количество слов в строке, начинающихся на указанную пользователем букву;
- 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран разницу множеств M2 и M1;
- 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с ключом больше 3.

Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «С» (площадь круга), «R» (площадь ромба), «Р» (площадь многоугольника), «Е» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать lambda.

**Вариант 7**

1. Написать программу, вычисляющую выражение:  $\sqrt[5]{\sin(x^n + \sqrt[n]{y})} + \sqrt[3]{\frac{e^{x^4}}{\cos y}}$ . Значения всех

переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль `math` и управляющие конструкции (не использовать `lambda`).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
  - 2) добавление нового элементов в конец списка (добавлять элементы разных типов);
  - 3) удаление указанного пользователем элемента из списка;
  - 4) сформировать кортеж, состоящий из элементов, стоящих на нечетных позициях списка; вывести содержимое кортежа на экран;
  - 5) найти произведение всех вещественных элементов списка;
  - 6) сформировать строку из значений элементов списка и посчитать количество арифметических операций в строке;
  - 7) задать с клавиатуры множество `M1`, сформировать множество `M2` из списка; вывести на экран множество, полученное путем пересечения множеств `M1` и `M2`;
  - 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с ключом меньше 3.
- Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «R» (площадь прямоугольного треугольника), «T» (площадь трапеции), «S» (площадь квадрата), «E» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать `lambda`.

**Вариант 8**

1. Написать программу, вычисляющую выражение:  $\frac{\operatorname{tg} x}{x} + \sqrt[n]{\sin x^n} + e^{(n+1)(x+2)}$ .

Значения всех переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль `math` и управляющие конструкции (не использовать `lambda`).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в начало списка (добавлять элементы разных типов);
- 3) удаление первого элемента списка;
- 4) сформировать кортеж, состоящий из вещественных элементов списка больших 100; вывести содержимое кортежа на экран;
- 5) найти сумму всех целочисленных отрицательных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать количество цифр в строке;
- 7) задать с клавиатуры множество `M1`, сформировать множество `M2` из списка; вывести на экран разницу множеств `M2` и `M1`;

8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с четными значениями ключа. Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «Т» (площадь равнобедренного треугольника), «С» (площадь круга), «Р» (площадь параллелограмма), «Е» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать lambda.

### Вариант 9

1. Написать программу, вычисляющую выражение:  $\frac{|x-y|}{(1+2x)^{a^y}-1} - e^{\sqrt{1+y}}$ . Значения всех

переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль math и управляющие конструкции (не использовать lambda).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в указанную пользователем позицию списка (добавлять элементы разных типов);
- 3) удаление указанного пользователем элемента из списка;
- 4) сформировать кортеж, состоящий из строковых элементов списка; вывести содержимое кортежа на экран;
- 5) найти сумму всех целочисленных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать количество цифр в строке;
- 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран разницу множеств M2 и M1;
- 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с нечетными значениями ключа.

Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «Т» (площадь равностороннего треугольника), «Е» (площадь эллипса), «Р» (площадь ромба), «Q» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать lambda.

### Вариант 10

1. Написать программу, вычисляющую выражение:  $\frac{5a^{nx}}{n+x} - \sqrt{\cos x^{3^n}}$ . Значения всех

переменных задавать с клавиатуры. При задании неподходящих данных программа



должна показать сообщение об ошибке. Использовать модуль `math` и управляющие конструкции (не использовать `lambda`).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в конец списка (добавлять элементы разных типов);
- 3) удаление указанного пользователем элемента из списка;
- 4) сформировать кортеж, состоящий из вещественных элементов списка меньших 100; вывести содержимое кортежа на экран;
- 5) найти произведение всех целочисленных отрицательных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать сколько раз встречается в строке указанное пользователем слово;
- 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран симметричную разницу множеств M1 и M2;
- 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с нечетными значениями ключа.

Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «R» (площадь прямоугольника), «T» (площадь прямоугольного треугольника), «M» (площадь многоугольника), «E» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать `lambda`.

## Вариант 11

1. Написать программу, вычисляющую выражение: 
$$\frac{4}{\sqrt[n]{x}} \cdot \frac{m^{2^n}}{2 \sqrt[4]{\left(x - \frac{m}{2}\right)^n + m}}$$
. Значения всех

переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль `math` и управляющие конструкции (не использовать `lambda`).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в указанную пользователем позицию списка (добавлять элементы разных типов);
- 3) удаление указанного пользователем элемента из списка;
- 4) сформировать кортеж, состоящий из элементов, стоящих на четных позициях списка; вывести содержимое кортежа на экран;
- 5) найти сумму всех целочисленных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать количество слов в строке;
- 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран множество, полученное путем объединения множеств M1 и M2;
- 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с ключом больше 5.

Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «S» (площадь квадрата), «T» (площадь треугольника), «R» (площадь ромба), «E» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать lambda.

### Вариант 12

1. Написать программу, вычисляющую выражение:  $|\cos a|^{n^m} + \frac{e^{n^3}}{\ln a} + \sqrt[n]{\sin a^2}$ . Значения всех переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль math и управляющие конструкции (не использовать lambda).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в конец списка (добавлять элементы разных типов);
- 3) удаление указанного пользователем элемента из списка;
- 4) сформировать кортеж, состоящий из элементов, стоящих на нечетных позициях списка; вывести содержимое кортежа на экран;
- 5) найти произведение всех вещественных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать количество знаков препинания в строке;
- 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран множество, полученное путем пересечения множеств M1 и M2;
- 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с ключом меньше 5.

Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «T» (площадь трапеции), «R» (площадь ромба), «E» (площадь равностороннего треугольника), «Q» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать lambda.

### Вариант 13

1. Написать программу, вычисляющую выражение:  $5\cos x^{n^m} + \sqrt[n]{\sin^2 x} - \frac{e^{n^4}}{\ln x}$ . Значения всех переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль math и управляющие конструкции (не использовать lambda).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в начало списка (добавлять элементы разных типов);

- 3) удаление последнего элемента списка;
  - 4) сформировать кортеж, состоящий из вещественных элементов списка; вывести содержимое кортежа на экран;
  - 5) найти сумму всех целочисленных положительных элементов списка;
  - 6) сформировать строку из значений элементов списка и посчитать количество цифр в строке;
  - 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран разницу множеств M1 и M2;
  - 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с четными значениями ключа.
- Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «М» (площадь многоугольника), «Т» (площадь равнобедренного треугольника), «Р» (площадь параллелограмма), «Е» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать lambda.

#### Вариант 14

1. Написать программу, вычисляющую выражение:  $\frac{5a^{n^x}}{\ln a} + \sqrt{|\cos b^n|} - 3\sin^2 a$ . Значения всех переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль math и управляющие конструкции (не использовать lambda).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в указанную пользователем позицию списка (добавлять элементы разных типов);
- 3) удаление последнего элемента списка;
- 4) сформировать кортеж, состоящий из строковых элементов списка; вывести содержимое кортежа на экран;
- 5) найти сумму всех целочисленных отрицательных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать количество букв в строке;
- 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран разницу множеств M2 и M1;
- 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с нечетными значениями ключа.

Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «С» (площадь круга), «Е» (площадь эллипса), «М» (площадь многоугольника), «Q» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать lambda.

**Вариант 15**

1. Написать программу, вычисляющую выражение:  $\frac{e^{\sqrt{1+m}} \cdot |x-y|}{(1+2x)^{a^m} - 1} - (x+y)^{m^a}$ . Значения всех

переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль `math` и управляющие конструкции (не использовать `lambda`).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в конец списка (добавлять элементы разных типов);
- 3) удаление указанного пользователем элемента из списка;
- 4) сформировать кортеж, состоящий из вещественных положительных элементов списка; вывести содержимое кортежа на экран;
- 5) найти произведение всех целочисленных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать сколько раз встречается в строке указанное пользователем слово;
- 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран симметричную разницу множеств M1 и M2;
- 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с нечетными значениями ключа.

Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «С» (площадь круга), «Т» (площадь трапеции), «Р» (площадь прямоугольника), «Е» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать `lambda`.

**Вариант 16**

1. Написать программу, вычисляющую выражение:  $\frac{3a^{n^x}}{\sin a^2} - \sqrt{|\cos b^n|} + \frac{1}{a}$ . Значения всех

переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль `math` и управляющие конструкции (не использовать `lambda`).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в конец списка (добавлять элементы разных типов);
- 3) удаление указанного пользователем элемента из списка;
- 4) сформировать кортеж, состоящий из вещественных элементов списка меньших 100; вывести содержимое кортежа на экран;
- 5) найти произведение всех целочисленных отрицательных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать сколько раз встречается в строке указанное пользователем слово;
- 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран симметричную разницу множеств M1 и M2;

8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с нечетными значениями ключа.

Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «М» (площадь многоугольника), «Е» (площадь эллипса), «Р» (площадь параллелограмма), «Q» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать lambda.

### Вариант 17

1. Написать программу, вычисляющую выражение:  $\sqrt{x^{n^m} + y \cdot \sqrt{|\cos x|}} + \frac{4m}{e^{m^n} - 1}$ . Значения

всех переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль math и управляющие конструкции (не использовать lambda).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в указанную пользователем позицию списка (добавлять элементы разных типов);
- 3) удаление указанного пользователем элемента из списка;
- 4) сформировать кортеж, состоящий из строковых элементов списка; вывести содержимое кортежа на экран;
- 5) найти сумму всех целочисленных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать количество цифр в строке;
- 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран разницу множеств M2 и M1;
- 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с нечетными значениями ключа.

Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «R» (площадь ромба), «Т» (площадь треугольника), «S» (площадь квадрата), «Е» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать lambda.

### Вариант 18

1. Написать программу, вычисляющую выражение:  $\frac{5a^{nx}}{|b+c|} - \sqrt{|\sin x^{n^x}|}$ . Значения всех

переменных задавать с клавиатуры. При вводе неподходящих данных программа должна

показать сообщение об ошибке. Использовать модуль `math` и управляющие конструкции (не использовать `lambda`).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
  - 2) добавление нового элементов в начало списка (добавлять элементы разных типов);
  - 3) удаление первого элемента списка;
  - 4) сформировать кортеж, состоящий из вещественных элементов списка больших 100; вывести содержимое кортежа на экран;
  - 5) найти сумму всех целочисленных отрицательных элементов списка;
  - 6) сформировать строку из значений элементов списка и посчитать количество цифр в строке;
  - 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран разницу множеств M2 и M1;
  - 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с четными значениями ключа.
- Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «Е» (площадь эллипса), «Т» (площадь треугольника), «R» (площадь прямоугольника), «Q» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать `lambda`.

### Вариант 19

1. Написать программу, вычисляющую выражение:  $\frac{|a-b|}{(2+5a)^{b^3}-1} - e^{\sqrt{1+a}}$ . Значения всех

переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль `math` и управляющие конструкции (не использовать `lambda`).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
  - 2) добавление нового элементов в указанную пользователем позицию списка (добавлять элементы разных типов);
  - 3) удаление последнего элемента списка;
  - 4) сформировать кортеж, состоящий из элементов, стоящих на четных позициях списка; вывести содержимое кортежа на экран;
  - 5) найти сумму всех вещественных элементов списка;
  - 6) сформировать строку из значений элементов списка и посчитать количество слов в строке, начинающихся на указанную пользователем букву;
  - 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран разницу множеств M2 и M1;
  - 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с ключом больше 3.
- Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «Р» (площадь многоугольника), «R» (площадь ромба), «С» (площадь круга), «Е» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать lambda.

### Вариант 20

1. Написать программу, вычисляющую выражение:  $\frac{5 \sin^2 \left( m \sqrt{x - \frac{\pi}{6}} \right)}{\frac{1}{2} + \cos y}$ . Значения всех

переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль math и управляющие конструкции (не использовать lambda).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в конец списка (добавлять элементы разных типов);
- 3) удаление указанного пользователем элемента из списка;
- 4) сформировать кортеж, состоящий из вещественных положительных элементов списка; вывести содержимое кортежа на экран;
- 5) найти произведение всех целочисленных элементов списка;
- 6) сформировать строку из значений элементов списка и посчитать сколько раз встречается в строке указанное пользователем слово;
- 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран симметричную разницу множеств M1 и M2;
- 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с нечетными значениями ключа.

Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «Р» (площадь параллелограмма), «Т» (площадь трапеции), «S» (площадь квадрата), «Е» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать lambda.

### Вариант 21

1. Написать программу, вычисляющую выражение:  $\frac{\lg x}{x} + \sqrt[n]{\cos x^n} + e^{(n+1)x}$ . Значения всех переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль math и управляющие конструкции (не использовать lambda).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
- 2) добавление нового элементов в указанную пользователем позицию списка (добавлять элементы разных типов);
- 3) удаление указанного пользователем элемента из списка;

- 4) сформировать кортеж, состоящий из элементов, стоящих на четных позициях списка; вывести содержимое кортежа на экран;
  - 5) найти сумму всех целочисленных элементов списка;
  - 6) сформировать строку из значений элементов списка и посчитать количество слов в строке;
  - 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран множество, полученное путем объединения множеств M1 и M2;
  - 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с ключом больше 5.
- Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «S» (площадь квадрата), «T» (площадь трапеции), «R» (площадь прямоугольного треугольника), «E» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать lambda.

