

Пример спецификации к Лабораторным работам №2-4 (Вариант 0)

Абстракция А0

1. Постановка задачи (ПЗ).

Задание: Написать программу обработки одномерного массива(ов) в соответствии с условием. (полностью задание и требования в Лабораторной работе №2)

Условие: Координатами x, y заданы n точек плоскости. Найти процент точек, удаление которых от начала координат больше заданной величины r и притом обе координаты положительны, а также подсчитать среднее удаление всех точек от начала координат.

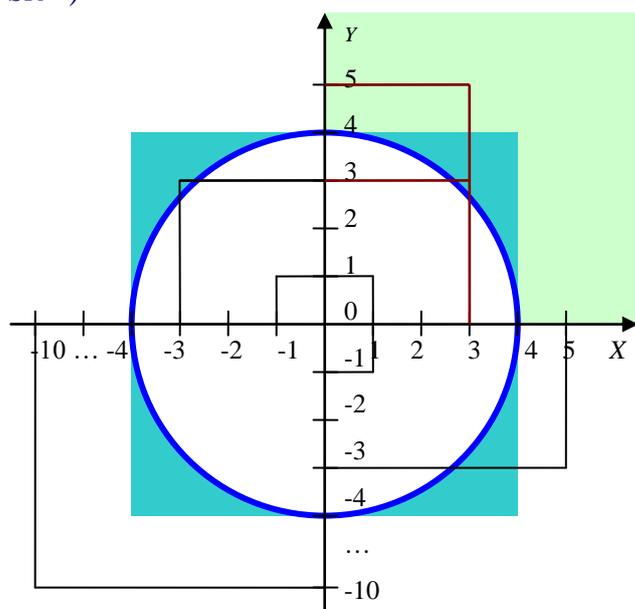
2. Уточненная постановка задачи (добавить недостающие имена, определить типы и структуры, добавить альтернативные (если есть – «отрицательные», если нужен особый ответ – «положительные») решения).

Даны два одномерных вещественных массива x и y из n ($0 < n \leq 20$) элементов – координаты точек плоскости, и целая величина r .

Найти:

p – процент точек, удаление которых от начала координат больше заданной величины r и притом обе координаты положительны; (Пример особого «положительного» ответа: Если $p=100\%$, то вывести сообщение «ВСЕ 100% точек лежат в заданной области»)

sr – среднее удаление всех точек от начала координат. (Пример «отрицательного» ответа: Если бы среднее удаление нужно было найти не среди ВСЕХ точек, а только среди НЕопавших в заданную для поиска r область, то следовало бы добавить следующее предложение: Если нет точек, НЕопавших в заданную область, то вывести сообщение «Невозможно вычислить среднее удаление SR»)



3. Пример с иллюстрацией (станет одним из тестов): (графическая иллюстрация в примере не всегда возможна, но всегда желательна)

Возьмем $r = 4$, $n = 10$. И точки из разных четвертей и областей (вне круга, в описанном квадрате, вне квадрата, на границах области, их пересечении):

x	1	1	-1	3	-3	0	3	5	5	-10
y	1	-1	1	3	3	4	5	-3	0	-10

Есть две точки с положительными координатами вне круга с радиусом $r = 4$. Это точки (3,3) и (3,5). Значит, искомый процент $p = 2/10 * 100\% = 20\%$

Среднее удаление всех точек от начала координат $sr = \frac{3\sqrt{1+1} + 2\sqrt{9+9} + 4 + 2\sqrt{25+9} + 5 + \sqrt{2} * 100}{10} \approx 4,753196$

4. Таблица данных

(начало таблицы данных)

Класс	Имя	Описание (смысл, диапазон, точность)	Тип	Структура	Формат в/в
Входные данные	n	число точек плоскости, $0 < n \leq 20$	цел	простая переменная	XX (:2)
	x	абсциссы точек плоскости, $ x_i \leq 10$, точн. 0.1	вещ	одномерный массив (20)	+XX.X (:5:1)
	y	ординаты точек плоскости, $ y_i \leq 10$, точн. 0.1	вещ	одномерный массив (20)	+XX.X (:5:1)
	r	критическое удаление, $1 \leq r \leq 14$ т.к. $0 < r < \sqrt{10^2 + 10^2} \approx 14,14$	цел	простая переменная	XX (:2)

(продолжение таблицы данных)

