

Дата: 2020/13/14
Предмет: Информатика
Тема: Элементы программирования
Тип занятия: Лабораторная работа
Группа: ТК-20

Порядок выполнения работы: решить задачи в среде программирования Thonny, сохранить в архив, выложить в личный кабинет.

Критерии оценки выполнения задания: работа считается зачтённой, если задания выполнены правильно.

Порядок распределения заданий между студентами:
студенты выполняют задания согласно распределению вариантов.

- 1 Батомункуев Баир Тумэнович
- 2 Богомоллов Федор Олегович
- 3 Бронникова Кристина Романовна
- 4 Виткаускас Павел Анатольевич
- 5 Власов Даниил Евгеньевич
- 6 Гурулева Дарья Викторовна
- 7 Дамдинжапов Тимур Владимирович
- 8 Дашидондоков Алдар Борисович
- 9 Доржиева Галина Шимитовна
- 10 Зверев Павел Васильевич
- 1 Иринчинова Валентина Баясхалановна
- 2 Концов Никита Витальевич
- 3 Курицын Сергей Викторович
- 4 Марковцев Константин Игоревич
- 5 Позднякова Александра Витальевна
- 6 Попов Александр Александрович
- 7 Попов Михаил Владимирович
- 8 Потапов Игорь Андреевич
- 9 Рыбаков Алексей Сергеевич
- 10 Сидаш Алина Андреевна
- 1 Стукалов Дмитрий Игоревич
- 2 Стуков Антон Валерьевич
- 3 Тарханов Денис Алексеевич
- 4 Тарханова Любовь Сергеевна
- 5 Толстоногов Виктор Викторович

- 6 Толстоногова Любовь Сергеевна
7 Тунгаланов Базар Очирович
8 Цыпылов Иван Владимирович
9 Цэдашиева Жаргалма Амгалановна
10 Щербатюк Илья Максимович
1 Яковлев Николай Викторович

1. Теоретический материал

1.1. Списки

1.1.1. Задание списков

Список представляет собой упорядоченную последовательность разнотипных элементов, которые, в свою очередь, также могут быть списками. Одномерные списки могут быть использованы в качестве представления векторов, двумерные – матриц.

Обратиться к i -му элементу списка a можно по номеру: $a[i]$. Нумерация элементов начинается с нуля. Если список включает в себя другие списки (вложенные списки), то индексы элементов перечисляются в квадратных скобках: $a[i][j]$.

Для определения длины списка a применяется функция $len(a)$.

Как уже было показано выше, список можно задать с помощью присваивания. Также можно ввести элементы списка с клавиатуры, используя при этом цикл. К примеру, зададим трёхэлементный список вещественных чисел.

Листинг 1. Ввод элементов списка с клавиатуры

```
myList = [ ]
for i in range(3):
    myList.append(float(input(f"Введи {i}-й элемент списка: ")))
print(myList)
```

Результат:

```
Введи 0-й элемент списка: 2
Введи 1-й элемент списка: 5
Введи 2-й элемент списка: 4
[2.0, 5.0, 4.0]
```

Одномерный список можно внести с клавиатуры целиком, ставя между элементами списка символ *разделитель*. Идея такова: после ввода строки с клавиатуры,

необходимо её разрезать на элементы, используя метод строки `split(разделитель)`. После этого мы получим список строк. Если требуется получить не список строк, а, например, список чисел, то каждый элемент этого списка нужно преобразовать с помощью функций преобразования типов `int` или `float`. Распределить эти функции по списку (то есть, применить их к каждому элементу списка как к аргументу) нужно с помощью функции `map(функция, список)`. Результат, возвращаемый функцией `map`, имеет тип `<class 'map'>`, поэтому к нему придётся применить функцию преобразования в список `list`.

Листинг 2. Ввод списка строкой

```
myString = input('Введите числа через запятые: ')
myList = myString.split(',')
print(list(map(float,myList)))
```

Результат:

```
Введите числа через запятые: 1, 2.3, 5.66
[1.0, 2.3, 5.66]
```

Наконец, список можно получить с помощью функции `list` из строки, кортежа, словаря или множества.

