Лабораторная работа №1

Задания выполняются по вариантам. Вариант выбирается по порядковому номеру из списка группы:

|  |  |  |  |
| --- | --- | --- | --- |
| Номер варианта | Номера заданий | Номер варианта | Номера заданий |
| 1 | 1, 6, 11, 16, 21 | 14 | 5, 10, 15, 20, 25 |
| 2 | 2, 7, 12, 17, 22 | 15 | 1, 7, 14, 18, 22 |
| 3 | 3, 8, 13, 18, 23 | 16 | 2, 9, 11, 16, 25 |
| 4 | 4, 9, 14, 19, 24 | 17 | 3, 6, 12, 19, 21 |
| 5 | 5, 10, 15, 20, 25 | 18 | 4, 7, 13, 20, 23 |
| 6 | 1, 7, 14, 18, 22 | 19 | 5, 8, 12, 17, 24 |
| 7 | 2, 9, 11, 16, 25 | 20 | 3, 8, 13, 18, 23 |
| 8 | 3, 6, 12, 19, 21 | 21 | 2, 9, 11, 16, 25 |
| 9 | 4, 7, 13, 20, 23 | 22 | 3, 6, 12, 19, 21 |
| 10 | 5, 8, 12, 17, 24 | 23 | 1, 6, 11, 16, 21 |
| 11 | 1, 6, 11, 16, 21 | 24 | 4, 9, 14, 19, 24 |
| 12 | 2, 7, 12, 17, 22 | 25 | 5, 10, 15, 20, 25 |
| 13 | 3, 8, 13, 18, 23 | 26 | 2, 7, 12, 17, 22 |

На языке программирования Java произвести вычисление функций.

1. Вычислить 

где - наибольший элемент массива X(20); - наибольший элемент массива Y(15).

Для вычисления наибольшего элемента массива использовать функцию.

1. Даны массивы действительных чисел А(20), В(15), С(10).

Вычислить  где S, T, K – суммы положительных элементов массивов А, В, С соответственно. Для вычисления суммы положительных элементов использовать функцию.

1. Даны целые числа m, n. Вычислить . Для вычисления факториала использовать функцию.
2. Даны действительные х, у, z. Составить программу вычисления значения



используя функцию для вычисления выражения 

1. Составить программу для вычисления среднего арифметического положительных элементов массивов Х(20), Y( 15), Z( 10), используя в качестве подпрограммы функцию.
2. Даны массивы A(15), B(10),C(12). Вычислить



Для вычислений использовать функцию.

1. Дан массив D(40) вещественных чисел. Найти среднее геометрическое его элементов, которые удовлетворяют условию 0 <di<12. Для вычислений использовать функцию.
2. Дан массив А(80) целых чисел. Найти сумму и количество теx элементов массива, которые отрицательны и нечетны. Использовать в качестве подпрограммы процедуру.
3. Функция, вычисляющая среднее арифметическое элементов массива

Написать функцию, которая вычисляет среднее арифметическое элементов массива, переданного ей в качестве аргумента.

1. Отсортировать массив по возрастанию суммы цифр

Дан одномерный массив, состоящий из натуральных чисел. Выполнить сортировку данного массива по возрастанию суммы цифр чисел. Например, дан массив чисел [14, 30, 103]. После сортировки он будет таким: [30, 103, 14], так как сумма цифр числа 30 составляет 3, числа 103 равна 4, числа 14 равна 5.

1. Вывести на экран исходный массив, отсортированный массив, а также для контроля сумму цифр каждого числа отсортированного массива.
2. Определить количество разрядов числа

Написать функцию, которая определяет количество разрядов введенного целого числа.

1. Сумма ряда с факториалом

Вычислить сумму ряда

5

s = ∑ (-1) \* i \* (x / i!)

i=1

Значение x вводится с клавиатуры.

1. Изменить порядок слов в строке на обратный

Вводится строка, состоящая из слов, разделенных пробелами. Следует заменить ее на строку, в которой слова идут в обратном порядке по сравнению с исходной строкой. Вывести измененную строку на экран.

