

Практическое занятие 5. Работа с матрицами

- Задание:** рассмотреть 1) пример ввода-вывода матрицы: таблица данных, формы ввода/вывода, код;
2) обработку исключительных ситуаций, возникающих при работе с файлами;
3) схему решения задачи из Лабораторной работы № 9

Часть 1. Пример ввода-вывода матрицы

1. Задача. Задана матрица a из n строк и m столбцов. Здесь нас интересует только ввод-вывод матрицы, поэтому все остальные элементы задачи и ее решения опускаем.

4. Таблица данных

Класс	Имя	Описание (смысл), диапазон, точность	Тип	Структура	Формат
Входные данные	a	исходная матрица, $ a_{ij} < 100$, точн. 0.1	вещ	двухмерный массив (10×10)	+XX.X (:5:1)
	n	число строк матрицы а, $0 < n \leq 10$	цел	простая переменная	XX (:2)
	m	число столбцов матрицы а, $0 < m \leq 10$	цел	простая переменная	XX (:2)
Выходные данные					
Промежуточные	i	индекс строки текущего элемента, $0 < i \leq 10$	цел	простая переменная	
	j	индекс столбца текущего элемента, $0 < j \leq 10$	цел	простая переменная	
	dat	входной файл vvod_dat<№теста>.txt	файл	Текстовый ф	
	res	выходной файл vvod_res<№теста>.txt	файл	Текстовый ф	

5. Входная форма.

(Файлы vvod_dat<№ теста>.txt)

```
<n> <m>
<a[1,1]> <a[1,2]> . . . <a[1,m]>
<a[2,1]> <a[2,2]> . . . <a[2,m]>
. . . . .
<a[n,1]> <a[n,2]> . . . <a[n,m]>
```

6. Выходная форма (Файлы vvod_res<№ теста>.txt)

```
Обр1 <20 пробелов> Обработка матрицы
Обр2 Исходная матрица из <n>строк и<m> столбцов:
Обр3 <a[1,1]> <a[1,2]> . . . <a[1,m]>
      <a[2,1]> <a[2,2]> . . . <a[2,m]>
      . . . . .
      <a[n,1]> <a[n,2]> . . . <a[n,m]>
Обр4 Ошибка при открытии файла с исходными данными
Обр5 Ошибка при создании выходного файла
Обр6 Ошибка при считывании или записи в файл
```

Кодирование на Паскале

Пусть имена входного и выходного файлов задаются как первый и второй параметры командной строки;

```
program vvod;
{$APPTYPE CONSOLE}
```

```
Uses SysUtils; //обязательно для работы с исключениями
```

```
const
```

```
  nmax=10;
```

```
  mmax=10;
```

```
var
```

```
  n, m, i, j: byte;
```

```
  a: array[1..nmax,1..mmax] of real;
```

```
  dat, res: textFile;
```

```
begin
```

```
  assignFile(dat, paramstr(1)); assignFile(res, paramstr(2));
```

```
  try // попытаться открыть {1}
```

```
    reset(dat);
```

```
    try // не забыть закрыть в любом случае, раз открыли {2}
```

```
      try // попытаться открыть/создать {3}
```

```
        rewrite(res);
```

```
        try // не забыть закрыть в любом случае, раз открыли {4}
```

```
          try // ошибки при вводе/выводе в файл {5}
```

```
      writeln(res, ':20, 'Обработка матрицы'); { вывод по обр1}
```

```
      readln(dat, n, m); {ввод n, m}
```

```
      {ввод по строчно элементов матрицы}
```

```
      for i:=1 to n do
```

```
        begin
```

```
          for j:=1 to m do read(dat, a[i,j]); {ввод из одной строки}
```

```
          readln(dat); {переход на след.строку в файле}
```

```
        end;
```

```
      writeln(res, ' Исходная матрица из ',n:2,' строк и ',m:2,' столбцов'); {вывод n,m по обр2}
```

```
      {вывод по строкам элементов матрицы}
```

```
      for i:=1 to n do
```

```
        begin
```

```
          for j:=1 to m do write(res, a[i,j]:5:1, ' '); {вывод в одну строку: элемент с указанием формата и разделитель (пробел)}
```

```
          writeln(res); {переход на след.строку в файле}
```

```
        end;
```

```
      write('Ok. File ',paramstr(2),' is ready.');
```

```
      Except writeln('Ошибка при считывании или записи в файл') End; {5}
```

```
      Finally closeFile(res) End; {4}
```