### Вариант 22

1. Написать программу, вычисляющую выражение:  $\sqrt{|\sin x|^n + \frac{e^{n^3}}{\ln x} + \sqrt[n]{|\cos x|}}$ . Значения всех переменных задавать с клавиатуры. При вводе неподходящих данных программа должна показать сообщение об ошибке. Использовать модуль math и управляющие конструкции (не использовать lambda).

2. Создать список с элементами разных типов. Функционал программы:

- 1) показать значения списка на экране;
  - 2) добавление нового элементов в начало списка (добавлять элементы разных типов);
  - 3) удаление последнего элемента списка;
  - 4) сформировать кортеж, состоящий из вещественных элементов списка; вывести содержимое кортежа на экран;
  - 5) найти сумму всех целочисленных положительных элементов списка;
  - 6) сформировать строку из значений элементов списка и посчитать количество цифр в строке;
  - 7) задать с клавиатуры множество M1, сформировать множество M2 из списка; вывести на экран разницу множеств M1 и M2;
  - 8) получить из списка словарь, ключом каждого элемента сделать позицию элемента в словаре; построчно отобразить на экране элементы словаря с четными значениями ключа.
- Программа должна обеспечить возможность выполнять пункты 1-8 до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла).

3. Написать программу, вычисляющую площадь фигур. Меню программы имеет следующие пункты: «R» (площадь ромба), «E» (площадь эллипса), «T» (площадь равностороннего треугольника), «Q» (выход из программы). В случае ввода неверных данных выдать сообщение об ошибке. Для проверки условий использовать условный оператор. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока пользователь не решит из неё выйти (использовать оператор цикла). Не использовать lambda.



## **Лабораторная работа № 2**

### **Функциональное программирование**

1. Написать программу, вычисляющую выражение (лабораторная работа № 1, задание 1). Значения всех переменных задавать с клавиатуры. При вводе неверных данных выдать сообщение об ошибке. Использовать модуль `math`. При разработке программы не использовать управляющие конструкции. Использовать функции первого класса и высшего порядка.

2. Написать программу, вычисляющую площадь фигур. Функционал программы разработать в соответствии с 3-м заданием первой лабораторной работы (например, для первого варианта: вычисление площади прямоугольника («R»); вычисление площади прямоугольного треугольника («T»); вычисление площади многоугольника («M»), выход из программы («E»); в случае ввода неверных данных выдать сообщение об ошибке).

Входные данные задать в виде одного списка. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока в списке есть входные данные. При разработке программы не использовать управляющие конструкции. Использовать функции первого класса и высшего порядка.

Пример списка с входными данными:

`L = [ 'R', 'r', 'M', 'T', 't', 'E'], [1, 2, 3, 4, 5, 6], <входные данные > ]`

### **Лабораторная работа № 3** **Объектно-ориентированное программирование**

#### **Вариант 1**

В программе разработать класс «Сотрудник» с обязательными атрибутами: табельный номер, ФИО, адрес, телефон, пол, дата рождения, должность, оклад, наличие высшего образования. Вводимую информацию о сотрудниках хранить в списке объектов класса «Сотрудник».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск сотрудников с высшим образованием.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

#### **Вариант 2**

В программе разработать класс «Учитель» с обязательными атрибутами: табельный номер, Фамилия, Имя, Отчество, пол, дата рождения, адрес, телефон, преподаваемый предмет, стаж работы. Вводимую информацию об учителях хранить в списке объектов класса «Учитель».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск учителей со стажем работы больше указанного пользователем.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

#### **Вариант 3**

В программе разработать класс «Ученик» с обязательными атрибутами: код, ФИО, адрес, телефон, пол, дата рождения, класс. Вводимую информацию об учениках хранить в списке объектов класса «Ученик».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск учеников из указанного пользователем класса.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

#### **Вариант 4**

В программе разработать класс «Книга» с обязательными атрибутами: ISBN, название, авторы, издательство, год выпуска, количество страниц, стоимость. Вводимую информацию о книгах хранить в списке объектов класса «Книга».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск книг указанного пользователем года выпуска.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

