

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Забайкальский государственный университет»
(ФГБОУ ВПО «ЗабГУ»)

Факультет естественных наук, математики и технологий
Кафедра информатики, теории и методике обучения информатике

УЧЕБНЫЕ МАТЕРИАЛЫ
для студентов заочной формы обучения

по дисциплине «Программирование»

для направления подготовки 44.03.01 «Педагогическое образование» про-
филь «Информатика и информационные технологии в образовании»

Общая трудоемкость дисциплины 468 часов, 13 зачетных единиц

| Виды занятий | Распределение по семестрам в часах | | | Всего часов |
|--|---------------------------------------|--------------|--------------|----------------|
| | 5 семестр | 6 семестр | 7 семестр | |
| 1 | 2 | 3 | 4 | 5 |
| Общая трудоемкость | 108 | 180 | 180 | 468 |
| Аудиторные занятия, в т.ч.: | 20 | 16 | 16 | 52 |
| лекционные (ЛК) | 10 | 8 | 8 | 26 |
| практические (семинарские) (ПЗ, СЗ) | - | - | - | - |
| лабораторные (ЛР) | 10 | 8 | 8 | 26 |
| Самостоятельная работа студентов (СРС) | 88 | 164 | 164 | 416 |
| Форма промежуточного контроля в семестре* | зачёт | экзамен | экзамен | - |
| Курсовая работа (курсовой проект) (КР, КП) | - | - | - | - |

Краткое содержание курса

Структурный подход к программированию

- Характеристики структурированных типов данных.
- Массивы. Линейные и двумерные массивы.
- Длинная арифметика.
- Строки.
- Множества.
- Записи.
- Типизированные файлы.
- Организация файлов записей.
- Нетипизированные файлы.
- Текстовые файлы.
- Прямой доступ к компонентам файлов.
- Сортировка файлов.

Модульное программирование. Программирование абстрактных типов данных

- Процедуры. Разработка и вызов.
- Функции. Разработка и вызов.
- Разработка программ на основе структурного подхода.
- Внешние подпрограммы.
- Рекурсивные подпрограммы.
- Модули. Структура и разработка.
- Стандартные модули.
- Динамически распределяемая память и ее использование при работе со стандартными типами данных.
- Однонаправленные списки.
- Двухнаправленные списки.
- Стеки.
- Очереди.

Объектно-ориентированное программирование

- Введение. Методология разработки объектно-ориентированного программного обеспечения Объектно-ориентированный анализ и проектирование;
- Абстрактные типы и структуры данных;
- Классы, объекты, поля, методы.;
- Конструкторы и деструкторы;
- Свойства и методы объектов;
- Раннее связывание и позднее связывание;
- Математические объекты: рациональные и комплексные числа, вектора, матрицы;
- Инкапсуляция – центральное понятие в парадигме объектно-ориентированного программирования Наследование – базовое понятие объектно-ориентированного программирования;

- Полиморфизм – базовое понятие объектно-ориентированного программирования;
- Основы объектно-ориентированного анализа;
- Основы объектно-ориентированного проектирования;
- Основы UML – унифицированного языка моделирования объектно-ориентированных систем

Форма текущего контроля

Контрольный тест 5 семестра, дисциплина «Программирование»

ВАРИАНТ III

1. Логическое выражение

$(N \text{ Mod } 10 \text{ Mod } 2) \text{ Or } (N \text{ Div } 10 \text{ Mod } 10 \text{ Mod } 2 = 0) \text{ Or } (N \text{ Div } 100 \text{ Mod } 2 = 0)$

должно принимать значение TRUE тогда и только тогда, когда истинно высказывание

- 1) в трёхзначном натуральном числе все цифры чётные; 2) в трёхзначном натуральном числе одна чётная цифра; 3) в трёхзначном натуральном числе две чётных цифры; 4) в трёхзначном натуральном числе хотя бы одна чётная цифра; 5) в трёхзначном натуральном числе нет чётных цифр.