Класс	Имя	Описание (смысл, диапазон, точность)	Тип	Структура	Формат в/в
Выходные данные	<i>p</i>	искомый процент точек, $0 \leq p \leq 100$, точн. 0.1	вещ	простая переменная	XXX.X (:5:1)
	<i>sr</i>	среднее удаление всех точек от начала координат, $0 \leq sr \leq \sqrt{10^2 + 10^2} \leq 10\sqrt{2} < 15$; точн. 0.00001	вещ	простая переменная	XX.XXXXX (:8:5)
Промежуточные данные*	<i>i</i>	индекс текущего элемента, $0 \leq i \leq 21$ (в Delphi выход из цикла с 1 до 20 с шагом +1 происходит при значении 21, из цикла с 20 до 1 – при 0)	цел	простая переменная	---

*Промежуточные данные заполняются не сразу, а по мере необходимости использовать дополнительные переменные. Оставляйте место для них

5. Входная форма:

1.1 обр1	Количество точек n: _____
1.2	<n>
2.1 обр2	Критическое удаление r: _____
2.2	<r>
обр3.1	Абсциссы и ординаты: _____
Обр3.2	<x[1]> <y[1]> <x[2]> <y[2]> <x[n]> <y[n]>

Образцы 1.1, 2.1 и 3 - приглашения для ввода n, r и координат точек в режиме диалога.

В лабораторной работе 4 (ввод из собственного файла) их можно не записывать в файл вовсе:

1.2	<n>
2.2	<r>
Обр3.2	<x[1]> <y[1]> <x[2]> <y[2]> <x[n]> <y[n]>

6. Выходная форма:

Обр4	Лаб.2
Обр5	Количество точек n = <n>
Обр6	Критическое удаление r = <r>
Обр7	Абсциссы и ординаты: _____
обр8	<x[1]> <y[1]> <x[2]> <y[2]> <x[n]> <y[n]>
обр9	Процент точек p = <p>
обр10	Среднее удаление sr = <sr>
Обр11	Сюда нужно добавить альтернативные положительные и отрицательные ответы, чтобы пронумеровать образцы ответов.

35 пробелов перед текстом (это примерно центр экрана при его ширине 80). Также полезно учитывать высоту экрана – 25 строк

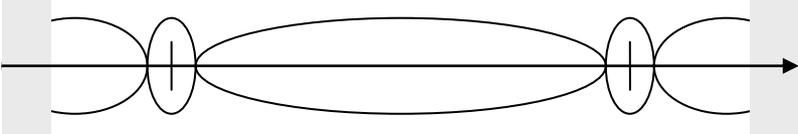
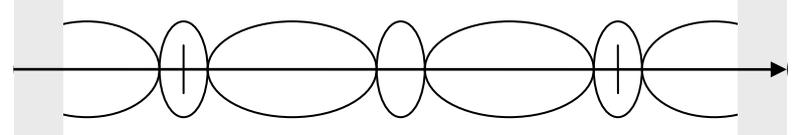
7. Аномалии

Пока не рассматриваем, но можете сделать ($n < 1, n > 20, r < 1, r > 14, \exists i: |x[i]| > 10, \exists i: |y[i]| > 10$), например:

№	Описание	Условие возникновения**	Реакция на аномалию
1	n меньше минимально допустимого значения	$n < 1$	Сообщение: «Некорректное n : $n < 1$ » (обр.А1) Действие: Завершение работы программы
2	...		

**В условиях желательно НЕ использовать связку \vee (ИЛИ), объединяя разные случаи воедино, и выводить *разные* сообщения, чтобы реакция была более конкретная.

8. Функциональные тесты

Исходные данные							Результаты		Тест №	
	<i>аном</i>	<i>граница</i>	<i>Средние значения</i>			<i>граница</i>	<i>аном</i>	p	$макс = 100$	2
									$мин = 0$ (для вещ. чисел про минимум по абсолют. значению, НЕ равный нулю, читайте ниже)	3
n	< 1	1	[2, 19]			20	> 20		$сред = (0, 100)$	1
Тест№	---	3	1			2	---		0	3
	<i>аном</i>	<i>граница</i>	<i>Средние значения</i>			<i>граница</i>	<i>аном</i>		$не\ сущ = не\ возможно$	---
r	< 1	1	[2, 13]			14	> 14		$Макс.выч.нагрузка = 20\ точек\ из\ 20\ попали$	2
Тест№	---	3	1			2	---			
	<i>аном</i>	<i>граница</i>	<i>отриц</i>	0	<i>полож</i>	<i>граница</i>	<i>аном</i>	sr	$макс = 10\sqrt{2}$	2
									$мин = 0$ (для вещ. чисел про минимум по абсолют. значению, НЕ равный нулю, читайте ниже)	3
$x[i]$	< -10	-10	(-10,0)	0	(0,10)	10	> 10		$сред = (0, 10\sqrt{2})$	1
Тест№	---	1	1	1	1	2	---		0	3
	<i>аном</i>	<i>граница</i>	<i>сред</i>	0	<i>сред</i>	<i>граница</i>	<i>аном</i>		$не\ сущ = не\ возможно$	---
$y[i]$	< -10	-10	(-10,0)	0	(0,10)	10	> 10		$Макс.выч.нагрузка = 20\ точек\ по\ (\pm 10, \pm 10)$	2
Тест№	---	1	1	1	1	2	---			