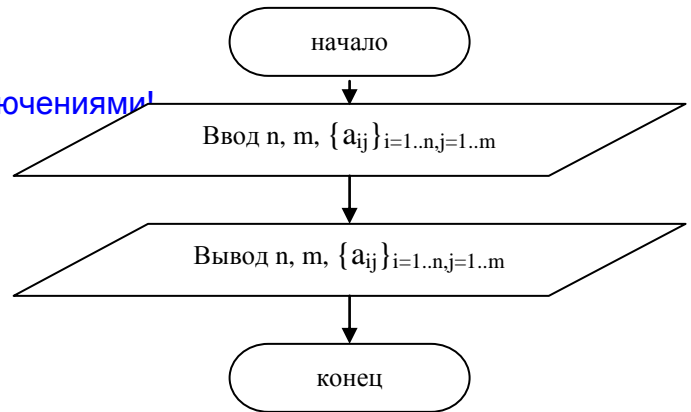
```
      Except writeln('Ошибка при создании выходного файла ', paramstr(2)) End; {3}
```

```
      Finally closeFile(dat) End; {2}
```

```
      Except writeln('Ошибка при открытии файла с исходными данными ', paramstr(1)) End; {1}
```

```
      write('Press ENTER'); readln;
```

```
end.
```



Часть 2. Пояснения к задачам из лабораторной работы №9

◆ Структура задач

Задана матрица a из n строк и m столбцов.

Для совокупности строк (столбцов) решается задача типа 1) или 2) из занятия 5 (поиск элемента на множестве – файл VseOdin.doc). Подзадачей этой задачи является задача типа 1) или 2) для совокупности элементов строки (столбца).

◆ Уточнения

•• Если в некоторой подзадаче задана матрица, то заданы:

- размеры
- элементы матрицы.

То есть входные данные: n , m , a (будем указывать имя массива a не $\{a_{ij}\}_{i=1..n,j=1..m}$).

•• Если задана строка матрицы (пусть i – номер строки), то входные данные:

- номер строки
- длина строки (количество столбцов)
- элементы матрицы.

То есть входные данные: i , m , a .

•• Если задан столбец матрицы (пусть j – номер столбца), то входные данные:

- номер столбца
- высота столбца (количество строк)
- элементы матрицы.

То есть входные данные: j , n , a .

◆ Общий метод (схема) решения задачи $m.0$ (из Лабораторной работы №9)

Задача $m.0$.

Абстракция $A0$. Спецификация

1. Условие

Условие для элементов строки

Найдем в матрице первую строку, все элементы которой удовлетворяют некоторому условию.

Условие для строки как единого целого

2. Уточненное условие

Дано....

Найти..., в случае отсутствия ...

3. Пример

Сделайте два примера

Например: $n=5$, $m=4$

1) Условие выполняется: Матрица, где есть искомая строка, лучше несколько строк:

<матрица из 5 строк и 4 столбцов с соответствующими значениями элементов>

Номер такой первой строки, где выполняется условие = <значение номера>

2) Условие не выполняется: Матрица, где нет искомой строки

Эти примеры затем включить в тесты.

4. Таблица данных

Класс	Имя	Описание (смысл), диапазон, точность	Тип	Структура	Формат
Входные данные	a	исходная матрица, . . .		Двухмерный массив (10×10)	
	n	число строк матрицы a, . . .			
	m	число столбцов матрицы a, . . .			
Выходные данные	k	номер искомой строки, . . .			
Промежуточные	i	индекс строки текущего элемента, . . .			
	j	индекс столбца текущего элемента, . . .			
	dat	входной файл			
	res	выходной файл			
	y1	результат проверки			

Диапазоны, типы, точность, структуру, формат заполнить самим.

5. Входная форма

см предыдущую задачу (ввод-вывод матрицы)

6. Выходная форма

см предыдущую задачу (ввод-вывод матрицы), плюс вывод результатов и/или сообщения о наличии или отсутствии искомой строки.

7. Аномальные ситуации

не анализируем

8. Тесты

Сделать следующие тесты (2 из примеров):

- *есть несколько искомых строк, результат – первая или последняя из них*
- *все строки подходят*
- *нет подходящих строк*

9. Метод

Отделим ввод-вывод от обработки

10. Алгоритм

Блок-схема задачи A0, с отделением ввода-вывода от обработки (подзадачи A0.1)

11. Программный код

Программный код с заглушкой для A0.1

```

Program m0;
{$APPTYPE CONSOLE}
const . . .
var . . .
begin
. . .
  {Абстракция A0.1. вх. N,m,A; Вых. y1, k}
. . .
end.

```

Подзадача А0.1. Спецификация**1,2.Условие.****Сами**

(тоже что у А0, но вместо вывода сообщения, присвоить результат проверки переменной у1)

3.Пример.

те же (А0)

4. Таблица данных

Сами

та же (А0), кроме *res, dat*

и дополнительно:

пром. лог у2 – результат проверки строки, Истина, если ..., Ложь, если...