Листинг 3. Получение списка цифр числа и преобразованием словаря

```
a = list(str(12346))
print(a)
b = list({"x": 1, "y": 2})
print(b)
```

Результат:

```
['1', '2', '3', '4', '6']
['x', 'y']
```

1.1.2. Срезы

Можно извлекать элементы из списка не по отдельности, но группой, используя срезы вида `a[i:j:n]`, где `a` – список, `i` – номер начального элемента среза, `j` – номер конечного элемента среза, `n` – шаг, с которым выбираются элементы списка. Заметим, что элемент с номером `j` на экран не выводится. Кроме того, можно

использовать отрицательные номера элементов: последний элемент списка имеет номер -1 , предпоследний -2 и т. д.

Листинг 4. Срезы

```
myList = ["a", "b", "c", "d", "e"]
print("myList[1:3] =",myList[1:3])
print("myList[1:] =",myList[1:])
print("myList[:3] =",myList[:3])
print("myList[-2:] =",myList[-2:])
print("myList[:-2] =",myList[:-2])
print("myList[:] =",myList[:])
print("myList[:2] =",myList[:2])
print("myList[4:2] =",myList[4:2])
```

Результат:

```
myList[1:3] = ['b', 'c']
myList[1:] = ['b', 'c', 'd', 'e']
a myList[:3] = ['a', 'b', 'c']
myList[-2:] = ['d', 'e']
myList[:-2] = ['a', 'b', 'c']
myList[:] = ['a', 'b', 'c', 'd', 'e']
myList[:2] = ['a', 'c', 'e']
myList[4:2] = ['a', 'c']
```

При извлечении среза получается новый список со своим идентификатором. Данный эффект полезен, если необходимо сделать копию списка, независимую от своего оригинала. К сожалению, элементы списков будут иметь одинаковые идентификаторы.

Листинг 5. Результат получения среза

```
a = [1, 2, 3]
b = a[:]
print(id(a), id(a[1]))
print(id(b), id(b[1]))
```

Результат:

```
45787776 1945068704
48301960 1945068704
```

1.1.3. Lambda-функции

Если тело функции можно записать в одну-две строчки кода, либо вызов функции производится в единственном месте основного кода, то можно не задавать такую функцию с помощью `def`, а определить в виде lambda-функции:

Определение 1. *Синтаксис lambda-функции*

`lambda последовательность_аргументов : выражение`

Функция вычисляет некоторое *выражение* и возвращает его значение в качестве результата. Чтобы применить lambda-функцию к конкретным аргументам, её определение окружают круглыми скобками и рядом в круглых скобках записывают эти аргументы.

Покажем, как можно написать функцию, возвращающую `True`, если целое число является точным квадратом, и `False`, если целое число точным квадратом не является.

Листинг 6. *Пример использования lambda-функции*

```
from math import sqrt, floor
print((lambda n : floor(sqrt(n)) == sqrt(n))(100))
```

Результат:

True

1.1.4. Полезные функции, применяемые к спискам

Функция `map(f, a)`, имея аргументами заголовок `f` функции одного аргумента `f(x)` и список `a`, имеющий вид `[a[0], a[1], ..., a[n]]`, возвращает выражение вида `[f(a[0]), f(a[1]), ..., f(a[n])]`. Другими словами, она распределяет некоторую функцию `f` по элементам списка, делая их аргументом функции, или *векторизует* функцию `f`. Напомним, что результат, возвращаемый функцией `map`, имеет тип `<class 'map'>`, поэтому к нему придётся применить функцию преобразования в список `list`.

Заметим, что удобно в качестве функции `f` брать lambda-функцию.

Приведём пример, в котором покажем, как можно проверить, являются ли элементы целочисленного списка точными квадратами.

Листинг 7. *Проверка элементов списка*

```
from math import sqrt, floor
myList = [1, 4, 5, 9, 10, 12]
myResult = list(map(lambda n : floor(sqrt(n)) == sqrt(n),myList))
print(myResult)
```

Результат:

```
[True, True, False, True, False, False]
```

Функция `all(a)`, применённая к списку `a`, возвращает `True`, если все элементы списка имеют значение `True`, либо не являются нулями (могут быть ненулевыми числами, непустыми строками, непустыми списками и т. п.). Так, можно изменить результат, выводимый в предыдущем примере, если нам нужно узнать, *все* ли элементы списка являются точными квадратами:

Листинг 8. Проверка элементов списка

```
from math import sqrt, floor
myList = [1, 4, 5, 9, 10, 12]
myResult = all(list(map(lambda n : floor(sqrt(n)) == sqrt(n),myList)))
print(myResult)
```

Результат:

```
False
```

Приведём ещё несколько полезных функций и методов списков.

