

1 курс

**ПЛАН – КОНСПЕКТ**  
проведения вводного занятия по теме 3.5  
по дисциплине «Информатика»

**Раздел 3. «Информационное моделирование.»**

**Тема 3.5:**  
**«Анализ алгоритмов в профессиональной области»**

Подготовил: преподаватель  
В.Н. Борисов

Рязань 2024

## **Тема №3.5. «Понятие алгоритма и основные алгоритмические структуры»**

**Цели занятий:** изучить со студентами основные сведения о структурированных типах данных, массивах, вспомогательных алгоритмах, рассмотреть задачи поиска элемента с заданными свойствами, анализ типовых алгоритмов обработки чисел, числовых последовательностей, массивов, построение алгоритма задачи отправления поезда со станции формирования.

**Виды занятий:** классно-групповые, комбинированные (по проверке знаний, умений по пройденному материалу, по изучению и первичному закреплению на практике нового материала).

**Метод проведения занятий:** лекционные занятия.

### **Основные вопросы:**

1. Структурированные типы данных.
2. Массивы. Описание переменной типа массив. Базовый тип массива.
3. Ввод и вывод элементов массива.
4. Действия над массивами и над элементами массива.
5. Задачи поиска элемента с заданными свойствами. Поиск оптимального элемента.
6. Формирование нового массива.
7. Сортировка массива линейным методом и методом пузырька.
8. Проверка упорядоченности.
9. Составление программы для вычислений в одномерном массиве.
10. Двумерные массивы. Матрицы.
11. Операции над строками и столбцами в двумерном массиве.
12. Составление программы для вычислений в двумерном массиве.
13. Вспомогательные алгоритмы. Подпрограммы.
14. Примеры использования стандартных процедур и функций в программах.
15. Пользовательские функции. Формат объявления пользовательской функции.
16. Пользовательские процедуры. Формат объявления пользовательской процедуры.
17. Составление программ с использованием пользовательских функций.
18. Составление программ с использованием пользовательских процедур.
19. Анализ типовых алгоритмов обработки чисел, числовых последовательностей и массивов.
20. Построение алгоритма задачи отправления поезда со станции формирования.

### Литература:

1. [5 учебник раздела «Дополнительной учебной литературы» рабочей программы изучения дисциплины]: Трофимов, В. В. Информатика в 2 т. Том 2 : учебник для среднего профессионального образования / В. В. Трофимов. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 406 с. — (Профессиональное образование). — ISBN 978-5-534-02519-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/513266>, глава 25;
2. 5 учебник раздела «Основной учебной литературы» рабочей программы изучения дисциплины: Босова, Л. Л. Информатика. 11 класс. Базовый уровень : учебник / Л.Л. Босова, А. Ю. Босова. — М. : БИНОМ. Лаборатория знаний, 2022. — 200 с. , ISBN 978-5-9963-3142-0, § 8-9 главы 2.

### Примерный расчет времени (по каждому практическому занятию):

1. Вступительная часть – 20 мин.
2. Основная часть – 60 мин.
3. Заключительная часть – 10 мин.

### Вступительная часть:

Занятия начать с объявления темы занятия, основных рассматриваемых вопросов, времени изучения темы (нового материала), перечисление литературы, проведения опроса по изученному ранее (пройденному) материалу.

### Основная часть (по каждому занятию):

#### Первый вопрос: Структурированные типы данных.

**Структурированными данными** называют такие **данные**, которые состоят из других **типов данных**.

**Структурированные типы данных** определяют наборы однотипных или разнотипных компонент.

К ним относятся, в частности (в языке прог-я Pascal), переменные **типов**:

- Array - массивы (табличные **данные**);
- String – строковые величины;
- Record – **данные типа** запись;
- Set – **данные типа** множество;
- File – **данные типа** файл.

## Структурированные типы данных

Структурированные типы данных определяют упорядоченную совокупность скалярных переменных и характеризуются типом своих компонентов.

Структурированные типы данных в отличие от простых задают множества сложных значений с одним общим именем.

Существует несколько методов структурирования, каждый из которых отличается способом обращения к отдельным компонентам и, следовательно, способом обозначения компонентов, входящих в структурные данные.