5,6.Формы.

не нужны. Ввода-вывода нет

7. Аномалии входных данных

ввода нет

8. Тесты

те же (А0)

9. Метод

- Проведем аналогию с задачами 1) и 2) Практического занятия 3 для матрицы как совокупности строк.
Одномерный массив – совокупность *элементов*
Матрица – совокупность *строк*

Значит, искать отдельную строку матрицы, удовлетворяющую условию данной задачи, можно так же, как искать элемент одномерного массива, удовлетворяющий условию задачи занятия 3.

Для совокупности строк решаем задачу типа 1). (Абстракция А0.1)

Для фиксации результата проверки совокупности строк используем логическую переменную:

$$\text{лог } y1 - \text{результат проверки;} \quad y1 = \begin{cases} \text{истина, если искомая строка есть,} \\ \text{ложь, в противном случае;} \end{cases}$$

- Проведем аналогию с задачами 1 и 2 занятия 3 для отдельной строки матрицы.

Одномерный массив – совокупность элементов с *одним индексом*, изменяющимся при просмотре элементов.

Строка матрицы – совокупность элементов с *индексом столбца*, изменяющимся при просмотре элементов, и фиксированным индексом строки.

Проверять элементы отдельной строки можно так же, как проверять элементы одномерного массива, изменяя не фиксированный индекс.

Для отдельной строки решаем задачу типа 2). (Абстракция А0.1.1)

Для фиксации результата проверки совокупности элементов отдельной строки используем логическую переменную:

$$\text{лог } y2 - \text{результат проверки;} \quad y2 = \begin{cases} \text{истина, если все элементы строки удовлетв. условию,} \\ \text{ложь, в противном случае;} \end{cases}$$

Абстрагируясь от подзадачи проверки строки, получаем:

цел i – текущий номер строки.

Положим сначала, что искомая строка не найдена, т.е. $y1 := \text{ложь}$.

Начнем с первой строки ($i:=1$)

Просматриваем матрицу по строкам в цикле ПОКА, пока строка не найдена ($y1=\text{ложь}$) и строки не исчерпаны ($i \leq m$):

Решаем подзадачу (A0.1.1) проверки строки (Вх. i, m, a ; Вых. $y2$)

анализ результата проверки ($y2$) и формирование $y1$ и k :

если строка удовл. условию ($y2 = \text{истина}$), то

$y1 := \text{истина}$; (искомая строка найдена)

$k := i$; {запоминаем ее номер}

Номер следующей строки $i:=i+1$

Конец просмотра строк

10. Алгоритм

блок-схема для A1 с выделением подзадачи A0.1.1

11. Программный код

Добавляем описание переменной $y2$:

Var
 $y2$: Boolean;

Программный код, пригодный для вставки в основной программный код (A0) вместо заглушки, В свою очередь с заглушкой для подзадачи A0.1.1

Подзадача A1.1. Спецификация

1.2. Условие.

Для i -й строки матрицы a , состоящей из m элементов, проверить условие...

3. Пример.

отдельный - не нужен (разные строки должны быть в примере для A0, пояснения по строкам там же)

4. Таблица данных

сделать таблицу:

Вх. i (номер строки), m (длина строки), a (матрица);

Вых. $y2$ (результат проверки: Истина, если..., Ложь, если...)

Пром. j – текущий номер элемента в строке

5,6. Формы.

не нужны. Ввода-вывода нет

7. Аномалии входных данных

ввода нет

8. Тесты

не нужны

9. Метод

Сами (аналогичен методу в задачах 1 или 2 из Практического занятия 3.)

10. Алгоритм

блок-схема для А1.1

11. Программный код

Программный код, пригодный для вставки в основной программный код подзадачи А0.1 вместо заглушки

Часть 3. Пример решения задачи m0 (Лабораторная работа №9)

Задача m.0.

Абстракция А0. Спецификация

1. Условие

Условие для элементов столбца

Проверить, все ли столбцы целочисленной матрицы содержат хотя бы один элемент с нечетным значением, если не все, то сообщить номер последнего столбца, в котором нет такого элемента

Условие для столбца как единого целого

2. Уточненное условие

Дана целочисленная квадратная матрица A , состоящая из N строк и M столбцов.

Проверить ($y1$), все ли столбцы целочисленной матрицы A содержат хотя бы один элемент с нечетным значением, если не все, то сообщить номер (k) последнего столбца, в котором нет такого элемента.