Функция	Описание
<code>min(a)</code>	Вычисляет минимальный элемент последовательности a
<code>max(a)</code>	Вычисляет максимальный элемент последовательности a
<code>sorted(a)</code>	Сортирует последовательность a
<code>a.sort()</code>	Сортирует список a
<code>a.append(e)</code>	Добавляет в конец списка a элемент e
<code>a.insert(n,e)</code>	Добавляет в список a элемент e в позицию с номером n
<code>a.remove(e)</code>	Удаляет из списка a первое вхождение элемента e
<code>a.pop(n)</code>	Удаляет из списка a элемент в позиции с номером n
<code>a.count(e)</code>	Возвращает количество вхождений элемента e в список a

Заметим, что функция `sorted` возвращает новый список, а метод `list.sort` меняет исходный список. Кроме того, три первые функции применимы не только к спискам, но к любым последовательностям (кортежи, словари), а первые две функции применимы и к последовательностям отдельных значений. Так, верны записи `min(1, 2)` и `min([1, 2])`. Функция `count` также применима к кортежам и строкам.

Приведём пример, в котором отсортируем список, содержащий данные вида `["имя", рост]` по имени, а затем по убыванию роста. По умолчанию сортировка осуществляется по первому элементу каждой «строки» списка.

Листинг 9. Сортировка

```
myList = ["Маша", 167], ["Ваня", 172], ["Петя", 181]]
print(sorted(myList))
myList.sort()
print(myList)
```

```
myList = ["Маша", 167], ["Ваня", 172], ["Петя", 181]]
print(sorted(myList, key = lambda elem : elem[0]))
myList.sort(key = lambda elem : elem[0])
print(myList)
```

```
myList = ["Маша", 167], ["Ваня", 172], ["Петя", 181]]
print(sorted(myList, key = lambda elem : elem[1], reverse = True))
myList.sort(key = lambda elem : elem[1], reverse = True)
print(myList)
```

Результат:

```
[['Ваня', 172], ['Маша', 167], ['Петя', 181]]
[['Ваня', 172], ['Маша', 167], ['Петя', 181]]
[['Ваня', 172], ['Маша', 167], ['Петя', 181]]
[['Ваня', 172], ['Маша', 167], ['Петя', 181]]
[['Петя', 181], ['Ваня', 172], ['Маша', 167]]
[['Петя', 181], ['Ваня', 172], ['Маша', 167]]
```

1.1.5. Генератор списков (списочное включение)

В Python имеется конструкция, называемая *генератором списка* или *списочным включением*, которая позволяет сделать код короче при автоматическом создании списка (в Python есть отдельное понятие *генератора*; термины «генератор» и «генератор списка» не стоит смешивать).

Определение 2. *Синтаксис генератора линейного списка*

[*выражение* for элемент in последовательность if условие]

Элементы списка создаются по формуле, записанной в *выражении*, пока цикл перебирает элементы из некоторой *последовательности*, которая может быть объектом `range`, списком, множеством, кортежем или словарём, и для каждого элемента проверяется *условие*, только при выполнении которого результат *выражения* добавляется в список.

Чтобы создать многомерный список, нужно в качестве *выражения* использовать генератор списка.

Листинг 10. *Примеры применения списочных включений*

```
print([n for n in range(5)])
print([i + j for j in range(2) for i in range(3)])
print([[i + j for j in range(2)] for i in range(3)])
a = [[1, 2, 3], [4, 5, 6]]
print([[elem[n]**2 for n in range(len(elem))] for elem in a])
```

Результат:

```
[0, 1, 2, 3, 4]
[0, 1, 2, 1, 2, 3]
[[0, 1], [1, 2], [2, 3]]
[[1, 4, 9], [16, 25, 36]]
```

Решим задачу: создать список случайных целых чисел и вывести все номера позиций минимальных элементов списка.

Листинг 11. Решение задачи с помощью списочного включения

```
from random import randint
a = [randint(-2,3) for i in range(10)]
b = [i for i in range(10) if a[i] == min(a)]
print(f"Исходный список: {a}.\nПозиции минимальных элементов: {b}.")
```

Результат:

Исходный список: [-2, 2, -2, -1, -2, 2, -1, 2, 3, -2].
Позиции минимальных элементов: [0, 2, 4, 9].

