**Задание 1–5)**

Задание 1 – task1() на рисунке 1;

Задание 3 – task3() на рисунке 1;

Задание 4)

Метод trimMargin() позволяет убрать "пустые" символы из текста по разделителю;

toShort();

toShortOrNull();

toInt();

toIntOrNull();

toFloat();

toDouble();

replace();

Задание 5 – task5() на рисунке 1;

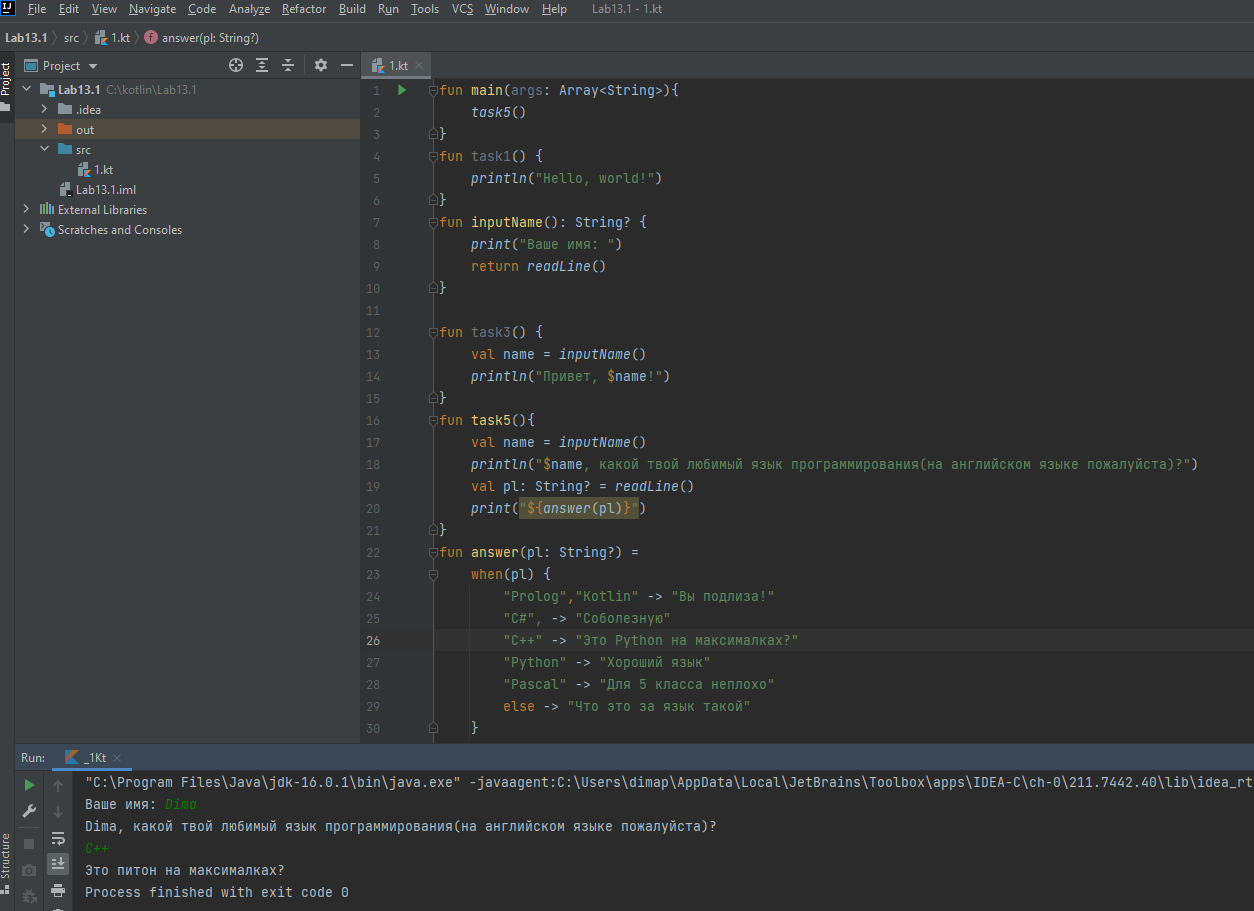
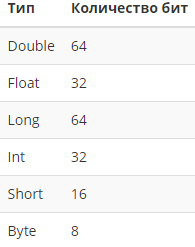


Рисунок 1 – Текст программы

**Задание 6)** Kotlin обрабатывает численные типы примерно так же, как и Java, хотя некоторые различия всё же присутствуют. Например, отсутствует неявное расширяющее преобразование для чисел, а литералы в некоторых случаях немного отличаются.



Каждый численный тип поддерживает следующие преобразования:

* toByte(): Byte
* toShort(): Short
* toInt(): Int
* toLong(): Long
* toFloat(): Float

