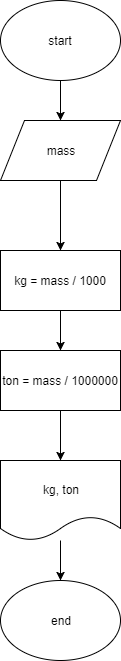
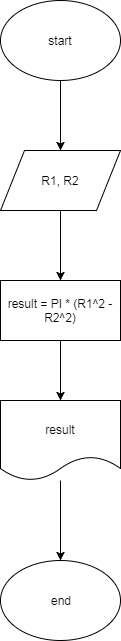
Задание 1.



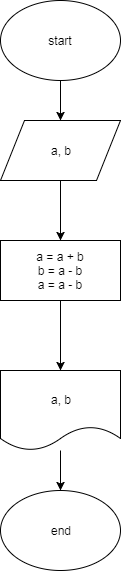
public class Task11 {  
 public static void main(String[] args) {  
 double mass = 150000000;  
 System.*out*.println("Mass in kg: "+*toKilograms*(mass));  
 System.*out*.println("Mass in t: "+*toTons*(mass));  
 }  
  
 public static double toKilograms(double mass){  
 return mass / 1000;  
 }  
  
 public static double toTons(double mass){  
 return mass / 1000000;  
 }  
}

Задание 2.



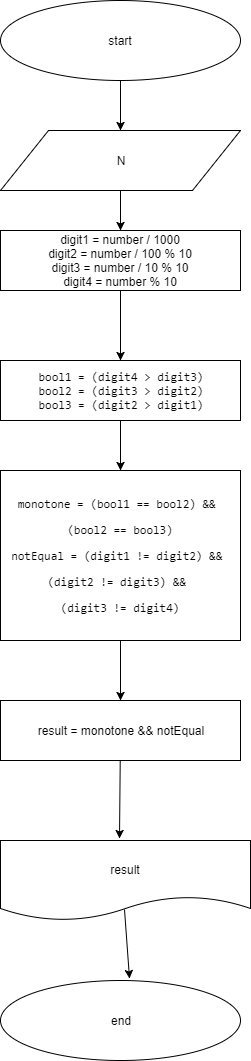
public class Task12 {  
 public static void main(String[] args) {  
 double R1 = 10;  
 double R2 = 3;  
 System.*out*.println(*getRingArea*(R1, R2));  
 }  
  
 public static double getRingArea(double r1, double r2){  
 return Math.*PI* \* (Math.*pow*(r1, 2) - Math.*pow*(r2, 2));  
 }  
  
}

Задание 3.



public class Task13 {  
 public static void main(String[] args) {  
 int a = 5;  
 int b = 1;  
 System.*out*.println("a = " + a + ", b = " + b);  
 *swap*(a, b);  
 }  
  
 public static void swap(int a, int b) {  
 a = a + b;  
 b = a - b;  
 a = a - b;  
 System.*out*.println("a = " + a + ", b = " + b);  
 }  
}

Задание 4.



public class Task14 {  
 public static void main(String[] args) {  
 int N = 4689;  
 System.*out*.println(*check*(N));  
 }  
  
 public static boolean check(int number) {  
 int digit1 = number / 1000;  
 int digit2 = number / 100 % 10;  
 int digit3 = number / 10 % 10;  
 int digit4 = number % 10;  
 boolean bool1 = (digit4 > digit3);  
 boolean bool2 = (digit3 > digit2);  
 boolean bool3 = (digit2 > digit1);  
 boolean monotone = (bool1 == bool2) && (bool2 == bool3);  
 boolean notEqual = (digit1 != digit2) && (digit2 != digit3) && (digit3 != digit4);  
 return monotone && notEqual;  
 }

monotone = (bool1 == bool2) &&

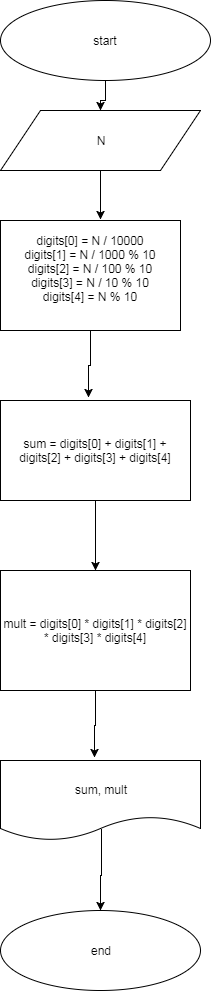
(bool2 == bool3)

notEqual = (digit1 != digit2) &&

(digit2 != digit3) &&

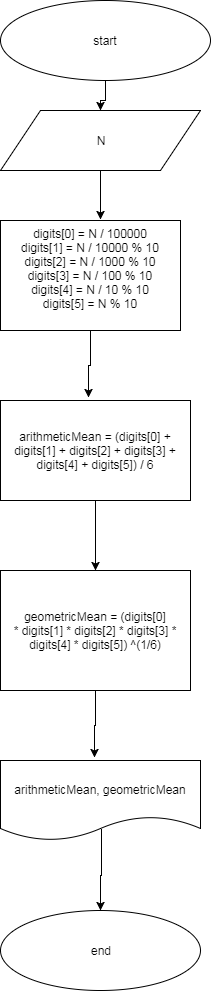
(digit3 != digit4)  
}

Задание 5.



