

# ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ

## ЛАБОРАТОРНАЯ РАБОТА №1

Решение всех задач оформить в виде одного класса с методами, решающими поставленные задачи. В классе могут присутствовать методы со спецификатором доступа `private` вспомогательного характера. В главном методе `Main` организовать вызов всех методов с дружественным интерфейсом. Если исходные данные вводятся с клавиатуры, то организовать проверку на ввод.

Первое задание предполагает исследование базовой информации по работе с типами данных. Второе задание позволяет отработать навыки применения условий с использованием инструкций `if` и `switch`. Третье задание позволяет отработать навыки работы с циклами через инструкции `for` и `while`. Четвертое задание предлагает отработать навыки решения типовых алгоритмов на массивах.

Необходимо решить по 5 задач из каждого задания согласно вашему варианту. Каждое задание оценивается по 0,4 балла. Максимально за лабораторную работу можно получить 10 баллов (8 баллов за решение задач + 2 балла за оформление отчета).

### Задание 1. Методы

1	<p>Дробная часть. Дана сигнатура метода: <b>public double fraction (double x);</b> Необходимо реализовать метод таким образом, чтобы он возвращал только дробную часть числа <b>x</b>. Подсказка: вещественное число может быть преобразовано к целому путем отбрасывания дробной части.</p> <p>Пример: <code>x=5,25</code> результат: <code>0,25</code></p>
2	<p>Сумма знаков. Дана сигнатура метода: <b>public int sumLastNums (int x);</b> Необходимо реализовать метод таким образом, чтобы он возвращал результат сложения двух последних знаков числа <b>x</b>, предполагая, что знаков в числе не менее двух. Подсказки: <code>int x=123%10; // x будет иметь значение 3</code> <code>int y=123/10; // y будет иметь значение 12</code></p> <p>Пример: <code>x=4568</code> результат: <code>14</code></p>

3	<p>Букву в число.  Дана сигнатура метода: <b>public int charToNum (char x);</b>  Метод принимает символ <b>x</b>, который представляет собой один из "0 1 2 3 4 5 6 7 8 9". Необходимо реализовать метод таким образом, чтобы он преобразовывал символ в соответствующее число. Подсказка: код символа '0' — это число 48.</p> <p>Пример:  x='3'  результат: 3</p>
4	<p>Есть ли позитив.  Дана сигнатура метода: <b>public bool isPositive (int x);</b>  Необходимо реализовать метод таким образом, чтобы он принимал число <b>x</b> и возвращал <b>true</b>, если оно положительное.</p> <p>Пример 1:  x=3  результат: true</p> <p>Пример 2:  x=-5  результат: false</p>
5	<p>Двузначное.  Дана сигнатура метода: <b>public bool is2Digits (int x);</b>  Необходимо реализовать метод таким образом, чтобы он принимал число <b>x</b> и возвращал <b>true</b>, если оно двузначное.</p> <p>Пример 1:  x=32  результат: true</p> <p>Пример 2:  x=516  результат: false</p>
6	<p>Большая буква.  Дана сигнатура метода: <b>public bool isUpperCase (char x);</b>  Необходимо реализовать метод таким образом, чтобы он принимал символ <b>x</b> и возвращал <b>true</b>, если это большая буква в диапазоне от 'A' до 'Z'.</p> <p>Пример 1:  x='D'  результат: true</p> <p>Пример 2:  x='q'  результат: false</p>

7	<p>Диапазон.</p> <p>Дана сигнатура метода: <b>public bool isInRange (int a, int b, int num);</b></p> <p>Метод принимает левую и правую границу (<b>a</b> и <b>b</b>) некоторого числового диапазона. Необходимо реализовать метод таким образом, чтобы он возвращал <b>true</b>, если <b>num</b> входит в указанный диапазон (включая границы). Обратите внимание, что отношение <b>a</b> и <b>b</b> заранее неизвестно (неясно кто из них больше, а кто меньше)</p> <p>Пример 1: a=5 b=1 num=3 результат: true</p> <p>Пример 2: a=2 b=15 num=33 результат: false</p>
8	<p>Делитель.</p> <p>Дана сигнатура метода: <b>public bool isDivisor (int a, int b);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал <b>true</b>, если любое из принятых чисел делит другое нацело.</p> <p>Пример 1: a=3 b=6 результат: true</p> <p>Пример 2: a=2 b=15 результат: false</p>
9	<p>Равенство.</p> <p>Дана сигнатура метода: <b>public bool isEqual(int a, int b, int c);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал <b>true</b>, если все три полученных методом числа равны</p> <p>Пример 1: a=3 b=3 c=3 результат: true</p> <p>Пример 2: a=2 b=15 c=2 результат: false</p>