**Вариант 5**

В программе разработать класс «Преподаватель» с обязательными атрибутами: табельный номер, ФИО, пол, дата рождения, адрес, телефон, преподаваемая дисциплина, стаж работы. Вводимую информацию о преподавателях хранить в списке объектов класса «Преподаватель».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск преподавателей преподающих указанную дисциплину.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

**Вариант 6**

В программе разработать класс «Товар» с обязательными атрибутами: артикул, название, отдел, стоимость, описание. Вводимую информацию о товарах хранить в списке объектов класса «Товар».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск товаров из указанного отдела.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

**Вариант 7**

В программе разработать класс «Лекарство» с обязательными атрибутами: артикул, название, стоимость, наличие рецепта, описание. Вводимую информацию о лекарствах хранить в списке объектов класса «Лекарство».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск лекарств, продающихся без рецепта.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

**Вариант 8**

В программе разработать класс «Студент» с обязательными атрибутами: номер зачетки, Фамилия, Имя, Отчество, пол, дата рождения, телефон, адрес, группа. Вводимую информацию о студентах хранить в списке объектов класса «Студент».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск студентов из указанной пользователем группы.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

**Вариант 9**

В программе разработать класс «Участник олимпиады» с обязательными атрибутами: код, ФИО, пол, балл за задание №1, балл за задание №2, балл за задание №3, дата рождения, телефон. Вводимую информацию об участниках хранить в списке объектов класса «Участник олимпиады».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск участника олимпиады с максимальным суммарным баллом.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

**Вариант 10**

В программе разработать класс «Фильм» с обязательными атрибутами: код, название, режиссер, год выпуска, актеры. Вводимую информацию о фильмах хранить в списке объектов класса «Фильм».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск фильмов указанного пользователем года выпуска.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

**Вариант 11**

В программе разработать класс «Блюдо» с обязательными атрибутами: код, название, состав, вес, стоимость. Вводимую информацию о блюдах хранить в списке объектов класса «Блюдо».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск всех блюд с названием на указанную пользователем букву.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

**Вариант 12**

В программе разработать класс «Автомобиль» с обязательными атрибутами: код, марка, модель, год выпуска, тип кузова, стоимость. Вводимую информацию об автомобилях хранить в списке объектов класса «Автомобиль».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск всех автомобилей указанного пользователем года выпуска.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

**Вариант 13**

В программе разработать класс «Страна» с обязательными атрибутами: код, название, численность населения, площадь, столица, язык. Вводимую информацию о странах хранить в списке объектов класса «Страна».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск всех стран в названии которых встречается указанная пользователем буква.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

**Вариант 14**

В программе разработать класс «Животное» с обязательными атрибутами: код, вид, кличка, окрас шерсти, дата рождения, ФИО владельца, телефон владельца. Вводимую информацию о животных хранить в списке объектов класса «Животное».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск информации о животном по указанным ФИО владельца.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

**Вариант 15**

В программе разработать класс «Врач» с обязательными атрибутами: табельный номер, фамилия, имя, отчество, специальность, стаж работы, время приема, кабинет, телефон. Вводимую информацию о врачах хранить в списке объектов класса «Врач».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск информации о врачах с указанным пользователем стажем работы.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

**Вариант 16**

В программе разработать класс «Город» с обязательными атрибутами: код, название, страна, численность населения, дата основания. Вводимую информацию о городах хранить в списке объектов класса «Город».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск всех городов в названии которых встречается указанная пользователем буква.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

**Вариант 17**

В программе разработать класс «Пациент» с обязательными атрибутами: номер полиса, фамилия, имя, отчество, пол, дата рождения, адрес, информация о последнем

посещении врача. Вводимую информацию о пациентах хранить в списке объектов класса «Пациент».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск информации о пациентах с фамилией на указанную пользователем букву.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

### **Вариант 18**

В программе разработать класс «Процессор» с обязательными атрибутами: артикул, фирма-производитель, название, количество ядер, тактовая частота, стоимость. Вводимую информацию о процессорах хранить в списке объектов класса «Процессор».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск информации о процессорах указанную пользователем фирмы-производителя.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

### **Вариант 19**

В программе разработать класс «Микроконтроллер» с обязательными атрибутами: артикул, фирма-производитель, название, разрядность, тактовая частота, архитектура, стоимость. Вводимую информацию о микроконтроллерах хранить в списке объектов класса «Микроконтроллер».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск информации о микроконтроллерах указанной пользователем архитектуры.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