1.1.6. Задания

Вариант 1.

1.1. Создайте массив A длины 10. Заполните его с клавиатуры целыми числами. Выведите его на экран. Найдите минимум среди всех элементов массива. Возведите все чётные элементы массива в квадрат. Выведите новый массив на экран.

1.2. Создайте массивы A и B , состоящие из 10 вещественных случайных чисел, взятых из единичного отрезка. Выведите их на экран. Создайте массив C , состоящий из 10 вещественных чисел, который составлен по правилу:

$$c_i = \min\{a_i, b_i\}.$$

Выведите массив C на экран.

1.3. Создайте массив A длины 10. Заполните его целыми случайными числами из промежутка $[-20; 45]$. Выведите его на экран. Вычислите среднее арифметическое элементов этого массива. Формула среднего арифметического:

$$s_1 = \frac{a_1 + a_2 + \dots + a_n}{n}$$

1.4. Создайте две матрицы A и B типа 2×3 , заполните их с клавиатуры целыми числами, выведите их на экран. Вычислите матрицу $C = 5A - 2B$. Выведите её на экран. Выведите на экран сумму элементов второго столбца матрицы C .

Вариант 2.

2.1. Создайте массив A длины 10. Заполните его с клавиатуры целыми числами. Выведите его на экран. Найдите максимум среди всех элементов массива. Возве-

дите все нечётные элементы массива в третью степень. Выведите новый массив на экран.

2.2. Создайте массивы A и B , состоящие из 10 вещественных случайных чисел, взятых из единичного отрезка. Выведите их на экран. Создайте массив C , состоящий из 10 вещественных чисел, который составлен по правилу:

$$c_i = \max\{a_i, b_i\}.$$

Выведите массив C на экран.

2.3. Создайте массив A длины 20. Заполните его целыми случайными числами из промежутка $[5; 25]$. Выведите его на экран. Вычислите среднее геометрическое элементов этого массива. Формула среднего геометрического:

$$s_0 = \sqrt[n]{a_1 \cdot a_2 \cdot \dots \cdot a_n}$$

2.4. Создайте две матрицы A и B типа 3×2 , заполните их с клавиатуры целыми числами, выведите их на экран. Вычислите матрицу $C = 2A - 7B$. Выведите её на экран. Выведите на экран сумму элементов второй строки матрицы C .

Вариант 3.

3.1. Создайте массив A длины 10. Заполните его с клавиатуры целыми числами. Выведите его на экран. Найдите минимум среди всех элементов массива. Замените все чётные элементы массива их половинами. Выведите новый массив на экран.

3.2. Создайте массивы A и B , состоящие из 10 вещественных случайных чисел, взятых из единичного отрезка. Выведите их на экран. Создайте массив C , состоящий из 10 вещественных чисел, который составлен по правилу:

$$c_i = a_i \cdot b_i.$$

Выведите массив C на экран.

3.3. Создайте массив A длины 15. Заполните его целыми случайными числами из промежутка $[10; 30]$. Выведите его на экран. Вычислите среднее квадратическое элементов этого массива. Формула среднего квадратического:

$$s_2 = \sqrt{\frac{a_1^2 + a_2^2 + \dots + a_n^2}{n}}$$

3.4. Создайте две матрицы A и B типа 2×4 , заполните их с клавиатуры целыми числами, выведите их на экран. Вычислите матрицу $C = -5A + 3B$. Выведите её на экран. Выведите на экран произведение элементов третьего матрицы C .

Вариант 4.

4.1. Создайте массив A длины 10. Заполните его с клавиатуры целыми числами.

Выведите его на экран. Найдите максимум среди всех элементов массива. Вычтите из всех нечётных элементов массива единицу и поделите полученные числа на два. Выведите новый массив на экран.

4.2. Создайте массивы A и B , состоящие из 10 вещественных случайных чисел, взятых из единичного отрезка. Выведите их на экран. Создайте массив C , состоящий из 10 вещественных чисел, который составлен по правилу:

$$c_i = a_i + b_i - a_i \cdot b_i.$$

Выведите массив C на экран.