ВАЖНО! Ожидаемые результаты необходимо считать САМИМ вручную (на калькуляторе или другим способом, например, в Excel), чтобы затем **сравнивать** с тем, что выдаст ваша программа. А не записывать, что она (программа) выдала!

№ теста	Входные данные	Ожидаемый результат	Смысл теста																						
1	$n=10, r=4$ <table border="1" style="display: inline-table;"> <tr><td>x</td><td>1</td><td>1</td><td>-1</td><td>3</td><td>-3</td><td>0</td><td>3</td><td>5</td><td>5</td><td>-10</td></tr> <tr><td>y</td><td>1</td><td>-1</td><td>1</td><td>3</td><td>3</td><td>4</td><td>5</td><td>-3</td><td>0</td><td>-10</td></tr> </table>	x	1	1	-1	3	-3	0	3	5	5	-10	y	1	-1	1	3	3	4	5	-3	0	-10	$p=20$ (обр9) $sr=4,753196$ (обр10)	Данные взяты из примера. Разнообразное расположение точек по четвертям и областям
x	1	1	-1	3	-3	0	3	5	5	-10															
y	1	-1	1	3	3	4	5	-3	0	-10															
2	$n=20; r=14$ <table border="1" style="display: inline-table;"> <tr><td>x</td><td>10</td><td>10</td><td>...</td><td>10</td></tr> <tr><td>y</td><td>10</td><td>10</td><td>...</td><td>10</td></tr> </table>	x	10	10	...	10	y	10	10	...	10	$p=100$ (обр9) $sr=10\sqrt{2} = 14,14214$ (обр10)	Максимальное значение всех исходных данных и результатов. Максимальная вычислительная нагрузка для p и sr												
x	10	10	...	10																					
y	10	10	...	10																					
3	$n=1; r=1$ <table border="1" style="display: inline-table;"> <tr><td>x</td><td>0</td></tr> <tr><td>y</td><td>0</td></tr> </table>	x	0	y	0	$p=0$ (обр9) $sr=0$ (обр10)	Минимальные значения n, r, p, sr . Нулевые значения x, y, p, sr																		
x	0																								
y	0																								
	Далее тесты на аномальные ситуации, если их рассматриваете, например: $n=0$	Сообщение по обр.А1	Аномалия №1 ...																						

При создании тестов можно выделить больше классов эквивалентности для выходных и исходных данных, например:

А) для вещественных значений: если при совпадении минимума или максимума с 0, искать минимум по абсолютной величине, отличный от 0. Это поможет проверить точность выходных данных.

Например, у нас минимумы p и sr совпали с 0. Выбираем новые минимумы, лучше сразу два:

Результаты	Тест №																						
p	2																						
макс =100 мин и не 0: $1/20=0,05 \rightarrow 5\%$ 20 кратно 10, поэтому один необходимый знак в дробной части «спрятался», возьмем 19: $1/19=0,051 \rightarrow 5,26\%$ без дробной части было бы тоже 5%, а $1/20$ не равна $1/19!$	Нужно такие два теста добавить 4,5																						
$сред = (0,100)$	1																						
$не\ суц = не\ возможно$	---																						
0	3																						
Макс.выч.нагрузка = 20 подходящих точек	2																						
sr	2																						
макс = $10\sqrt{2}$ мин и не 0: $n=20$ <table border="1" style="margin: 5px 0;"> <tr><td>x</td><td>0,1</td><td>0</td><td>...</td><td>0</td><td>0</td></tr> <tr><td>y</td><td>0</td><td>0</td><td>...</td><td>0</td><td>0</td></tr> </table> $sr = (0,1 + 19*0)/20 = 0,005$ - снова проблема с 20 и с корнем из квадрата 0,1 – пара знаков дробной части могла не проявить себя, возьмем 19: $n=19$ <table border="1" style="margin: 5px 0;"> <tr><td>x</td><td>0,1</td><td>0</td><td>...</td><td>0</td></tr> <tr><td>y</td><td>0</td><td>0</td><td>...</td><td>0</td></tr> </table> $sr = (0,1 + 18*0)/19 = 0,00526$, для различия с предыдущим средним удалением нужно как минимум 4 знака после запятой при выводе результата.	x	0,1	0	...	0	0	y	0	0	...	0	0	x	0,1	0	...	0	y	0	0	...	0	Нужно такие два теста добавить 6,7
x	0,1	0	...	0	0																		
y	0	0	...	0	0																		
x	0,1	0	...	0																			
y	0	0	...	0																			
$сред = (0, 10\sqrt{2})$	1																						
$не\ суц = не\ возможно$	---																						
0	3																						
Макс.выч.нагрузка = 20 точек по (10,10)	2																						