3. Примеры

Пример 1) во всех столбцах есть нечетное число

$n=5, m=4$

A

<u>1</u>	2	<u>3</u>	4
2	2	2	4
2	2	2	<u>7</u>
2	<u>5</u>	2	<u>1</u>
<u>3</u>	2	2	4

Условие выполняется ($y1=true$)

Пример 2) (Построен на основе примера 1) НЕ во всех столбцах есть нечетное число

$n=5, m=4$

A

<u>4</u>	2	<u>6</u>	4
2	2	2	4
2	2	2	<u>7</u>
2	<u>5</u>	2	<u>1</u>
<u>4</u>	2	2	4

Условие не выполняется для столбцов 1 и 3 ($y1=false, k=3$)

4. Таблица данных

Класс	Имя	Описание (смысл), диапазон, точность	Тип	Структура	Формат
Входные данные	a	исходная матрица, $ a_{ij} < 100$	цел	Двумерный массив (10×10)	+XX (:3)
	n	число строк матрицы a, $0 < n \leq 10$	цел	Простая	XX (:2)
	m	число столбцов матрицы a, $0 < m \leq 10$	цел	Простая	XX (:2)
Выходные данные	y1	результат проверки, true, если все столбцы удовлетворяют условию, false, если не все	лог	Простая	
	k	номер последнего столбца, который не удовлетворяет условию, $0 < k \leq 10$	цел	Простая	XX (:2)
Промежуточные	i	индекс строки текущего элемента, $0 < i \leq 10$	цел	Простая	
	j	индекс столбца текущего элемента, $0 < j \leq 10$	цел	Простая	
	dat	входной файл m0_dat<№>.txt	Текстовый	Файл	
	res	выходной файл m0_res<№>.txt	Текстовый	Файл	

5. Входная форма (файлы m0_dat<№>.txt)

Сами (см предыдущую задачу «ввод-вывод матрицы»)

6. Выходная форма (файлы m0_res<№>.txt)

Обр1	Обработка матрицы (m0.№)
Обр2	Исходная матрица из <n>строк и<m> столбцов:
Обр3	<pre> <a [1, 1]> <a [1, 2]> . . . <a [1, m]> <a [2, 1]> <a [2, 2]> . . . <a [2, m]> <a [n, 1]> <a [n, 2]> . . . <a [n, m]> </pre>
Обр4	Условие выполняется для всех столбцов
Обр5	Условие выполняется НЕ для всех столбцов Последний такой столбец имеет номер <k>