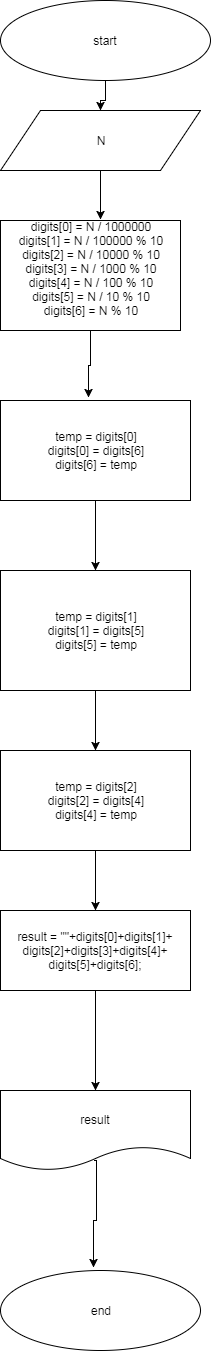
public class Task15 {  
 public static void main(String[] args) {  
 int N = 75946;  
 System.*out*.println("sum: " + *getDigitSum*(N));  
 System.*out*.println("multiplication: " + *getDigitMult*(N));  
 }  
  
 public static int getDigitSum(int n){  
 int digits[] = *getDigits*(n);  
 return digits[0] + digits[1] + digits[2] + digits[3] + digits[4];  
 }  
  
 public static int getDigitMult(int n){  
 int digits[] = *getDigits*(n);  
 return digits[0] \* digits[1] \* digits[2] \* digits[3] \* digits[4];  
 }  
  
 public static int[] getDigits(int n){  
 int digits[] = new int[5];  
 digits[0] = n / 10000;  
 digits[1] = n / 1000 % 10;  
 digits[2] = n / 100 % 10;  
 digits[3] = n / 10 % 10;  
 digits[4] = n % 10;  
 return digits;  
 }  
}

Задание 6.



public class Task16 {  
 public static void main(String[] args) {  
 int N = 286751;  
 System.*out*.println("Arithmetic mean: "+*getMeanArithmetic*(N));  
 System.*out*.println("Geometric mean: "+*getMeanGeometric*(N));  
 }  
  
 public static double getMeanArithmetic(int n){  
 int digits[] = *getDigits*(n);  
 double sum = digits[0] + digits[1] + digits[2] + digits[3] + digits[4] + digits[5];  
 return sum / digits.length;  
 }  
  
 public static double getMeanGeometric(int n){  
 int digits[] = *getDigits*(n);  
 double mult = digits[0] \* digits[1] \* digits[2] \* digits[3] \* digits[4] \* digits[5];  
 return Math.*pow*(mult, 1.0/digits.length);  
 }  
  
 public static int[] getDigits(int n){  
 int digits[] = new int[6];  
 digits[0] = n / 100000;  
 digits[1] = n / 10000 % 10;  
 digits[2] = n / 1000 % 10;  
 digits[3] = n / 100 % 10;  
 digits[4] = n / 10 % 10;  
 digits[5] = n % 10;  
 return digits;  
 }  
}

Задание 7.



public class Task17 {  
 public static void main(String[] args) {  
 int N = 1958675;  
 System.*out*.println(*reverse*(N));  
 }  
  
 public static String reverse(int n){  
 int digits[] = *getDigits*(n);  
  
 int temp = digits[0];  
 digits[0] = digits[6];  
 digits[6] = temp;  
  
 temp = digits[1];  
 digits[1] = digits[5];  
 digits[5] = temp;  
  
 temp = digits[2];  
 digits[2] = digits[4];  
 digits[4] = temp;  
  
 return ""+digits[0]+digits[1]+digits[2]+digits[3]+digits[4]+digits[5]+digits[6];  
 }  
  
 public static int[] getDigits(int n){  
 int digits[] = new int[7];  
 digits[0] = n / 1000000;  
 digits[1] = n / 100000 % 10;  
 digits[2] = n / 10000 % 10;  
 digits[3] = n / 1000 % 10;  
 digits[4] = n / 100 % 10;  
 digits[5] = n / 10 % 10;  
 digits[6] = n % 10;  
 return digits;  
 }  
}