2. Ошибку "Structure too large" (структура превышает максимально допустимый объём статической памяти) вызовет описание

- 1) `Type Vector = Array[Byte] Of Integer; Var C : Array[1..10] Of Vector;`
- 2) `Var T : File Of String;`
- 3) `Type A = Record S : String; A, B, C : Array[10..20] Of Real End;`
`Var M : Array[1..5, 1..8] Of A;`
- 4) `Var K : Array [Byte, Byte] Of String[6];`
- 5) `Var S : Array[-10000..10000] Of Sring[2].`

3. К процедурам для работы с динамическими переменными не относится

- 1) Mark; 2) New; 3) Release; 4) Seek; 5) Dispose.

4. Имеется описание

`Type A = Array[0..100] Of Real; B = ^A; Var M : Array[1..5] Of B;`

Для хранения массива M необходим объём памяти (байт)

- 1) 606; 2) 4; 3) 20; 4) 12120; 5) 6.

5. Фрагмент программы

```

K := 0;
While Not Eof(F) Do
Begin ReadLn(F, S); I := 1;
  While I <= Length(S) Do
  Begin If S[I] In ['A'..'Я', 'a'..'п', 'р'..'я']
    Then Begin K := K + 1;
          Delete(S, I, 1); I := I - 1
    End;
End;

```

```

        I := I + 1
    End
End;

```

выполняет следующее действие:

- 1) удаляет из текстового файла F все русские буквы; 2) определяет в текстовом файле количество символов, являющихся русскими буквами; 3) определяет в текстовом файле количество символов, не являющихся русскими буквами; 4) определяет в текстовом файле количество символов; 5) удаляет из текстового файла F все символы, не являющиеся русскими буквами.

6. В фрагменте программы (здесь Var F : File Of Integer; I, K, Vsp : Integer;)

```

    Reset(F); K := FileSize(F) - 1;
    For I := 0 To K Do
    Begin Seek(F, I); Read(F, Vsp); Seek(F, FileSize(F));
    Write(F, Vsp) End;

```

выполняется

- 1) сортировка файла; 2) изменение порядка следования элементов на обратный; 3) дописывание в конец исходного файла полную его копию с сохранением порядка следования элементов; 4) дописывание в конец исходного файла полную его копию с изменением порядка следования элементов на противоположный; 5) не выполняется никаких действий по изменению файла.

7. Имеется описание

```

Type Dn = (pn, vt, sr, cht, ptn, sb, vs); Mn = Set Of Dn;
Var V : Mn;

```

и фрагмент программы

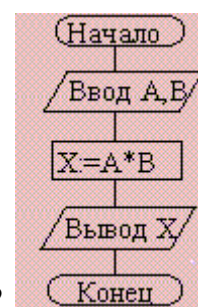
```

V := [pn..ptn] * [sr, ptn..vs] - [sb];

```

После исполнения этого фрагмента переменная V имеет значение

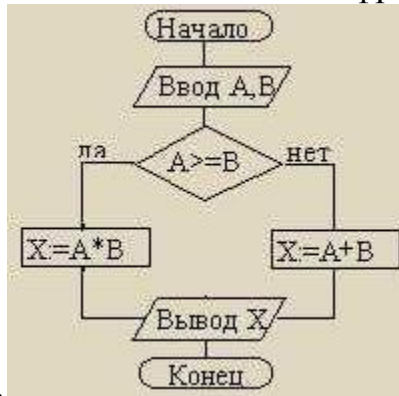
- 1) [pn..vs]; 2) [sr, ptn]; 3) [sb]; 4) []; 5) [pn..ptn].



8. Алгоритм какого типа изображен на блок-схеме?

- 1) циклический; 2) разветвляющийся; 3) вспомогательный; 4) линейный; 5) комбинация развилки и цикла.

9. После исполнения фрагмента программы, изображенного на блок-



схеме,

при $A = 5$, $B = 4$ значение X будет равно

1) 20; 2) 9; 3) 5; 4) 4; 5) 1.