7. Аномальные ситуации

не анализируем пока

8. Тесты

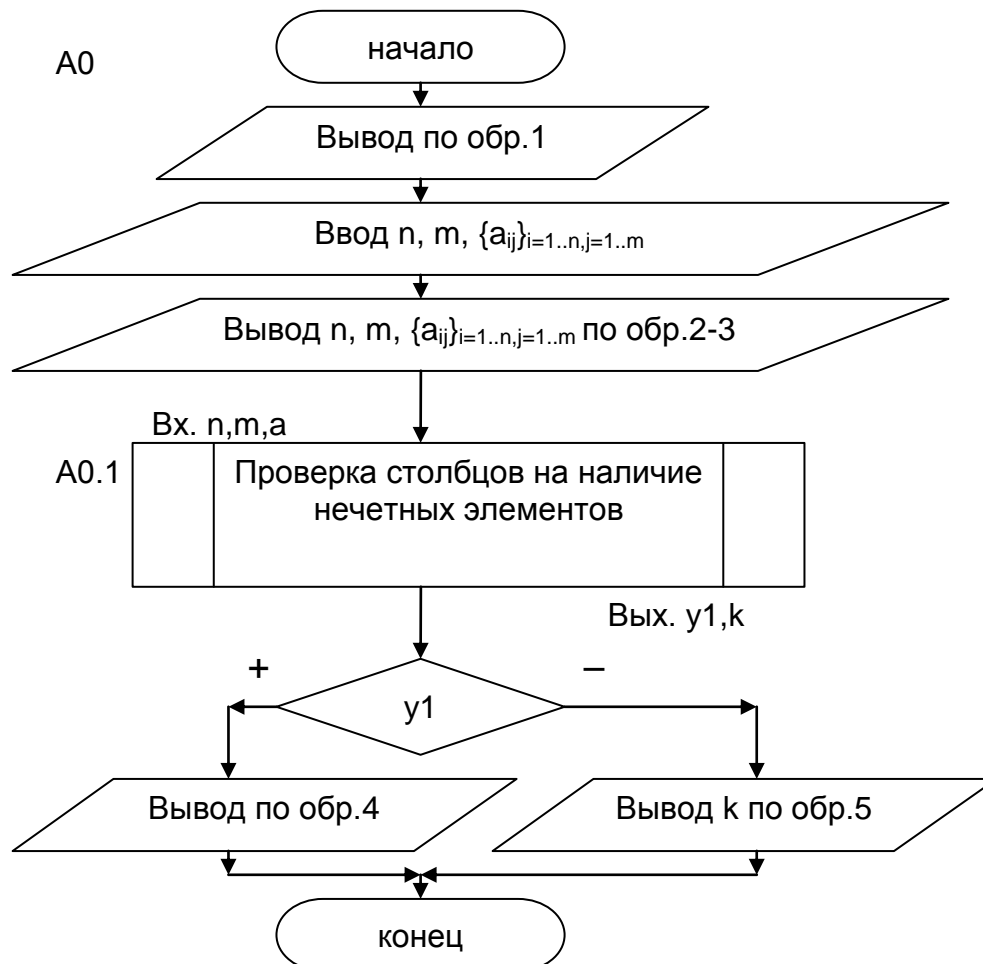
№ теста	Входные данные	Ожидаемый результат	Смысл теста																				
1	n=5; m=4; A <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td><u>1</u></td><td>2</td><td><u>3</u></td><td>4</td></tr> <tr><td>2</td><td>2</td><td>2</td><td>4</td></tr> <tr><td>2</td><td>2</td><td>2</td><td><u>7</u></td></tr> <tr><td>2</td><td><u>5</u></td><td>2</td><td><u>1</u></td></tr> <tr><td><u>3</u></td><td>2</td><td>2</td><td>4</td></tr> </table>	<u>1</u>	2	<u>3</u>	4	2	2	2	4	2	2	2	<u>7</u>	2	<u>5</u>	2	<u>1</u>	<u>3</u>	2	2	4	Y1=true (Обр 4)	Все столбцы подходят, данные взяты из примера 1
<u>1</u>	2	<u>3</u>	4																				
2	2	2	4																				
2	2	2	<u>7</u>																				
2	<u>5</u>	2	<u>1</u>																				
<u>3</u>	2	2	4																				
2	n=5; m=4; A <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td><u>4</u></td><td>2</td><td><u>6</u></td><td>4</td></tr> <tr><td>2</td><td>2</td><td>2</td><td>4</td></tr> <tr><td>2</td><td>2</td><td>2</td><td><u>7</u></td></tr> <tr><td>2</td><td><u>5</u></td><td>2</td><td><u>1</u></td></tr> <tr><td><u>4</u></td><td>2</td><td>2</td><td>4</td></tr> </table>	<u>4</u>	2	<u>6</u>	4	2	2	2	4	2	2	2	<u>7</u>	2	<u>5</u>	2	<u>1</u>	<u>4</u>	2	2	4	Y1=false K=3 (из 1 и 3) (Обр 5)	Не все столбцы подходят, есть два неподходящих; данные взяты из примера 2
<u>4</u>	2	<u>6</u>	4																				
2	2	2	4																				
2	2	2	<u>7</u>																				
2	<u>5</u>	2	<u>1</u>																				
<u>4</u>	2	2	4																				
3	n=2; m=2; A <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>2</td><td>6</td></tr> <tr><td>4</td><td>8</td></tr> </table>	2	6	4	8	Y1=false K=2 (из 1 и 2) (Обр 5)	Все столбцы НЕ подходят																
2	6																						
4	8																						

9. Метод

Отделим ввод-вывод от обработки

10. Алгоритм

Блок-схема задачи A0, с отделением ввода-вывода от обработки (подзадачи A0.1):



11. Программный код

Программный код с заглушкой для А0.1

```

Program m0;
{$APPTYPE CONSOLE}
const
  nmax=10;
  mmax=10;
var
  n, m, i, j, k: byte;
  a: array[1..nmax,1..mmax] of real;
  y1: Boolean;
  dat, res: textFile;

<другие описания>

begin
  assignFile(dat, paramstr(1)); assignFile(res, paramstr(2));
  reset(dat); rewrite(res);

  writeln(res, ' ':15,'Обработка матрицы (m0.0)');           { вывод по обр1}

<ввод-вывод исходных данных см. задачу о вводе-выводе >

  {Абстракция А0.1. вх. n,m,A; Вых. y1, k}
  {Для теста 1} y1:=True;
  {Для теста 2} //y1:=False; k:=3;
  {Для теста 3} //y1:= False; k:=2;

  if y1 then
    writeln(res, ' Условие выполняется для всех столбцов ')
  else
    begin
      writeln(res, 'Условие выполняется НЕ для всех столбцов ');
      writeln(res, 'Последний такой столбец имеет номер ', k:2);
    end;

  CloseFile(dat); CloseFile(res);
end.