4.3. Создайте массив A длины 10. Заполните его целыми случайными числами из промежутка $[-5; 20]$. Выведите его на экран. Вычислите среднее кубическое элементов этого массива. Формула среднего кубического:

$$s_3 = \sqrt[3]{\frac{a_1^3 + a_2^3 + \dots + a_n^3}{n}}$$

4.4. Создайте две матрицы A и B типа 4×2 , заполните их с клавиатуры целыми числами, выведите их на экран. Вычислите матрицу $C = -9A - 2B$. Выведите её на экран. Выведите на экран произведение элементов первой строки матрицы.

Вариант 5.

5.1. Создайте массив A длины 10. Заполните его с клавиатуры целыми числами. Выведите его на экран. Найдите минимум среди всех элементов массива. Замените элементы массива, стоящие на нечётных местах, их квадратами. Выведите новый массив на экран.

5.2. Создайте массивы A и B , состоящие из 10 вещественных случайных чисел, взятых из единичного отрезка. Выведите их на экран. Создайте массив C , состоящий из 10 вещественных чисел, который составлен по правилу:

$$c_i = |a_i - b_i|.$$

Выведите массив C на экран.

5.3. Создайте массив A длины 15. Заполните его целыми случайными числами из промежутка $[-10; 10]$. Выведите его на экран. Вычислите среднее степенное четвёртого порядка элементов этого массива. Формула среднего степенного четвёртого порядка:

$$s_4 = \sqrt[4]{\frac{a_1^4 + a_2^4 + \dots + a_n^4}{n}}$$

5.4. Создайте две матрицы A и B типа 4×3 , заполните их с клавиатуры целыми числами, выведите их на экран. Вычислите матрицу $C = 5A - 6B$. Выведите её на экран. Выведите на экран минимальный элемент третьей строки матрицы C .

Вариант 6.

6.1. Создайте массив A длины 10. Заполните его с клавиатуры целыми числами. Выведите его на экран. Найдите максимум среди всех элементов массива. Замените элементы массива, стоящие на чётных местах, их кубами. Выведите новый массив на экран.

6.2. Создайте массивы A и B , состоящие из 10 вещественных случайных чисел, взятых из единичного отрезка. Выведите их на экран. Создайте массив C , состоящий из 10 вещественных чисел, который составлен по правилу:

$$c_i = \sqrt{a_i^2 + b_i^2}.$$

Выведите массив C на экран.

6.3. Создайте массив A длины 15. Заполните его целыми случайными числами из промежутка $[10; 20]$. Выведите его на экран. Вычислите среднее гармоническое элементов этого массива. Формула среднего гармонического:

$$s_{-1} = \frac{n}{\frac{1}{a_1} + \frac{1}{a_2} + \dots + \frac{1}{a_n}}$$

6.4. Создайте две матрицы A и B типа 3×4 , заполните их с клавиатуры целыми числами, выведите их на экран. Вычислите матрицу $C = 6A - 3B$. Выведите её на экран. Выведите на экран минимальный элемент четвёртого столбца матрицы C .

Вариант 7.

7.1. Создайте массив A длины 10. Заполните его с клавиатуры целыми числами. Выведите его на экран. Найдите минимум среди всех элементов массива. Замените все элементы массива, кроме первого и последнего, их противоположными (число, противоположное x , равно $-x$). Выведите новый массив на экран.

7.2. Создайте массивы A и B , состоящие из 10 вещественных случайных чисел, взятых из единичного отрезка. Выведите их на экран. Создайте массив C , состоящий из 10 вещественных чисел, который составлен по правилу:

$$c_i = \frac{a_i + b_i}{2}.$$

Выведите массив C на экран.

7.3. Создайте массив A длины 10. Заполните его целыми случайными числами из промежутка $[1; 5]$. Выведите его на экран. Вычислите среднее экспоненциальное элементов этого массива. Формула среднего экспоненциального:

$$s_e = \ln \left(\frac{e^{a_1} + e^{a_2} + \dots + e^{a_n}}{n} \right)$$

7.4. Создайте две матрицы A и B типа 2×5 , заполните их с клавиатуры целыми числами, выведите их на экран. Вычислите матрицу $C = -4A - 5B$. Выведите её на экран. Выведите на экран максимальный элемент четвёртого столбца матрицы C .