### **Вариант 20**

В программе разработать класс «Дисциплина» с обязательными атрибутами: код, название, количество лекций, количество лабораторных работ, количество практических занятий, наличие курсовой работы, форма контроля (зачет, экзамен), ФИО преподавателя. Вводимую информацию о дисциплинах хранить в списке объектов класса «Дисциплина».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск информации о дисциплинах по которым сдается зачет.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

### **Вариант 21**

В программе разработать класс «Университет» с обязательными атрибутами: код, название, город, количество студентов, адрес, телефон, ФИО ректора. Вводимую информацию об университетах хранить в списке объектов класса «Университет».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск информации об университетах, расположенных в указанном пользователем городе.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

### **Вариант 22**

В программе разработать класс «Абитуриент» с обязательными атрибутами: код, ФИО, специальность, адрес, телефон, баллы за предмет №1, баллы за предмет №2, баллы за предмет №3. Вводимую информацию об абитуриентах хранить в списке объектов класса «Абитуриент».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск информации об абитуриенте, набравшем максимальных суммарный балл.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

### **Лабораторная работа № 4**

#### **Разработка графического интерфейса**

Разработать и реализовать графический интерфейс для задания лабораторной работы № 3.

Обязательные элементы графического интерфейса: надписи, кнопки, текстовые поля, выпадающий список, чекбоксы и/или радиокнопки, всплывающее окно с сообщением.



### **Лабораторная работа № 5**

#### **Работа с файлами**

В программу лабораторной работы № 4 добавить главное меню с пунктами «Файл», «Справка».

Пункт главного меню «Файл» должен включать подпункты «Создать» (очистка формы для ввода новых данных), «Открыть» (считывание из файла (по указанному пути) информации об объектах и отображение её на форме), «Сохранить» (обновление текущего файла; если файл не создан, то действия пункта «Сохранить как...»), «Сохранить как...» (сохранение информации об объектах в файл по указанному пути), «Выход» (дружественный выход из программы).

Пункт главного меню «Справка» должен показывать информацию о разработчике программы.

## **Лабораторная работа № 6** **Сетевое программирование**

### **Вариант 1**

Разработать клиент-серверное приложение для обработки текста.

Требования к клиенту:

- отправка на сервер введенного пользователем текста;
- получение обработанного сервером текста;
- удобный графический интерфейс.

Требования к серверу:

- обработка полученного текста по следующим правилам: в начале предложения не должно быть знаков препинания; в конце текста должна стоять одна точка; после каждой точки в тексте должен стоять один пробел, за которым следует заглавная буква.

### **Вариант 2**

Разработать клиент-серверное приложение (чат).

Требования к клиенту:

- отправка на сервер введенного пользователем сообщения;
- получение сообщений, отправленных другими клиентами;
- удобный графический интерфейс.

Требования к серверу:

- организация чата между клиентами (до 3 клиентов), а именно отправка всем клиентам полученных сервером сообщений;
- полученные сообщения перед отправкой клиентам необходимо преобразовывать к верхнему регистру.

Клиенты должны последовательно отправлять сообщения.

### **Вариант 3**

Разработать клиент-серверное приложение для обработки текста.

Требования к клиенту:

- отправка на сервер введенного пользователем текста и двух слов;
- получение обработанного сервером текста;
- удобный графический интерфейс.

Требования к серверу:

- обработка полученного текста по следующему правилу: заменить в полученном тексте первое слово на второе, а второе слово на первое.

### **Вариант 4**

Разработать клиент-серверное приложение для проведения тестирования.

Требования к клиенту:

- получение от сервера вопросов с вариантами ответов и отправка на сервер выбранного варианта ответа;
- получение результатов тестирования;
- удобный графический интерфейс.

Требования к серверу:

- хранение 5 вопросов с вариантами ответов и правильным ответом;
- отправка клиенту вопросов в случайном порядке;
- обработка результатов тестирования и отправка заключения клиенту.

### **Вариант 5**

Разработать клиент-серверное приложение для обработки текста.

Требования к клиенту:

- отправка на сервер введенного пользователем текста;
- получение обработанного сервером текста;
- удобный графический интерфейс.

Требования к серверу:

- обработка полученного текста по следующим правилам: перед запятой не может быть пробела; после запятой должен стоять один пробел; не могут стоять подряд две запятые; текст не может начинаться с запятой.

### **Вариант 6**

Разработать клиент-серверное приложение для вычисления выражений.