По способу организации и типу компонентов в сложных типах данных выделяют следующие разновидности:

- регулярный тип (массивы);
- комбинированный тип (записи);
- файловый тип (файлы);
- множественный тип (множества);
- строковый тип (строки);
- в языке Турбо Паскаль версии 6.0 и старше введен объектный тип (объекты).

Также сведения по данному вопросу представлены в приложении № 1.

### **Второй вопрос: Массивы. Описание переменной типа массив. Базовый тип массива.**

Сведения по данному вопросу представлены в приложениях № 1, 2.

### **Третий вопрос: Ввод и вывод элементов массива.**

Сведения по данному вопросу представлены в приложении № 2.

### **Четвёртый вопрос: Действия над массивами и над элементами массива.**

Сведения по данному вопросу представлены в приложениях № 1, 3.

### **Пятый вопрос: Задачи поиска элемента с заданными свойствами. Поиск оптимального элемента.**

Сведения по данному вопросу представлены в приложениях № 1, 3.

**Шестой вопрос: Формирование нового массива.**

Сведения по данному вопросу представлены в приложении № 4.

**Седьмой вопрос: Сортировка массива линейным методом и методом пузырька.**

Сведения по данному вопросу представлены в приложении № 4.

**Восьмой вопрос: Проверка упорядоченности.**

Сведения по данному вопросу представлены в приложении № 4.

**Девятый вопрос: Составление программы для вычислений в одномерном массиве.**

Сведения по данному вопросу представлены в приложении № 4.

**Десятый вопрос: Двумерные массивы. Матрицы.**

Сведения по данному вопросу представлены в приложении № 5.

**Одиннадцатый вопрос: Операции над строками и столбцами в двумерном массиве.**

Сведения по данному вопросу представлены в приложении № 5.

**Двенадцатый вопрос: Составление программы для вычислений в двумерном массиве.**

Сведения по данному вопросу представлены в приложении № 5.

**13 вопрос: Вспомогательные алгоритмы. Подпрограммы.**

Сведения по данному вопросу представлены в приложениях № 1, 6.

**14 вопрос: Примеры использования стандартных процедур и функций в программах.**

Сведения по данному вопросу представлены в приложениях № 1, 6.

**15 вопрос: Пользовательские функции. Формат объявления пользовательской функции.**

Сведения по данному вопросу представлены в приложениях № 1, 6.

**16 вопрос: Пользовательские процедуры. Формат объявления пользовательской процедуры.**

Сведения по данному вопросу представлены в приложениях № 1, 6.

**17 вопрос: Составление программ с использованием пользовательских функций.**

Сведения по данному вопросу представлены в приложениях № 1, 6.

**18 вопрос: Составление программ с использованием пользовательских процедур.**

Сведения по данному вопросу представлены в приложениях № 1, 6.

**19 вопрос: Анализ типовых алгоритмов обработки чисел, числовых последовательностей и массивов.**

**Анализ типовых алгоритмов обработки чисел, числовых последовательностей.**

**Генерация случайных чисел на заданном промежутке [a;b]**

x



*Randomize;*  
*x:= random(b - a) + a;*

**Перестановка двух переменных с использованием дополнительной ячейки**

*Z:=X;*  
*X:=Y;*  
*Y:=Z;*

## Разбиение числа известной длины на цифры

$a := 3862;$

$$S := a \bmod 10 + \underbrace{(a \operatorname{div} 10) \bmod 10}_{\text{Отброс 2}} + \underbrace{(a \operatorname{div} 100) \bmod 10}_{\text{Взять 6}} + \underbrace{(a \operatorname{div} 1000)}_{\text{Взять 8}};$$

Взять 2                      Взять 6                      Взять 8

$$+ a \operatorname{div} 1000;$$

3

## Принадлежит ли число $x$ заданному интервалу $[a, b]$

If  $(x > a)$  and  $(x < b)$  then

write ('yes')

else

write ('no');

If  $(x < a)$  or  $(x > b)$  then

write ('no')

else

write ('yes');



## Поиск наибольшего (наименьшего) из 3-х чисел

1. If  $(a > b)$  and  $(a > c)$  then

$max := a$

else

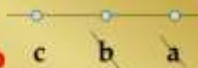
if  $b > c$

then

$max := b$

else

$max := c;$



## Вычисление факториала числа $n$

**Факториал** – это произведение всех натуральных чисел от 1 до  $n$ .

Обозначается  $n!$

$$n! = (n-1)! * n, \quad 0! = 1, \quad 1! = 1$$

Реализация в программе:

...