Вариант 8.

8.1. Создайте массив A длины 10. Заполните его с клавиатуры целыми числами. Выведите его на экран. Найдите максимум среди всех элементов массива. Замените все отрицательные элементы массива их абсолютными величинами (модулями). Выведите новый массив на экран.

8.2. Создайте массивы A и B , состоящие из 10 вещественных случайных чисел, взятых из единичного отрезка. Выведите их на экран. Создайте массив C , состоящий из 10 вещественных чисел, который составлен по правилу:

$$c_i = \sqrt{a_i \cdot b_i}.$$

Выведите массив C на экран.

8.3. Создайте массив A длины 15. Заполните его целыми случайными числами из промежутка $[10; 20]$. Выведите его на экран. Вычислите среднее логарифмическое элементов этого массива. Формула среднего логарифмического:

$$s_e = e^{\frac{\ln a_1 + \ln a_2 + \dots + \ln a_n}{n}}$$

8.4. Создайте две матрицы A и B типа 5×2 , заполните их с клавиатуры целыми числами, выведите их на экран. Вычислите матрицу $C = -5A + 8B$. Выведите её на экран. Выведите на экран максимальный элемент второй строки матрицы C .

Вариант 9.

9.1. Создайте массив A длины 10. Заполните его с клавиатуры целыми числами. Выведите его на экран. Найдите минимум среди всех элементов массива. Прибавьте к каждому элементу массива его номер. Выведите новый массив на экран.

9.2. Создайте массивы A и B , состоящие из 10 вещественных случайных чисел, взятых из единичного отрезка. Выведите их на экран. Создайте массив C , состоящий из 10 вещественных чисел, который составлен по правилу:

$$c_i = \sin(a_i) \cdot \cos(b_i).$$

Выведите массив C на экран.

9.3. Создайте массив A длины 15. Заполните его целыми случайными числами из промежутка $[-10; 10]$. Выведите его на экран. Вычислите среднее тангенциальное

элементов этого массива. Формула среднего тангенциального:

$$s_{\text{tg}} = \text{arctg} \left(\frac{\text{tg}(a_1) + \text{tg}(a_2) + \dots + \text{tg}(a_n)}{n} \right)$$

9.4. Создайте две матрицы A и B типа 3×3 , заполните их с клавиатуры целыми числами, выведите их на экран. Вычислите матрицу $C = 8A - 5B$. Выведите её на экран. Выведите на экран среднее арифметическое элементов второго столбца матрицы C .

Вариант 10.

10.1. Создайте массив A длины 10. Заполните его с клавиатуры целыми числами. Выведите его на экран. Найдите максимум среди всех элементов массива. Вычтите из каждого элемента массива его номер. Выведите новый массив на экран.

10.2. Создайте массивы A и B , состоящие из 10 вещественных случайных чисел, взятых из единичного отрезка. Выведите их на экран. Создайте массив C , состоящий из 10 вещественных чисел, который составлен по правилу:

$$c_i = \text{tg}(a_i) \cdot \text{ctg}(b_i).$$

Выведите массив C на экран.

10.3. Создайте массив A длины 15. Заполните его целыми случайными числами из промежутка $[-10; 10]$. Выведите его на экран. Вычислите среднее синусоидальное элементов этого массива. Формула среднего синусоидального:

$$s_{\text{sin}} = \arcsin \left(\frac{\sin(a_1) + \sin(a_2) + \dots + \sin(a_n)}{n} \right)$$

10.4. Создайте две матрицы A и B типа 4×4 , заполните их с клавиатуры целыми числами, выведите их на экран. Вычислите матрицу $C = 5A - 9B$. Выведите её на экран. Выведите на экран среднее арифметическое элементов третьей строки матрицы C .

Дополнительные задачи

1. Создайте матрицу A типа 3×4 и матрицу B типа 4×5 , заполните их случайными целыми числами, выведите их на экран. Вычислите матрицу $C = A \cdot B$. Выведите её на экран.
2. Создайте массив длины 25, состоящий из случайных целых чисел, взятых из промежутка $[0; 100]$. Выведите на экран его различные элементы. Подсчитайте количество повторений его элементов.
3. Создайте массив длины 25, состоящий из случайных целых чисел, взятых из промежутка $[0; 100]$. Упорядочьте его по возрастанию или убыванию.