Б) Даже если минимум/максимум не совпадает с нулем, можно в любом случае вместо максимума делать тесты на граничные значения результата (максимальное значение и максимальное по абсолютной величине отрицательное значение), а в качестве минимума – отрицательное и/или положительное значения, по абсолютной величине ближайšie к 0;

В) Для исходных данных при проверке аномальных ситуаций можно рекомендовать для проверки брать значение, ближайšie к граничному – пограничное, чтобы четче проверять границу аномального и нормального значения переменной;

Г) При проверке аномальных ситуаций, если 0 (ноль) входит в область аномальной ситуации, проверять отдельно нули и НЕнули. Например, для ($n < 1$) отдельно тесты для 0 и -1;

Д) Можно расширить проверку исходных данных на аномалии, когда введены не цифры, а буквы или «лишние» символы присутствуют во введённом значении. Или, как частный случай, вместо точки в качестве разделителя дробной части использована запятая, или наоборот. В лабораторной работе 9 будут проверки ошибок при открытии и чтении из файла. Ошибки при чтении как раз могут быть из-за некорректных символов и попытки ввода букв вместо цифр.

9. Метод

Отделим ввод-вывод от обработки данных (собственно решения задачи).

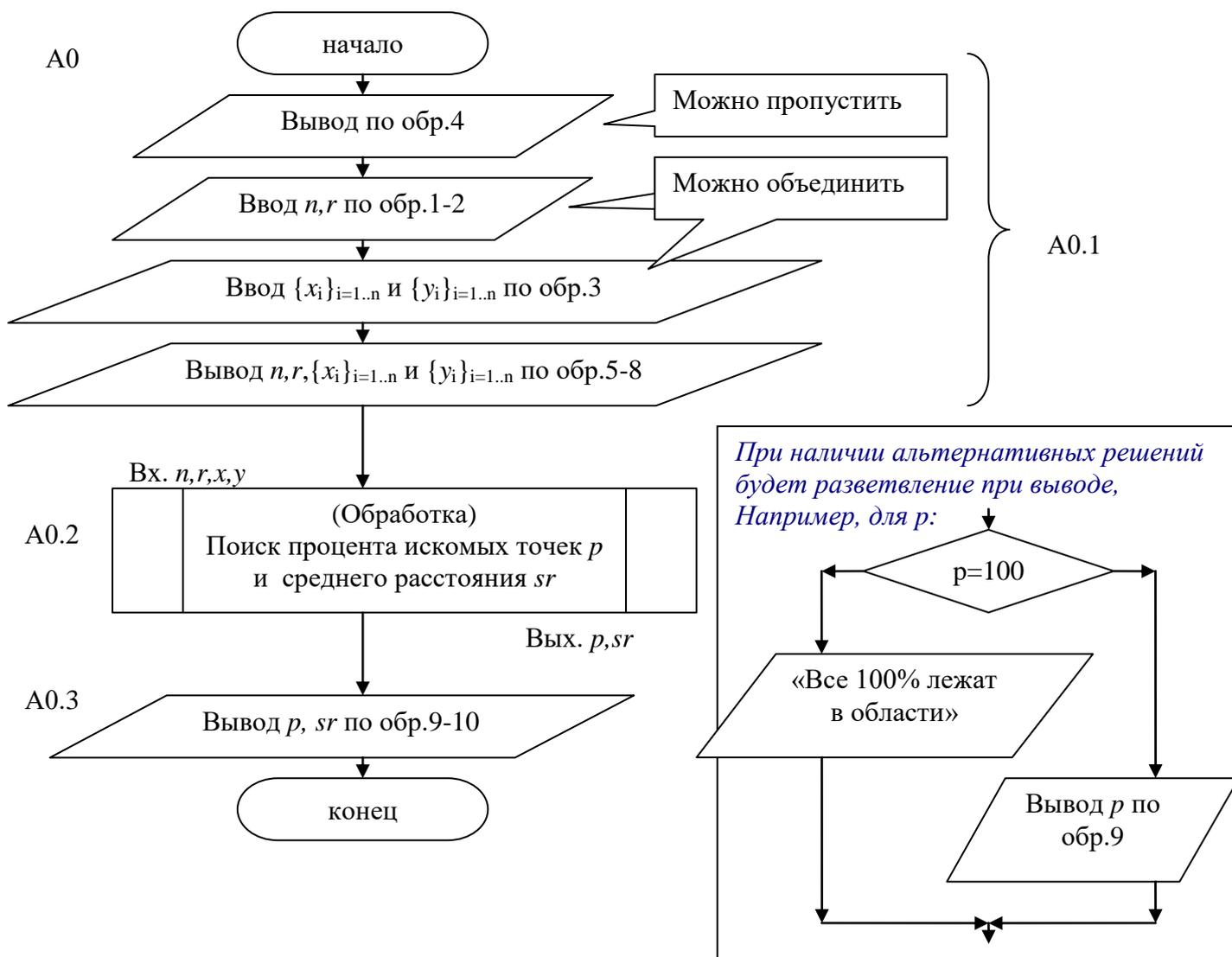
То есть, разделим нашу задачу на три подзадачи:

1. Подзадача A0.1. Ввести исходные данные (обр.1-3) и вывести их на экран (обр.5-8) для визуального подтверждения.
2. Подзадача A0.2. Решение задачи: на основе введенных исходных данных (n, r, x, y) найти процент подходящих точек p и среднее расстояние sr .
3. Подзадача A0.3. Ввести вывести полученные результаты (p, sr) на экран (обр.9-10)

Здесь ввод-вывод тривиален и представляет собой чисто техническую задачу; в алгоритмизации нуждается только обработка. Поэтому на нулевом уровне сразу опишем ввод-вывод (подзадачи A0.1 и A0.3), оставив задачу обработки A0.2 в виде абстракции (которую раскроем на следующем уровне проектирования). На данном уровне эту абстракцию в алгоритме заменим ЗАГЛУШКОЙ («Заглушка» – это простейший алгоритм, временно заменяющий реальный алгоритм и помогающий уменьшить область поиска ошибок и отладить уже написанную часть программы, не внося дополнительных ошибок, потенциально содержащихся в том самом реальном алгоритме. Алгоритм «заглушки» должен быть настолько простым, чтобы в нем не было ошибок или было очень просто их обнаружить и исправить. В простейшем случае, это просто присваивание результата. На следующем шаге разработки «заглушка» заменяется реальным алгоритмом, и, поскольку остальная часть программы уже проверена, то можно с высокой долей вероятности считать, что все вновь обнаруженные ошибки находятся в добавленном алгоритме.)

Для ввода-вывода массивов потребуется промежуточная переменная - индекс элемента. Пусть i – индекс текущего элемента массива. (Добавляем его в таблицу данных)

10. Алгоритм (блок-схемы можно рисовать вручную, оставив место в распечатке)



11. Программа на Delphi. Диалоговый вариант {ввод с клавиатуры, вывод на экран}

Комментариев в программе можно так много не делать

```

program Lab2;
{$APPTYPE CONSOLE}
Uses Windows; // для русификации
  {раздел определения констант:}
  const
    Nmax=20; {верхняя граница индексов массива – максимальное количество точек}
  {раздел описания типов type- пустой
Или можно определить тип массива:
  Type Tmassiv = array [1..Nmax] of real;}

  {раздел описания переменных:}
  var
    n, i, r: integer;
    p, sr: real;
    x, y: array [1..Nmax] of real;
    {или, если выше описан тип, то
    x,y: Tmassiv;}

  {раздел операторов:}
  begin
    setConsoleCP(1251); // для ввода русификации
    setConsoleOutputCP(1251); // для вывода

```

Использование именованной константы **Nmax** для границ лучше с точки зрения стиля.

Если константа упомянута несколько раз в коде (допустим есть ещё строка: *if n>Nmax then*), то изменение значения такой константы позволит изменить значения по всему коду. Без создания именованной константы можно случайно пропустить одно из значений и внести ошибку в программу.