10	<p>Множественный вызов.</p> <p>Дана сигнатура метода: <b>public int lastNumSum(int a, int b)</b></p> <p>Необходимо реализовать метод таким образом, чтобы он считал сумму цифр двух чисел из разряда единиц. Выполните с его помощью последовательное сложение пяти чисел и результат выведите на экран. Постарайтесь выполнить задачу, используя минимально возможное количество вспомогательных переменных.</p> <p>Пример:</p> <p>5+11 это 6</p> <p>6+123 это 9</p> <p>9+14 это 13</p> <p>13+1 это 4</p> <p>Итого 4</p>
----	---

## Задание 2. Условия

1	<p>Модуль числа.</p> <p>Дана сигнатура метода: <b>public int abs (int x);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал модуль числа <b>x</b> (если оно было положительным, то таким и остается, если оно было отрицательным – то необходимо вернуть его без знака минус).</p> <p>Пример 1:</p> <p>x=5</p> <p>результат: 5</p> <p>Пример 2:</p> <p>x=-3</p> <p>результат: 3</p>
2	<p>Безопасное деление.</p> <p>Дана сигнатура метода: <b>public double safeDiv (int x, int y);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал деление <b>x</b> на <b>y</b>, и при этом гарантировал, что не будет выкинута ошибка деления на 0. При делении на 0 следует вернуть из метода число <b>0</b>. Подсказка: смотри ограничения на операции типов данных.</p> <p>Пример 1:</p> <p>x=5 y=0</p> <p>результат: 0</p> <p>Пример 2:</p> <p>x=8 y=2</p> <p>результат: 4</p>

3	<p>Тридцать пять.</p> <p>Дана сигнатура метода: <b>public bool is35 (int x);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал <b>true</b>, если число <b>x</b> делится нацело на 3 или 5. При этом, если оно делится и на 3, и на 5, то вернуть надо <b>false</b>. Подсказка: оператор <b>%</b> позволяет получить остаток от деления.</p> <p>Пример 1: x=5 результат: true</p> <p>Пример 2: x=8 результат: false</p> <p>Пример 3: x=15 результат: false</p>
4	<p>Строка сравнения.</p> <p>Дана сигнатура метода: <b>public String makeDecision (int x, int y);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал строку, которая включает два принятых методом числа и корректно выставленный знак операции сравнения (больше, меньше, или равно).</p> <p>Пример 1: x=5 y=7 результат: "5&lt; 7"</p> <p>Пример 2: x=8 y=-1 результат: "8 &gt;-1"</p> <p>Пример 3: x=4 y=4 результат: "4==4"</p>
5	<p>Тройной максимум.</p> <p>Дана сигнатура метода: <b>public int max3 (int x, int y, int z);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал максимальное из трех полученных методом чисел. Подсказка: идеальное решение включает всего две инструкции <b>if</b> и не содержит вложенных <b>if</b>.</p> <p>Пример 1: x=5 y=7 z=7 результат: 7</p> <p>Пример 2: x=8 y=-1 z=4 результат: 8</p>

6	<p>Тройная сумма.  Дана сигнатура метода: <b>public bool sum3 (int x, int y, int z);</b>  Необходимо реализовать метод таким образом, чтобы он возвращал <b>true</b>, если два любых числа (из трех принятых) можно сложить так, чтобы получить третье.</p> <p>Пример 1:  x=5 y=7 z=2  результат: true</p> <p>Пример 2:  x=8 y=-1 z=4  результат: false</p>
7	<p>Двойная сумма.  Дана сигнатура метода: <b>public int sum2 (int x, int y);</b>  Необходимо реализовать метод таким образом, чтобы он возвращал сумму чисел <b>x</b> и <b>y</b>. Однако, если сумма попадает в диапазон от 10 до 19, то надо вернуть число 20.</p> <p>Пример 1:  x=5 y=7  результат: 20</p> <p>Пример 2:  x=8 y=-1  результат: 7</p>
8	<p>Возраст.  Дана сигнатура метода: <b>public String age (int x);</b>  Необходимо реализовать метод таким образом, чтобы он возвращал строку, в которой сначала будет число <b>x</b>, а затем одно из слов:  год  года  лет  Слово “год” добавляется, если число <b>x</b> заканчивается на 1, кроме числа 11.  Слово “года” добавляется, если число <b>x</b> заканчивается на 2, 3 или 4, кроме чисел 12, 13, 14.  Слово “лет” добавляется во всех остальных случаях.  Подсказка: оператор % позволяет получить остаток от деления.</p> <p>Пример 1:  x=5  результат: “5 лет”</p> <p>Пример 2:  x=31  результат: “31 год”</p> <p>Пример 3:  x=44  результат: “44 года”</p>