Требования к клиенту:

- отправка на сервер введенного пользователем выражения (числа в двоичном представлении, знаки «+» и «-»), например: «1111+10-11+111»;
- получение результата вычисления выражения;
- удобный графический интерфейс.

Требования к серверу:

- вычисление полученного от клиента выражения и отправка результата клиенту.

### **Вариант 7**

Разработать клиент-серверное приложение для вычисления выражений.

Требования к клиенту:

- отправка на сервер введенного пользователем выражения (числа в восьмеричном представлении, знаки «+» и «-»), например: «33+7-10+2»;
- получение результата вычисления выражения;
- удобный графический интерфейс.

Требования к серверу:

- вычисление полученного от клиента выражения и отправка результата клиенту.

### **Вариант 8**

Разработать клиент-серверное приложение для вычисления выражений.

Требования к клиенту:

- отправка на сервер введенного пользователем выражения (числа в шестнадцатеричном представлении, знаки «+» и «-»), например: «F11+E8-8+10»;
- получение результата вычисления выражения;
- удобный графический интерфейс.

Требования к серверу:

- вычисление полученного от клиента выражения и отправка результата клиенту.

### **Вариант 9**

Разработать клиент-серверное приложение для обработки текста.

Требования к клиенту:

- отправка на сервер введенного пользователем текста;
- получение обработанного сервером текста;
- удобный графический интерфейс.

Требования к серверу:

- обработка полученного текста по следующим правилам: перед дефисом и после дефиса должны стоять по одному пробелу; не могут стоять подряд два дефиса; предложение не может начинаться с дефиса.

### **Вариант 10**

Разработать клиент-серверное приложение шифрования и дешифрования текста.

Требования к клиенту:

- отправка на сервер введенного пользователем текста с указанием требуемого действия (шифрования или дешифрования);
- получение обработанного сервером текста;
- удобный графический интерфейс.

Требования к серверу:

- для шифрования и дешифрования полученного текста использовать азбуку Морзе.

### **Вариант 11**

Разработать клиент-серверное приложение шифрования и дешифрования текста.

Требования к клиенту:

- отправка на сервер введенного пользователем текста с указанием требуемого действия (шифрования или дешифрования);
- получение обработанного сервером текста;
- удобный графический интерфейс.

Требования к серверу:

- для шифрования и дешифрования полученного текста использовать шифр Цезаря.

### **Вариант 12**

Разработать клиент-серверное приложение шифрования и дешифрования текста.

Требования к клиенту:

- отправка на сервер введенного пользователем текста с указанием требуемого действия (шифрования или дешифрования);
- получение обработанного сервером текста;
- удобный графический интерфейс.

Требования к серверу:

- для шифрования и дешифрования полученного текста использовать транспозирующий шифр (каждое слово писать задом наперед, например «Hello, world!» зашифруется как «olleH, dlrow!»).

### **Вариант 13**

Разработать клиент-серверное приложение шифрования и дешифрования текста.

Требования к клиенту:

- отправка на сервер введенного пользователем текста с указанием требуемого действия (шифрования или дешифрования);
- получение обработанного сервером текста;
- удобный графический интерфейс.

Требования к серверу:

- для шифрования и дешифрования полученного текста использовать транспозирующий шифр (менять местами каждые две буквы, например «Hello, 12345 world!» зашифруется как «eHllo, 12345 owlrd!»).

### **Вариант 14**

Разработать клиент-серверное приложение для обработки текста.

Требования к клиенту:

- отправка на сервер введенного пользователем текста;
- получение от сервера результатов обработки текста;
- удобный графический интерфейс.

Требования к серверу:

- для полученного текста посчитать сколько раз встречается каждая буква в тексте и отправить результаты анализа клиенту.

**Вариант 15**

Разработать клиент-серверное приложение шифрования и дешифрования текста.

Требования к клиенту:

- отправка на сервер введенного пользователем текста с указанием требуемого действия (шифрования или дешифрования);
- получение обработанного сервером текста;
- удобный графический интерфейс.

Требования к серверу:

- для шифрование и дешифрование полученного текста использовать квадрат Полибия.

**Вариант 16**

Разработать клиент-серверное приложение для вычисления выражений.

Требования к клиенту:

- отправка на сервер введенного пользователем выражения (числа в десятичном представлении, знаки «+», «-» и «\*»), например: «33+7\*4-10+2\*1»;
- получение результата вычисления выражения;
- удобный графический интерфейс.

Требования к серверу:

- вычисление полученного от клиента выражения и отправка результата клиенту.