Имя константы произвольно, но должно отражать смысл во избежание путаницы, а значит и ошибок.

```

{A0.1 - ввод-вывод входных данных}
  {вывод заголовка в выходной документ: }
  writeln('Лаб.2' :40); {вывод с переходом на следующую строку}

  {ввод исходных данных: }
  writeln('Количество точек n:'); readln(n); {обр 1.1,1.2}
  writeln('Критическое удаление r:'); readln(r); {обр 2.1,2.2}
  {ввод массивов x и y:}
  writeln('Абсциссы и ординаты:'); {обр3.1}
  for i:=1 to n do
    readln(x[i], y[i]); {обр3.2:ввод 2 элементов через пробел и переход на след. строку}

  {вывод входных данных в выходной документ для подтверждения: }
  for i:=1 to 80 do write('='); {отделим визуально чертой и строкой введенные и выводимые значения }
  writeln;

  {вывод n, r по обр5-6}
  writeln('    Количество точек =', n:3);
  writeln('    Критическое удаление = ', r:2);
  {вывод массивов x и y}
  writeln('    Абсциссы и ординаты: '); {обр7}
  for i:=1 to n do
    writeln(' :5, x[i]:5:1, ' :8, y[i]:5:1); {обр8}

{A0.2 - обработка}

  // p:=20; sr:=4.573196;      {заглушка для теста 1}
  p:=100; sr:=10*sqrt(2);    {заглушка для теста 2}
  // p:=0; sr:=0;           {заглушка для теста 3}

{A0.3 - вывод результатов}
  writeln('    Искомый процент точек = ',p:5:1); {p по обр9}
  writeln('    Среднее удаление = ',sr:8:5); {sr по обр10}

  write('Press Enter...'); readln; {задерживаем экран до нажатия ENTER}
end.

```

Раскрытие абстракции A0.2

Поскольку условие всей задачи **совпадает** (за исключением необходимости ввода и вывода) с условием выделенной подзадачи, вместо полной спецификации с пунктами 1-11. сделаем просто дополнение к пунктам 4, 9-11.

9. Метод

Пусть

i – номер текущей точки,

si – удаление точки $(x[i], y[i])$ от начала координат; тогда

$$si = \sqrt{x[i]^2 + y[i]^2}$$

Пусть s – сумма расстояний всех точек от начала координат; тогда

$sr = s/n$; чтобы найти sr , надо найти s :

$$s = \sum_{i=1}^n si$$

Пусть np – число точек, попавших в область (удаление больше r , коорд-ты положительны); тогда

$$p = (np/n)*100;$$

чтобы найти p , надо найти np :