9	<p>День недели.</p> <p>Дана сигнатура метода: <b>public String day (int x);</b></p> <p>Метод принимает число x, обозначающее день недели. Необходимо реализовать метод таким образом, чтобы он возвращал строку, которая будет обозначать текущий день недели, где 1- это понедельник, а 7 – воскресенье. Если число не от 1 до 7 то верните текст “это не день недели”. Вместо <b>if</b> в данной задаче используйте <b>switch</b>.</p> <p>Пример: x=5 результат: “пятница”</p>
10	<p>Вывод дней недели.</p> <p>Дана сигнатура метода: <b>public void printDays (String x);</b></p> <p>В качестве параметра метод принимает строку, в которой записано название дня недели. Необходимо реализовать метод таким образом, чтобы он выводил на экран название переданного в него дня и всех последующих до конца недели дней. Если в качестве строки передан не день, то выводится текст “это не день недели”. Первый день понедельник, последний – воскресенье. Вместо <b>if</b> в данной задаче используйте <b>switch</b>.</p> <p>Пример 1: x=“четверг” результат: четверг пятница суббота воскресенье</p> <p>Пример 2: x=“чг” результат: это не день недели</p>

### Задание 3. Циклы

1	<p>Числа подряд.</p> <p>Дана сигнатура метода: <b>public String listNums (int x);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал строку, в которой будут записаны все числа от 0 до x (включительно).</p> <p>Пример: x=5 результат: “0 1 2 3 4 5”</p>
2	<p>Числа наоборот.</p> <p>Дана сигнатура метода: <b>public String reverseListNums (int x);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал строку, в которой будут записаны все числа от x до 0 (включительно).</p> <p>Пример: x=5 результат: “5 4 3 2 1 0”</p>

3	<p>Четные числа.</p> <p>Дана сигнатура метода: <b>public String chet (int x);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал строку, в которой будут записаны все четные числа от <b>0</b> до <b>x</b> (включительно). Подсказка для обеспечения качества кода: инструкцию <b>if</b> использовать не следует.</p> <p>Пример:  x=9  результат: "0 2 4 6 8"</p>
4	<p>Степень числа.</p> <p>Дана сигнатура метода: <b>public int pow (int x, int y);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал результат возведения <b>x</b> в степень <b>y</b>.</p> <p>Подсказка: для получения степени необходимо умножить единицу на число <b>x</b>, и сделать это <b>y</b> раз, т.е. два в третьей степени это <math>1*2*2*2</math></p> <p>Пример:  x=2  y=5  результат: 32</p>
5	<p>Длина числа.</p> <p>Дана сигнатура метода: <b>public int numLen (long x);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал количество знаков в числе <b>x</b>.</p> <p>Подсказка:  Int y=123/10; // y будет иметь значение 12</p> <p>Пример:  x=12567  результат: 5</p>
6	<p>Одинаковость.</p> <p>Дана сигнатура метода: <b>public bool equalNum (int x);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал <b>true</b>, если все знаки числа одинаковы, и <b>false</b> в ином случае.</p> <p>Подсказки:  int x=123%10; // x будет иметь значение 3  int y=123/10; // y будет иметь значение 12</p> <p>Пример 1:  x=1111  результат: true</p> <p>Пример 2:  x=1211  результат: false</p>



7	<p>Квадрат.</p> <p>Дана сигнатура метода: <b>public void square (int x);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он выводил на экран квадрат из символов '*' размером <b>x</b>, у которого <b>x</b> символов в ряд и <b>x</b> символов в высоту.</p> <p>Пример 1:</p> <p>x=2</p> <p>результат:</p> <pre>** **</pre> <p>Пример 2:</p> <p>x=4</p> <p>результат:</p> <pre>**** **** **** ****</pre>
8	<p>Левый треугольник.</p> <p>Дана сигнатура метода: <b>public void leftTriangle (int x);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он выводил на экран треугольник из символов '*' у которого <b>x</b> символов в высоту, а количество символов в ряду совпадает с номером строки.</p> <p>Пример 1:</p> <p>x=2</p> <p>результат:</p> <pre>* **</pre> <p>Пример 2:</p> <p>x=4</p> <p>результат:</p> <pre>* ** *** ****</pre>