$f := 1;$

For  $i := 1$  to  $n$  do

$f := f * i;$

## Числа Фибоначчи

Последовательность чисел

1, 1, 2, 3, 5, 8, 13, 21...

называется числами Фибоначчи.

$$f_1 = 1, \quad f_2 = 1, \quad f_3 = f_1 + f_2, \quad f_i = f_{i-2} + f_{i-1}$$

Реализация на Паскале:

...

read( $n$ );    [количество чисел в последовательности]

$f_1 := 1; \quad f_2 := 1; \quad \text{write}(f_1, f_2);$

for  $i := 3$  to  $n$  do

begin

$f := f_1 + f_2; \quad \text{write}(f);$

$f_1 := f_2;$

$f_2 := f;$

end;

## Поиск наибольшего (наименьшего) из 3-х чисел

2. If  $a > b$  then

$max := a$

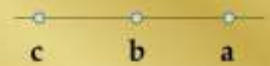
else

$max := b;$

if  $c > max$  then

$max := c;$

WriteLn('max=', max);



$max = b$



## Возведение числа в степень:

$X^n$

$X^n = X * X * X * \dots * X$  - n раз

Реализация в программе:

```
...
read (x,n);
st:=1;
for i:=1 to n do
    st:=st*x;
```

## Делители натурального числа n

2. Количество делителей (кроме 1 и самого числа):

```
...
read (n);
k:=2; {1,n}
for d:=2 to n div 2 do
    if n mod d =0 then
        inc(k);
```

## Делители натурального числа n

1. Все делители:

```
...
read (n);
For d:=1 to n do
    if n mod d =0 then
        write('d=',d);
```

### Анализ типовых алгоритмов обработки массивов.

Сведения по данному вопросу представлены в текущем документе – План-конспекте при рассмотрении вопросов № 3, 4, 5, 6, 7, 8, 9, 11, 12.

**20 вопрос: Построение алгоритма задачи отправления поезда со станции формирования.**

Формулировка задачи:

Сформировать файл, содержащий сведения об отправлении поездов со станции: номер поезда, станция назначения, время отправления, наличие и стоимость билетов. Вывести на экран время отправления поезда X.



## Листинг программы

```

1. uses crt;
2. const n=3;
3.     zag='                               Расписание поездов';
4.     line='-----';
5.     shap='|  Номер  |  Станция назначения  |  Отправление  |  Билеты  |  Стоимость  |';
6.
7. type trains = record
8.     nom: string[8];
9.     naz: string[20];
10.    voc: byte;
11.    vom: byte;
12.    bil: char;
13.    zen: real;
14. end;
15.
16. var f:file of trains;
17.     t: trains;
18.     i,z:byte;
19.     s:string;
20.
21. begin
22. assign(f,'C:\trains.dta');
23. rewrite(f);
24. writeln('Внесите данные о поездах: ');
25. for i:=1 to n do
26.     with t do
27.         begin
28.             write('Номер ',i,' поезда: '); readln(nom);

```

Скопировать

```

29.             write('Назначение: '); readln(naz);
30.             repeat
31.                 write('Отправление(часы и минуты через пробел): '); read(voc,vom);
32.                 until (voc in [1..23]) and (vom in [1..59]);
33.                 repeat
34.                     write('Наличие билетов (0-нет, 1-есть): '); readln(z);
35.                     until z in [0..1];
36.                     if z=1 then bil:='+' else bil:='-';
37.                     write('Цена билета: '); readln(zen);
38.                     write(f,t);
39.                 end;
40.             close(f);
41.             clrscr;
42.             z:=1;
43.             repeat
44.                 writeln('0-завершение работы');
45.                 writeln('1-вывод расписания поездов из файла');
46.                 writeln('2-вывод времени отправления поезда по номеру');
47.                 readln(z);
48.                 if z=1 then
49.                     begin
50.                         clrscr;
51.                         writeln(zag);
52.                         writeln(line);
53.                         writeln(shap);
54.                         writeln(line);
55.                         reset(f);
56.                         while not eof(f) do
57.                             begin
58.                                 read(f,t);