**Пример**: val i: Int = b.toInt()

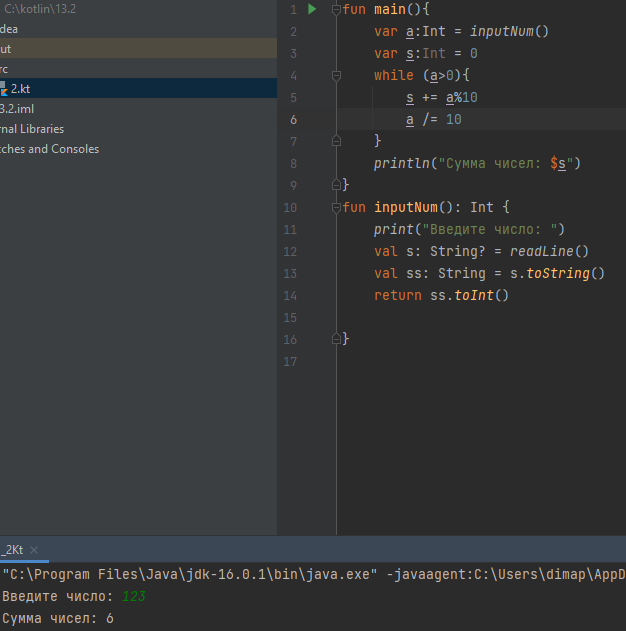
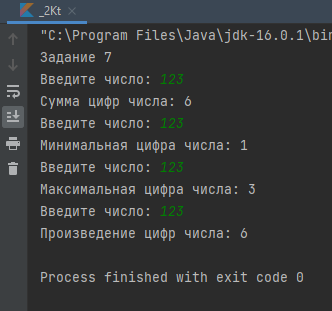


Рисунок 2 – Текст программы

**Задание 7)** Код программы ниже:

fun main(){  
 *println* ("Задание 7")  
 *println*(*sum*())  
 *println*(*min*())  
 *println*(*max*())  
 *println*(*pr*())  
  
}  
fun min():Int {  
 var a:Int = *inputNum*()  
 var s:Int = 10  
 while (a>0){  
 if (s > (a%10))  
 s = a%10  
 a /= 10  
 }  
 *print*("Минимальная цифра числа: ")  
 return s  
}  
fun max():Int {  
 var a:Int = *inputNum*()  
 var s:Int = 0  
 while (a>0){  
 if(s < a%10)  
 s = a%10  
 a /= 10  
 }  
 *print*("Максимальная цифра числа: ")  
 return s  
}  
fun pr():Int {  
 var a:Int = *inputNum*()  
 var s:Int = 1  
 while (a>0){  
 s \*= a%10  
 a /= 10  
 }  
 *print*("Произведение цифр числа: ")  
 return s  
}  
fun sum():Int {  
 var a:Int = *inputNum*()  
 var s:Int = 0  
 while (a>0){  
 s += a%10  
 a /= 10  
 }  
 *print*("Сумма цифр числа: ")  
 return s  
}  
fun inputNum(): Int {  
 *print*("Введите число: ")  
 val s: String? = *readLine*()  
 val ss: String = s.*toString*()  
 return ss.*toInt*()  
  
}

Результат работы программы:



**Задание 8)** Код 3 методов ниже:

fun method1():Int{  
 var a:Int = *inputNum*()  
 var n:Int = 0  
 for (i in 2..a){  
 if (i%2 == 0)  
 for (j in 2..i){  
 if (a%j==0)  
 if (i%j==0)  
 n+=1  
  
 if (a%j==0)  
 if (i%j==0)  
 break  
 }  
 }  
 return (n)  
}  
fun method2():Int{  
 var a:Int = *inputNum*()  
 var s:Int = -1  
 while (a>0){  
 if(s < a%10 && (a % 10) % 3 != 0)  
 s = a%10  
 a /= 10  
 }  
  
 return s  
}  
fun method3():Int{  
 var a:Int = *inputNum*()  
 var n:Int = 1  
 var b:Int = 1  
 for (i in 2..a) {  
 if (a%i == 0)  
 n = i  
 break  
 }  
 for (i in 2..a){  
  
 for (j in 2..i){  
  
 if (a % j == 0)  
 if (i % j == 0)  
 if (i % n != 0)  
 b = i  
 if (a % j == 0)  
 if (i % j == 0)  
 if (i % n != 0)  
 break  
 }  
 }  
 n = 0  
 a = b  
 while (b>0){  
 if (b%10<5)  
 n += b%10  
 b /= 10  
 }  
 return (a \* n)  
}

**Задание 9)** Измененная main() :

fun main(){  
 var a:String? = ""  
 *print*("Введите число: ")  
 var b:Int = *inputNum*()  
 while (a!="Закрыть") {  
 *print*("Введите команду")  
 a = *readLine*()  
 when (a){  
 "Новое число"->b = *inputNum*()  
 "Сумма"->*println*(*sum*(b))  
 "Минимальное"->*println*(*min*(b))  
 "Максимальное"->*println*(*max*(b))  
 "Произведение"->*println*(*pr*(b))  
 "Метод 1"->*print*(*method1*(b))  
 "Метод 2"->*print*(*method2*(b))  
 "Метод 3"->*print*(*method3*(b))  
 "Закрыть"-> break  
 else->*println*("Ожидаю команду")  
 }  
  
 }  
}

**Задание 10)**

Код 1 и 2 задачи ниже:

fun q1():Int{  
 var a:Double = 1.0  
 var s:Int = 0  
 for (n in 1..100){  
 a \*= n  
 }  
 while(a>0){  
 s += (a%10).toInt()  
 a /= 10  
 }  
 return s  
}  
fun q2():Int{  
  
 var a = ""  
  
 var n:Int = 1  
  
 while (a.*count*()<1000000){  
 a += n.toString()  
 n += 1  
 }  
 var s:Int = 0  
 n = 1  
 while (n<=1000000){  
 s += a[n-1].*digitToInt*()  
  
 n \*= 10  
 }  
 return (s)  
}