```

Подзадача А0.1. Спецификация**1. Условие.**

Переменной $y1$ присвоить значение $TRUE$, если все столбцы целочисленной матрицы содержат хотя бы один элемент с нечетным значением, или $FALSE$ – в противном случае. Найти также номер последнего столбца, в котором нет элемента с нечетным значением.

2. Уточненное условие

Дана целочисленная квадратная матрица A , состоящая из N строк и M столбцов.

Переменной $y1$ присвоить значение $TRUE$, если все столбцы целочисленной матрицы содержат хотя бы один элемент с нечетным значением, или $FALSE$ – в противном случае. Найти также номер (k) последнего столбца, в котором нет элемента с нечетным значением.

3. Пример.

те же (А0)

4. Таблица данных

Класс	Имя	Описание (смысл), диапазон, точность	Тип	Структура	Формат
Входные данные	a	исходная матрица, $ a_{ij} < 100$	цел	Двумерный массив (10×10)	
	n	число строк матрицы a, $0 < n \leq 10$	цел	Простая	
	m	число столбцов матрицы a, $0 < m \leq 10$	цел	Простая	
Выходные данные	y1	результат проверки, true, если все столбцы удовлетворяют условию, false, если не все	лог	Простая	
	k	номер последнего столбца, который не удовлетворяет условию, $0 < k \leq 10$	цел	Простая	
Промежуточные	i	индекс строки текущего элемента, $0 < i \leq 10$	цел	Простая	
	j	индекс столбца текущего элемента, $0 < j \leq 10$	цел	Простая	
	y2	результат проверки столбца, true, если есть элемент с нечетным значением, false, если нет	лог	Простая	

та же (А0), кроме res, dat и дополнительно: пром. лог y2

5.6. Формы.

не нужны. Ввода-вывода нет

7. Аномалии входных данных

ввода нет

8. Тесты

те же (А0)

9. Метод

Для совокупности строк решаем задачу типа 2). (Абстракция А0.1)

Для фиксации результата проверки совокупности строк используем логическую переменную:

$$\text{лог } y_1 \text{ – результат проверки; } y_1 = \begin{cases} \text{истина, если все столбцы удовл. условию;} \\ \text{ложь, в противном случае;} \end{cases}$$

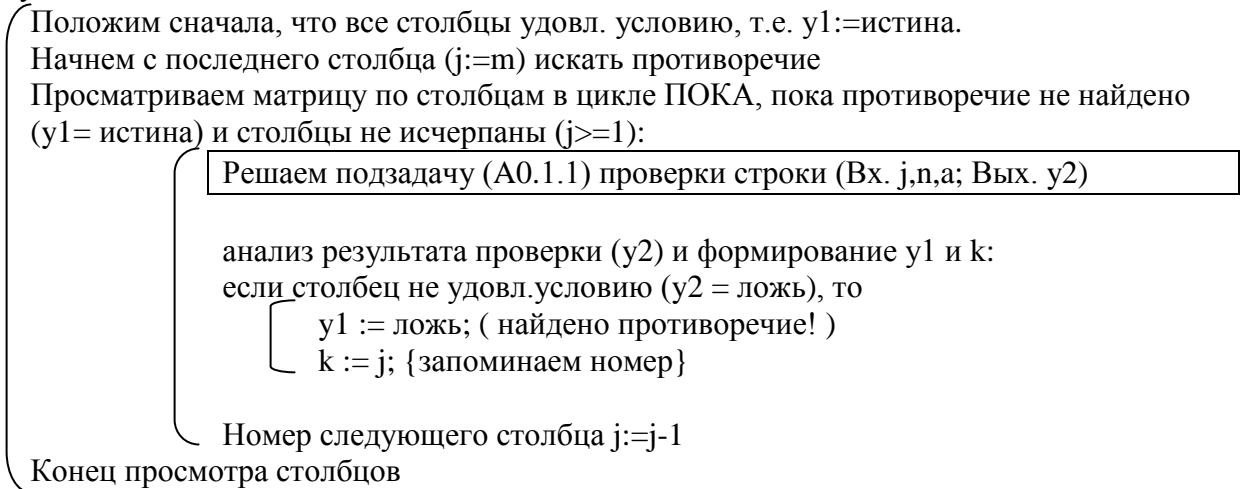
Для отдельного столбца решаем задачу типа 1. (Абстракция А0.1.1)

Для фиксации результата проверки совокупности элементов отдельного столбца используем логическую переменную:

$$\text{лог } y_2 \text{ – результат проверки; } y_2 = \begin{cases} \text{истина, если есть элемент с нечетным значением,} \\ \text{ложь, в противном случае;} \end{cases}$$