```

```

59.     with t do
60.         writeln('|',nom:8,'|':2,naz:20,'|':3,voc:7,'|':4,bil:5,'|':4,zen:8:1,'|':4);
61.         writeln(line);
62.     end;
63. end;
64. if z=2 then
65.     begin
66.         write('Номер поезда: '); readln(s);
67.         reset(f);
68.         while not (eof(f)) do
69.             begin
70.                 read(f,t);
71.                 if t.nom=s then
72.                     writeln('Время отправления поезда ',t.voc:2,'|':1,t.vom:2);
73.                 end;
74.             end;
75.         until z=0;
76.         close(f);
77.         readln;
78.     end.

```

### Объяснение кода листинга программы

1. Заголовок программы и описание переменных:
  - uses crt;
  - const n=3;
  - zag=' Расписание поездов';
  - line='-----';
  - shar='| Номер | Станция назначения | Отправление | Билеты |  
Стоимость |';
  - trains = record
    - nom: string[8];
    - naz: string[20];
    - voc: byte;
    - vom: byte;
    - bil: char;
    - zen: real;
    - end;
  - var f: file of trains;
  - var t: trains;
  - var i,z: byte;
  - var s: string;
2. Ввод данных о поездах:
  - assign(f,'C:\trains.dta');
  - rewrite(f);
  - writeln('Внесите данные о поездах: ');
  - for i:=1 to n do
    - with t do
      - begin
      - write('Номер ',i,' поезда: '); readln(nom);

```

▪ write('Назначение: '); readln(naz);
▪ repeat
▪ write('Отправление(часы и минуты через пробел): ');
read(voc,vom);
▪ until (voc in [1..23]) and (vom in [1..59]);
▪ repeat
▪ write('Наличие билетов (0-нет, 1-есть): '); readln(z);
▪ until z in [0..1];
▪ if z=1 then bil:='+ ' else bil:='- ';
▪ write('Цена билета: '); readln(zen);
▪ write(f,t);
▪ end;

```

### 3. Вывод расписания поездов из файла:

```

○ clrscr;
○ z:=1;
○ repeat
▪ writeln('0-завершение работы');
▪ writeln('1-вывод расписания поездов из файла');
▪ writeln('2-вывод времени отправления поезда по номеру');
▪ readln(z);
▪ if z=1 then
▪ begin
▪ clrscr;
▪ writeln(zag);
▪ writeln(line);
▪ writeln(shap);
▪ writeln(line);
▪ reset(f);
▪ while not eof(f) do
▪ begin
▪ read(f,t);
▪ with t do
▪ writeln('|',nom:8,'|':2,naz:20,'|':3,voc:7,'|':vom:2,'|':
4,bil:5,'|':4,zen:8:1,'|':4);
▪ writeln(line);
▪ end;
▪ end;

```

### 4. Вывод времени отправления поезда по номеру:

```

○ if z=2 then
▪ begin
▪ write('Номер поезда: '); readln(s);
▪ reset(f);
▪ while not (eof(f)) do
▪ begin
▪ read(f,t);
▪ if t.nom=s then

```

- writeln('Время отправления поезда  
' ,t.voc:2,'!',t.vom:2);
  - end;
  - end;
5. Основной цикл программы:
- until z=0;
  - close(f);
  - readln;

### **Заключительная часть (по каждому занятию).**

1. Закончить изложение материала.
2. Ответить на возникшие вопросы.
3. Подвести итоги занятия.
4. Дать задание на самоподготовку (домашние задания).

### **Задание на самоподготовку (домашние задания):**

1. Детально проработать, законспектировать материал занятия, размещенный в данном план-конспекте, в учебниках, указанных на с.2 текущего документа.
2. Подготовиться к опросу по пройденному материалу, защите ранее выполненных практических работ.