**Вариант 17**

Разработать клиент-серверное приложение (чат).

Требования к клиенту:

- отправка на сервер введенного пользователем сообщения;
- получение сообщений, отправленных другими клиентами;
- удобный графический интерфейс.

Требования к серверу:

- организация чата между клиентами (до 3 клиентов), а именно отправка всем клиентам полученных сообщений;
- полученные сообщения перед отправкой клиентам необходимо проверить на грамотность и исправить: после пробельного символа должен идти непробельный символ (например, если указаны подряд 3 пробела, то необходимо оставить только один пробел, а остальные удалить).

Клиенты должны последовательно отправлять сообщения.

**Вариант 18**

Разработать клиент-серверное приложение (чат).

Требования к клиенту:

- отправка на сервер введенного пользователем сообщения;
- получение сообщений, отправленных другими клиентами;
- удобный графический интерфейс.

Требования к серверу:

- организация чата между клиентами (до 3 клиентов), а именно отправка всем клиентам полученных сообщений;
- полученные сообщения перед отправкой клиентам необходимо проверить на грамотность и исправить: после запятой должен стоять один пробел, далее должен идти непробельный символ.

Клиенты должны последовательно отправлять сообщения.

**Вариант 19**

Разработать клиент-серверное приложение (чат).

Требования к клиенту:

- отправка на сервер введенного пользователем сообщения;
- получение сообщений, отправленных другими клиентами;
- удобный графический интерфейс.

Требования к серверу:

- организация чата между клиентами (до 3 клиентов), а именно отправка всем клиентам полученных сообщений;
  - полученные сообщения перед отправкой клиентам необходимо проверить на грамотность и исправить: не может идти подряд два знака препинания.
- Клиенты должны последовательно отправлять сообщения.

**Вариант 20**

Разработать клиент-серверное приложение для вычисления выражений.

Требования к клиенту:

- отправка на сервер введенного пользователем выражения (числа в шестнадцатеричном представлении, знаки «+», «-» и скобки «(», «)»), например: «AF3+(14-1B+1)-2+(4+FF+(3-E))»;
- получение результата вычисления выражения;
- удобный графический интерфейс.

Требования к серверу:

- вычисление полученного от клиента выражения и отправка результата клиенту.

**Вариант 21**

Разработать клиент-серверное приложение для вычисления выражений.

Требования к клиенту:

- отправка на сервер введенного пользователем выражения (числа в десятичном представлении, знаки «+», «-» и «/»), например: «33+15/4-10+8/2-2»;
- получение результата вычисления выражения;
- удобный графический интерфейс.

Требования к серверу:

- вычисление полученного от клиента выражения и отправка результата клиенту.

**Вариант 22**

Разработать клиент-серверное приложение для обработки текста.

Требования к клиенту:

- отправка на сервер введенного пользователем текста;
- получение от сервера списка слов;
- удобный графический интерфейс.

Требования к серверу:

- сортировка слов в полученном тексте по алфавиту, результаты сортировки представить в виде списка слов;
- отправка списка слов клиенту.

### **Лабораторная работа № 7** **Разработка приложения, взаимодействующего с базой данных**

Разработать приложение (можно web-приложение), взаимодействующее с базой данных. Приложение должно иметь удобный графический интерфейс. Базу данных разработать в соответствии с темой своего варианта (варианты представлены ниже). База данных должна состоять из 2-3 связанных таблиц; одна таблица основная.

Функционал приложения:

- добавление информации в основную таблицу;
- удаление информации из основной таблицы;
- отображение информации из основной таблицы.

Добавление и отображение информации должно быть реализовано в читаемой для пользователя форме (внешние ключи не отображать, вместо них отображать пользователю понятную информацию).

Варианты:

1. База данных «Сотрудники».
2. База данных «Учителя».
3. База данных «Ученики».
4. База данных «Книги».
5. База данных «Преподаватели».
6. База данных «Товары».
7. База данных «Лекарства».
8. База данных «Студенты».
9. База данных «Участники олимпиады».
10. База данных «Фильмы».
11. База данных «Ресторан».
12. База данных «Автомобили».
13. База данных «Страны».
14. База данных «Животные».
15. База данных «Врачи поликлиники».
16. База данных «Города».
17. База данных «Пациенты».
18. База данных «Процессоры».
19. База данных «Микроконтроллеры».
20. База данных «Дисциплины».
21. База данных «Университеты».
22. База данных «Абитуриенты».