$$np = \text{Count}_{i=1}^n ((si > r) \wedge (x[i] > 0) \wedge (y[i] > 0))$$

Методы подсчета суммы и количества описаны в файле *Base-Algorithms.doc*

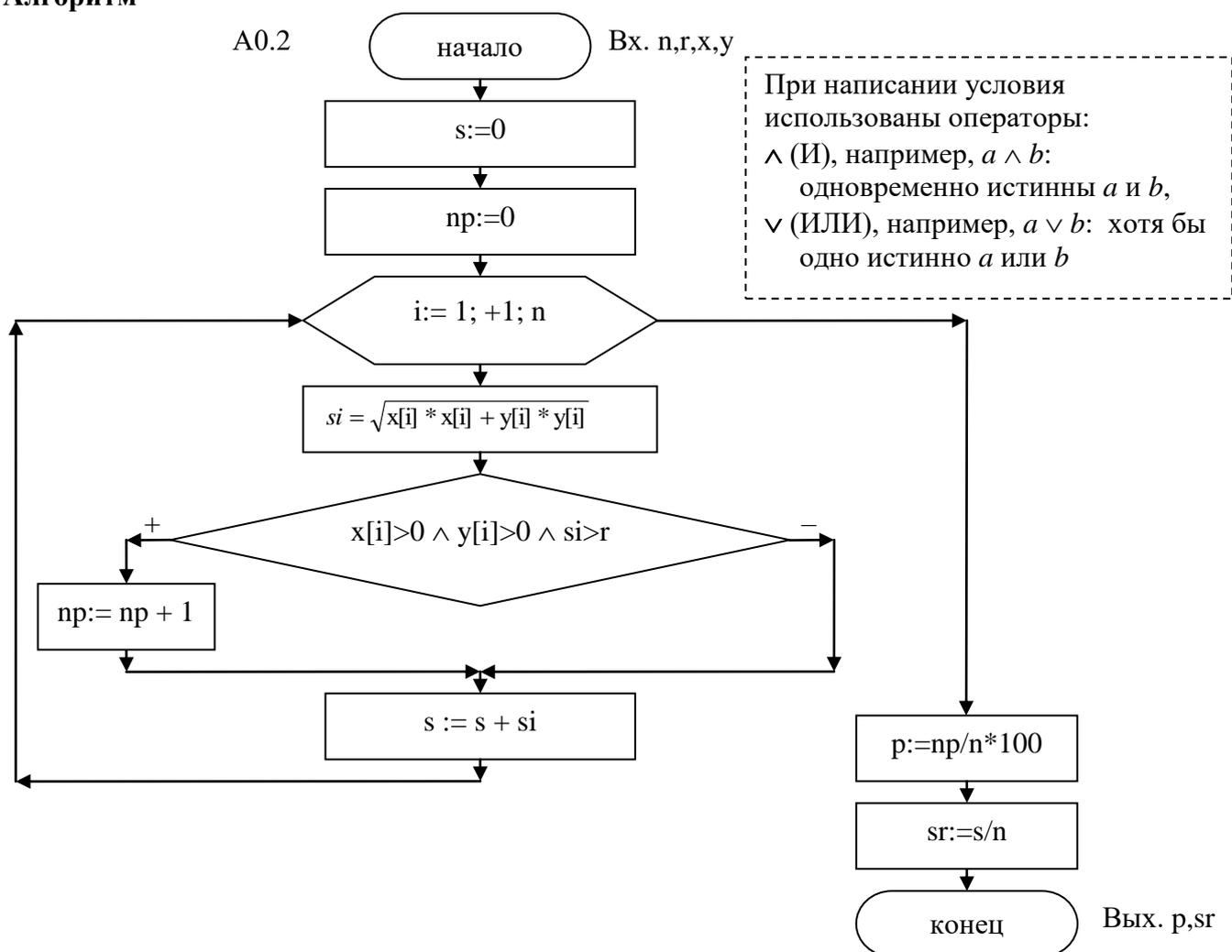
Оба алгоритма (поиск суммы и количества) выполним в одном и том же цикле по элементам массивов, иначе расстояние (si) каждой точки от центра пришлось бы считать дважды.

4. Таблица данных (продолжение)

(в рукописной спецификации можно дополнить исходную таблицу)

Промежуточные	i	индекс текущего элемента, $0 < i \leq 21$	цел	простая переменная	---
	np	число точек, удовлетворяющих условию, $0 \leq np \leq 20$	цел	простая переменная	---
	s	сумма расстояний всех точек от начала координат, $0 \leq s \leq 20\sqrt{10^2 + 10^2} \leq 20*10*\sqrt{2} < 300$; точн. 0.001	вещ	простая переменная	---
	si	расстояние до i -ой точки от начала координат, $0 \leq si \leq \sqrt{10^2 + 10^2} = 10\sqrt{2} \leq 15$; точн. 0.001	вещ	простая переменная	---

10. Алгоритм



11. Программный код

Алгоритм не будем оформлять в виде отдельной процедуры, а просто вставим в основной алгоритм вместо заглушки.

В разделе **var** следует добавить описание трех переменных:

```

{описания других промежуточных данных: }
var
  np: integer;
  si, s: real;

```

Вместо заглушки в раздел операторов программного кода:

```

{A0.2 - обработка}
s:=0;
np:=0;

for i:=1 to n do // перебираем по-очереди все n точек
begin
  si:= sqrt(x[i]*x[i] + y[i]*y[i]); {вычисляем удаление точки от центра координат }
  s:= s+si; // и сумму
  if (x[i]>0) and (y[i]>0) and (si>r) then // если точка в нужной области, то
    np:=np+1; // и количество увеличиваем
end;

p:=np / n *100;
sr:= s / n;