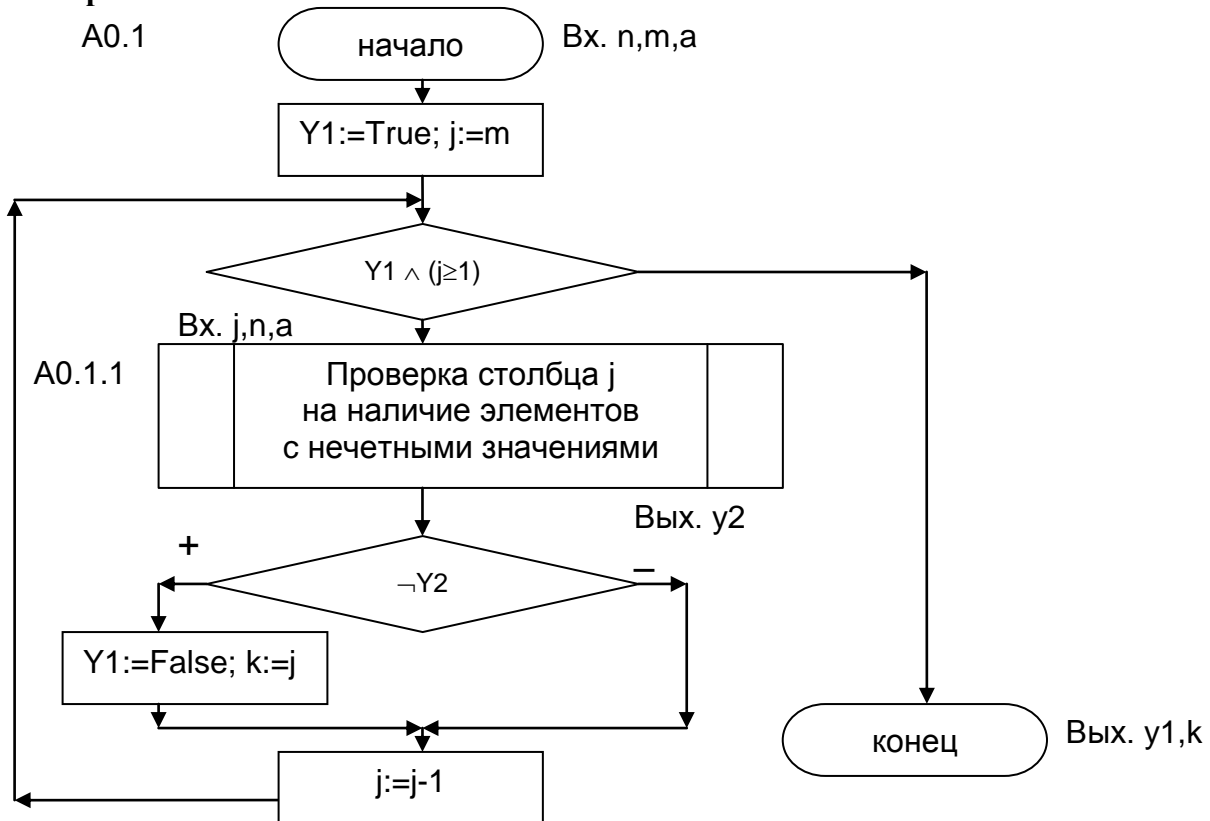
Абстрагируясь от подзадачи проверки столбца, получаем:

цел j – номер текущего столбца.



10. Алгоритм

А0.1



11. Программный код (Всю программу не надо – в ней указано место А0.1, сейчас там заглушка)

Добавляем описание переменной y2:

```
Var
  y2: Boolean;
```

Программный код, пригодный для вставки в основной программный код (А0) вместо заглушки, В свою очередь с заглушкой для подзадачи А0.1.1:

```
y1:=true;
j:=m;
while y1 and (j>=1) do
begin
    {Абстракция А0.1.1 вх. j,n,A; Вых. y2}
    {Для теста 1} y2:=True;
    {Для теста 2} { case j of
                    1,3: y2:=False;
                    2,4: y2:=True;
                    end;}
    {Для теста 3} //y2:= False;

    if not y2 then
    begin
        y1:=False; k:=j;
    end;
    dec(j);
end;
```

Подзадача А1.1. Спецификация

1. Условие.

В j-ом столбце матрицы a, состоящем из n элементов, проверить наличие нечетного элемента.

2. Уточненное условие

Дана целочисленная матрица A.

Присвоить переменной y2 значение TRUE, если в j-ом столбце матрицы a, состоящем из n элементов есть хотя бы один элемент с нечетным значением, и FALSE – в противном случае.