1. Функция бинарного поиска в массиве

Пользователь вводит число. Сообщить, есть ли оно в массиве, элементы которого расположены по возрастанию значений, а также, если есть, в каком месте находится. При решении задачи использовать бинарный (двоичный) поиск, который оформить в виде отдельной функции.

1. Вычисление наибольших общих делителей

Найти наибольшие общие делители (НОД) для множества пар чисел.

1. Найти площади разных фигур

В зависимости от выбора пользователя вычислить площадь круга, прямоугольника или треугольника. Для вычисления площади каждой фигуры должна быть написана отдельная функция.

1. Найти массив с максимальной суммой элементов

Сгенерировать десять массивов из случайных чисел. Вывести их и сумму их элементов на экран. Найти среди них один с максимальной суммой элементов. Указать какой он по счету, повторно вывести этот массив и сумму его элементов.

Заполнение массива и подсчет суммы его элементов оформить в виде отдельной функции.

1. Вычислить сумму элементов главной или побочной диагонали матрицы

Дана квадратная матрица. Вычислить сумму элементов главной или побочной диагонали в зависимости от выбора пользователя. Сумма элементов любой диагонали должна вычисляться в одной и той же функции.

1. Функция перевода десятичного числа в двоичное

Переводить в двоичную систему счисления вводимые в десятичной системе счисления числа до тех пор, пока не будет введен 0. Для перевода десятичного числа в двоичное написать функцию.

1. Вычислить значения функции y=f(x) на заданном диапазоне

Вычислить значения нижеприведенной функции в диапазоне значений x от -10 до 10 включительно с шагом, равным 1.

y = x2 при -5 <= x <= 5;

y = 2\*|x|-1 при x < -5;

y = 2x при x > 5.

Вычисление значения функции оформить в виде программной функции, которая принимает значение x, а возвращает полученное значение функции (y).

1. Функция заполнения массива случайными числами

Написать функцию, которая заполняет массив случайными числами в диапазоне, указанном пользователем. Функция должна принимать два аргумента - начало диапазона и его конец, при этом ничего не возвращать. Вывод значений элементов массива должен происходить в основной ветке программы.

1. Написать функцию вычисления величины силы тока на участке электрической цепи сопротивлением R Ом при напряжении U В.
2. Написать функцию вычисления напряжения на каждом из последовательно

соединенных участков электрической цепи сопротивлением R1, R2, R3 Ом, если сила

тока при напряжении U В составляет I А.

1. Составить программу для ввода на экран номера дня недели и вывода соответствующего ему дня недели на русском языке.

Создание функций на языке Java

Функция — это небольшая программа, которая выполняет определённые действия при вызове функции по имени.

В языке Java существуют свои встроенные и готовые к использованию функции, но никто не отменял пользовательских функций.

Пользовательская функция — это функция созданная разработчиком для конкретной задачи.

Написание функций относят к процедурному подходу в программировании.

Чаще всего в функции помещают повторяющийся программный код. Для того, чтобы не писать одни и те же строчки кода, их выносят в отдельный блок, а затем вызывают по имени в основном коде программы когда это необходимо.

Различают два основных вида функций:

1. Функция, которая что-то возвращает;
2. Функция, которая ничего не возвращает;

Также функции можно разделить на две категории:

1. Функция без параметров;
2. Функция с параметрами;

*Запомните: переменная созданная в функции, умирает после завершения функции и более не существует!*

Формула создания функции:  
<спецификатор доступа> <указание статичности функции> <тип функции> <имя функции> <(параметры функции)> {тело функции}

Создание функции, которая возвращает

Данный тип функций является наиболее используемым, так как почти всегда требуется что-то посчитать, произвести некое преобразование и тому подобное, следовательно полученный результат необходимо использовать вне функции. Всё что было создано в функции в конечном счёте умирает в ней же, в связи с этим необходимо вернуть результат в основной код программы.