```

Получившиеся программы
Лабораторная работа 3:

```

program Lab3;
{$APPTYPE CONSOLE}
Uses Windows; // для русификации
{раздел определения констант:}
const
  Nmax=20; {верхняя граница индексов массива – максимальное количество точек}
{раздел описания переменных:}
var
  n, i, r: integer;
  p, sr: real;
  x, y: array [1..Nmax] of real;

  np: integer;
  si, s: real;
{раздел операторов:}
begin
  setConsoleCP(1251); // для ввода русификация и
  setConsoleOutputCP(1251); // для вывода

  {A0.1 - ввод-вывод входных данных}
  {вывод заголовка в выходной документ: }
  writeln('Лаб.3' :40); {вывод с переходом на следующую строку}
  {ввод исходных данных: }
  writeln('Количество точек n:'); readln(n); {обр 1.1,1.2}
  writeln('Критическое удаление r:'); readln(r); { обр 2.1,2.2}
  {ввод массивов x и y:}
  writeln('Абсциссы и ординаты:'); {обр3.1}
  for i:=1 to n do
    readln(x[i], y[i]); {обр3.2:ввод 2 элементов через пробел и переход на след. строку}

  {вывод входных данных в выходной документ для подтверждения: }
  for i:=1 to 80 do write('='); { отделим визуально чертой и строкой введенные и выводимые значения }
  writeln;

  {вывод n, r по образцам 5-6}
  writeln('Количество точек =', n:3);
  writeln('Критическое удаление = ', r:2);
  {вывод исходных массивов x и y}
  writeln('Абсциссы и ординаты: '); {обр7}
  for i:=1 to n do
    writeln(' :5, x[i]:5:1, ' ':8, y[i]:5:1); {обр8}

  {A0.2 - обработка}
  s:=0;
  np:=0;

  for i:=1 to n do // перебираем по-очереди все n точек
  begin
    si:= sqrt(x[i]*x[i] + y[i]*y[i]); {вычисляем удаление точки от центра координат }
    s:= s+si; // и сумму
    if (x[i]>0) and (y[i]>0) and (si>r) then // если точка в нужной области, то
      np:=np+1; // и количество увеличиваем
  end;

  p:=np / n *100;
  sr:= s / n;

  {A0.3 - вывод результатов}
  writeln(' Искомый процент точек = ',p:5:1); {p по обр9}
  writeln(' Среднее удаление = ',sr:8:5);{sr по обр10}

  write('Press Enter...'); readln; {задерживаем экран до нажатия ENTER}
end.

```

Лабораторная работа 4:

```

program Lab4;
{$APPTYPE CONSOLE}
Uses Windows; // для русификации
const
  Nmax=20; {
var
  n, i, r: integer;
  p, sr: real;
  x, y: array [1..Nmax] of real;
  np: integer;
  si, s: real;
  fin, fout: TextFile; { описание файловых переменных, для работы с текстовыми файлами}
begin
  setConsoleCP(1251); // для ввода русификация и
  setConsoleOutputCP(1251); // для вывода

```

```

{Для работы программы необходимо передать ей параметры - имена исходного и выходного файла:
Run→ Parameters в среде разработки или через ярлык exe-файла или bat-файл – читайте семинар Sem2.pdf}
if ParamCount<2 then begin writeln('CriticalSection: Не заданы параметры программы'); readln; exit end;

AssignFile(fin, ParamStr(1)); AssignFile(fout, ParamStr(2));
Reset(fin); {открытие для чтения текстового файла, связанного с fin в предыдущей строке}
Rewrite(fout); {создание и открытие для записи файла, связанного с переменной fout}

```

```

{A0.1 - ввод-вывод входных данных}
{вывод заголовка в выходной документ: }
writeln(fout, 'Лаб.4' :40); {вывод в файл, связанный с переменной fout, и переход на след.строку в нём}

{ввод исходных данных из файла, связанного с переменной fin, и открытого для чтения: }
readln(fin, n); {обр 1.2}
readln(fin,r); {обр 2.2}
{ввод массивов x и y:}
for i:=1 to n do
  readln(fin,x[i], y[i]); {обр3.2:ввод 2 элементов через пробел и переход на след. строку}

{вывод n, r по образцам 5-6}
writeln(fout,' Количество точек =', n:3);
writeln(fout,' Критическое удаление = ', r:2);
{вывод массивов x и y}
writeln(fout,' Абсциссы и ординаты: '); {обр7}
for i:=1 to n do
  writeln(fout,' :5, x[i]:5:1, ' ':8, y[i]:5:1); {обр8}

```

```

{A0.2 - обработка}
s:=0; np:=0;
for i:=1 to n do // перебираем по-очереди все n точек
begin
  si:= sqrt(x[i]*x[i] + y[i]*y[i]);
  s:= s+si;
  if (x[i]>0) and (y[i]>0) and (si>r) then
    np:=np+1;
end;
p:=np / n *100;
sr:= s / n;

```

```

{A0.3 - вывод результатов}
writeln(fout,' Искомый процент точек = ',p:5:1); {p по обр9}
writeln(fout,' Среднее удаление = ',sr:8:5); {sr по обр10}

```

```

CloseFile(fin); CloseFile(fout);
write('Press Enter...'); readln; {задерживаем экран до нажатия ENTER}
end.

```