9	<p>Правый треугольник.</p> <p>Дана сигнатура метода: <b>public void rightTriangle (int x);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он выводил на экран треугольник из символов '*' у которого <b>x</b> символов в высоту, а количество символов в ряду совпадает с номером строки, при этом треугольник выровнен по правому краю. Подсказка: перед символами '*' следует вывести необходимое количество пробелов.</p> <p>Пример 1: x=3 результат: * ** ***</p> <p>Пример 2: x=4 результат: * ** *** ****</p>
10	<p>Угадайка.</p> <p>Дана сигнатура метода: <b>public void guessGame()</b></p> <p>Необходимо реализовать метод таким образом, чтобы он генерировал случайное число от 0 до 9, далее считывал с консоли введенное пользователем число и выводил, угадал ли пользователь то, что было загадано, или нет. Метод запускается до тех пор, пока пользователь не угадает число. После этого выведите на экран количество попыток, которое потребовалось пользователю, чтобы угадать число.</p> <p>Пример: Введите число от 0 до 9: 5 Вы не угадали, введите число от 0 до 9: 9 Вы угадали! Вы отгадали число за 2 попытки</p>

#### Задание 4. Массивы

1	<p>Поиск первого значения.</p> <p>Дана сигнатура метода: <b>public int findFirst (int[] arr, int x);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал индекс <u>первого</u> вхождения числа <b>x</b> в массив <b>arr</b>. Если число не входит в массив – возвращается -1.</p> <p>Пример: arr=[1,2,3,4,2,2,5] x=2 результат: 1</p>
---	---

2	<p>Поиск последнего значения.  Дана сигнатура метода: <b>public int findLast (int[] arr, int x);</b>  Необходимо реализовать метод таким образом, чтобы он возвращал индекс <u>последнего</u> вхождения числа <b>x</b> в массив <b>arr</b>. Если число не входит в массив – возвращается -1.</p> <p>Пример:  arr=[1,2,3,4,2,2,5]  x=2  результат: 5</p>
3	<p>Поиск максимального.  Дана сигнатура метода: <b>public int maxAbs (int[] arr);</b>  Необходимо реализовать метод таким образом, чтобы он возвращал наибольшее по модулю (то есть без учета знака) значение массива <b>arr</b>.</p> <p>Пример:  arr=[1,-2,-7,4,2,2,5]  результат: -7</p>
4	<p>Добавление в массив.  Дана сигнатура метода: <b>public int[] add (int[] arr, int x, int pos);</b>  Необходимо реализовать метод таким образом, чтобы он возвращал новый массив, который будет содержать все элементы массива <b>arr</b>, однако в позицию <b>pos</b> будет вставлено значение <b>x</b>.</p> <p>Пример:  arr=[1,2,3,4,5]  x=9  pos=3  результат: [1,2,3,9,4,5]</p>
5	<p>Добавление массива в массив.  Дана сигнатура метода: <b>public int[] add (int[] arr, int[] ins, int pos);</b>  Необходимо реализовать метод таким образом, чтобы он возвращал новый массив, который будет содержать все элементы массива <b>arr</b>, однако в позицию <b>pos</b> будут вставлены значения массива <b>ins</b>.</p> <p>Пример:  arr=[1,2,3,4,5]  ins=[7,8,9]  pos=3  результат: [1,2,3,7,8,9,4,5]</p>
6	<p>Реверс.  Дана сигнатура метода: <b>public void reverse (int[] arr);</b>  Необходимо реализовать метод таким образом, чтобы он изменял массив <b>arr</b>. После проведенных изменений массив должен быть записан задом-наперед.</p> <p>Пример:  arr=[1,2,3,4,5]  результат: arr=[5,4,3,2,1]</p>

7	<p>Возвратный реверс.</p> <p>Дана сигнатура метода: <b>public int[] reverseBack (int[] arr);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал новый массив, в котором значения массива <b>arr</b> записаны задом наперед.</p> <p>Пример:  arr=[1,2,3,4,5]  результат: [5,4,3,2,1]</p>
8	<p>Объединение.</p> <p>Дана сигнатура метода: <b>public int[] concat (int[] arr1,int[] arr2);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал новый массив, в котором сначала идут элементы первого массива (<b>arr1</b>), а затем второго (<b>arr2</b>).</p> <p>Пример:  arr1=[1,2,3]  arr2=[7,8,9]  результат: [1,2,3,7,8,9]</p>
9	<p>Все вхождения.</p> <p>Дана сигнатура метода: <b>public int[] findAll (int[] arr, int x);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал новый массив, в котором записаны индексы всех вхождений числа <b>x</b> в массив <b>arr</b>.</p> <p>Пример:  arr=[1,2,3,8,2,2,9]  x=2  результат: [1,4,5]</p>
10	<p>Удалить негатив.</p> <p>Дана сигнатура метода: <b>public int[] deleteNegative (int[] arr);</b></p> <p>Необходимо реализовать метод таким образом, чтобы он возвращал новый массив, в котором записаны все элементы массива <b>arr</b> кроме отрицательных.</p> <p>Пример:  arr=[1,2,-3,4,-2,2,-5]  результат: [1,2,4,2]</p>