Сделать это помогает оператор return

Создадим функцию, которая вернёт значение переменной и присвоит переменной в основном коде это значение.

Функция, которая возвращает на Java

Java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | public class Main {        public static int func(){  //идентификатор доступа, функция статична, тип возвращаемого значения и имя функции без параметров          int n = 10;  //создание переменной со значением          return n;  //возврат значения при вызове данной функции      }        public static void main(String[] args) {  //блок основного кода программы      }  } |

Создание функции очень похоже на основной блок кода программы. Собственно, это всё.  
Создаётся переменная, а потом она возвращается в нужном разработчику месте.

В теле функции можно прописать любой код, создать массивы, циклы и любая другая структура.

Создание функции, которая ничего не возвращает

Данный тип функции используется при работе с глобальными переменными и если нужно что-то напечатать, вывести на экран.

Нет, почти, никакого отличия между данным типом функции и предыдущим. Необходимо лишь указать другой тип функции и не использовать return.

Создание функции без параметров

Java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | public class Main {        public static void func1(){  //идентификатор доступа, функция статична, тип возвращаемого значения и имя функции без параметров          System.out.println("Привет из функции!");  //вывод на экран сообщения      }        public static void main(String[] args) {  //основной блок программы      }  } |

Вызов функции без параметров

Чтобы работать с функциями и получать от них какой-то результат, необходимо вызвать функцию по имени в нужном месте.

Используем предыдущие функции и вызовем их в основном блоке.

Вызов двух функций на Java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | public class Main {        public static int func(){  //идентификатор доступа, функция статична, тип возвращаемого значения и имя функции без параметров          int n = 10;  //создание переменной со значением          return n;  //возврат значения при вызове данной функции      }        public static void func1(){  //идентификатор доступа, функция статична, тип возвращаемого значения и имя функции без параметров          System.out.println("Привет из функции!");  //вывод на экран сообщения      }        public static void main(String[] args) {  //основной блок программы            int u = func();  //присвоение переменной значения возвращённого функцией          System.out.println(u);  //вывод на экран значения переменной            func1();  //вызов функции по имени      }  } |

В результате получим на экран

Результат от присвоения на Java

|  |  |
| --- | --- |
| 1  2 | 10  Привет из функции! |

Функция, которая что-то возвращает, обязательно должна вызываться таким образом, чтобы возвращаемое значение было чему-то присвоено.

Создание функции с параметрами

Бывает необходимость провести над значениями некие действия. Для этого необходимо передать функции эти самые значения. Когда значения передаются в функцию, то они становятся аргументами функции.

Создадим функцию с параметрами и вызовем её в основном блоке используя аргументы.  
Возведём переменную в определённую степень и вернём значение в переменную.

Указываем параметры при создании функции (переменные в скобках после имени функции). Аргументы должны быть переданы обязательно, иначе функция не заработает, у неё элементарно не будет значения с которым она должна взаимодействовать.

Указываем аргументы при вызове функции (два целочисленных значения).

Создание и вызов функции с параметрами на Java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | public class Main {        public static int func2(int var\_1, int var\_2){  //функция с параметрами          int n = 1;  //создаём переменную, в которую будем помещать расчётное значение          for(int i = 0; i < var\_2; i++){  //цикл для возведения в степень              n = n \* var\_1;  //умножаем новое значение со старым и тем самым возводим в степень          }          return n;  //возвращаем посчитанное значение      }        public static void main(String[] args) {  //основной блок программы          int result = func2(10,2);  //вызываем функцию передав два аргумента (возводим десять во вторую степень)          System.out.println(result);  //вывод значения переменной      }  } |

В результате получаем

Возведение в степень через функцию на Java

|  |  |
| --- | --- |
| 1 | 100 |

В функции вы можете помещать всё что угодно. В данном случае аргументы должны быть переданы обязательно, иначе произойдёт ошибка.

Чтобы ошибки не возникало, можно использовать значения по умолчанию при создании функции, но к сожалению Java обделена такой возможностью, значение умолчанию в функции задать невозможно.