3. Примеры

те же. См А0

4. Таблица данных

Класс	Имя	Описание (смысл), диапазон, точность	Тип	Структура
Входные данные	a	исходная матрица, $ a_{ij} < 100$	цел	Двумерный массив (10×10)
	n	число элементов в столбце a, $0 < n \leq 10$	цел	Простая
	j	Номер столбца матрицы a, $0 < m \leq 10$	цел	Простая
Выходные данные	y2	результат проверки, true, если есть элемент с нечетным значением, false, если нет	лог	Простая
Промежуточные	i	индекс текущего элемента в столбце, $0 < i \leq 10$	цел	Простая

5.6.Формы.

не нужны. Ввода-вывода нет

7. Аномалии входных данных

ввода нет

8. Тесты

см. А0

9. Метод

Для отдельного столбца решаем задачу типа 1. (Абстракция А0.1.1)

Для фиксации результата проверки совокупности элементов отдельного столбца используем логическую переменную:

лог у2 – результат проверки; $y2 = \begin{cases} \text{истина, если есть элемент с нечетным значением,} \\ \text{ложь, в противном случае;} \end{cases}$

Абстрагируясь от подзадачи проверки столбца, получаем:

цел i – номер текущего элемента в столбце j .

Положим сначала, что нет элемента, удовл. условию, т.е. $y2 := \text{ложь}$.

Начнем с первого элемента ($i := 1$)

Просматриваем столбец j в цикле ПОКА, пока не нашли нечетный элемент ($y2 = \text{ложь}$) и элементы столбца не исчерпаны ($i \leq n$):

если i -ый элемент нечетный ($\text{odd}(a[i,j] = \text{true})$), то

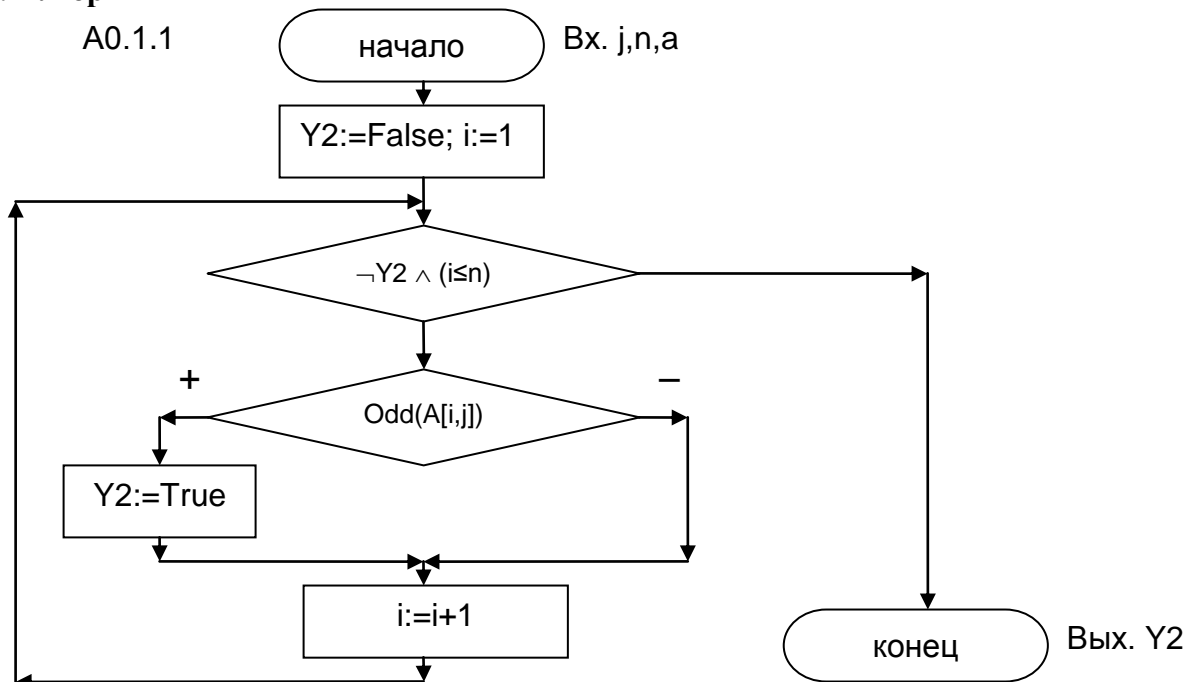
$y2 := \text{true}$; (нашли!)

Номер следующего элемента $i := i + 1$

Конец просмотра элементов столбца j

10. Алгоритм

А0.1.1



11. Программный код

Программный код, пригодный для вставки в основной программный код подзадачи A0.1 вместо заглушки

```
y2:=False;
i:=1;
while not y2 and (i<=n) do
begin
  if Odd(a[i,j]) then // если нечетный
  begin
    y2:=True
  end;
  inc(i);
end;
```