10. В приведенном фрагменте программы (N типа LongInt, $N > 0$)

```

P := 1;
While P <= N Do
Begin
    Left := N Div (P * 10) * (P * 10);
    Right := N Mod P;
    K := ((N Mod (P * 10) Div P + 1) Mod 10) * P;
    N := Left + K + Right; P := P * 10
End;
  
```

натуральное число N изменяется по следующему правилу

1) не изменяется; 2) в каждый разряд прибавляется 1; 3) из каждого разряда вычитается 1; 4) в каждый разряд прибавляется 1, если значение в разряде — не девять, иначе заменяется на нуль; 5) каждая девятка в десятичной записи числа заменяется на нуль.

11. Цикл с предусловием выполняется так:

1) выполняется тело цикла, изменяется параметр цикла, проверяется условие продолжения выполнения цикла; 2) изменяется параметр цикла, проверяется условие продолжения выполнения цикла, выполняется тело цикла; 3) проверяется условие продолжения выполнения цикла, выполняется тело цикла; 4) тело цикла выполняется N раз (N — натуральное); 5) определяется, сколько раз должен быть выполнен цикл, и далее цикл с предусловием сводится к циклу с параметром.

12. В текстовом файле каждая строка заканчивается

1) числами 10 и 13; 2) символами с кодами 10 и 13; 3) символом с кодом 13; 4) числом 0; 5) символом с кодом 10.

13. Процедуры ReadLn и WriteLn можно использовать при работе с

1) типизированными файлами; 2) нетипизированными файлами; 3) типизированными и нетипизированными файлами; 4) текстовыми файлами; 5) любыми файлами.

14. Значение выражения

```
Ord(x > y) + Ord(Ord(z = 'F'))
```

при $x = 7$, $y = 0$, $z = 'F'$ равно

1) TRUE; 2) FALSE; 3) 0; 4) 1; 5) 2.

15. Идентификатор в Turbo Pascal не может начинаться с

1) латинской буквы; 2) заглавной латинской буквы; 3) цифры; 4) знака подчёркивания; 5) латинской буквы, а затем знака подчёркивания.

16. В приведенном фрагменте программы (First — ссылка на первый элемент списка; список объявлен следующим образом: Type SS = ^List; List = Record A : LongInt; Next : SS End;)

```
P := First; S := 0; While Not (P = Nil) Do Begin S := S + 1; P := P^.Next End;
```

определяется

1) первый элемент списка; 2) сумма элементов списка; 3) сумма первого и последнего элементов списка; 4) количество элементов списка; 5) количество звеньев списка, где указатель на следующее звено не Nil.

17. При выполнении фрагмента программы

```
Var C : Integer;
  Procedure R1(Var A : Integer; C : Boolean);
    Procedure R2;
      Var C : String;
      Begin A := 1 End;
    Begin C := True; R2 End;
  Begin C := 100; R1(C, False); WriteLn(C) End.
```

будет напечатано значение переменной C

1) True; 2) 1; 3) 100; 4) неизвестно что, поскольку значение переменной C не определено; 5) False.

18. Цикл в фрагменте программы

```
P := 4; Repeat P := P * 0.1 Until P < 0.0001;
```

будет исполнен

1) 0 раз; 2) 1 раз; 3) 4 раза; 4) 5 раз; 5) бесконечное число раз.

19. Кодировается шестнадцатибитовое целое со знаком (тип Integer).

1111111111110000₂ — это код числа

1) -15; 2) 15; 3) 16; 4) -16; 5) -30000.

20. Свойством алгоритма является

1) результативность; 2) цикличность; 3) возможность изменения последовательности выполнения команд; 4) возможность выполнения алгоритма в обратном порядке; 5) простота при записи на языках программирования.

21. Из перечисленных ниже в программе обязателен

1) раздел Var; 2) раздел Const; 3) раздел Type; 4) раздел Label; 5) раздел Begin ... End.

22. Ввод данных — это

1) процесс передачи данных из оперативной памяти на внешний носитель; 2) процесс ввода с клавиатуры каких-либо значений; 3) передача данных от внешнего носителя в оперативную память для обработки; 4) присваивание конкретных значений переменным, которые используются в программе; 5) запись файла на диск.

23. Значение R после выполнения операции логического присваивания

```
R := Not (A Or B Or (X > 2) And (Y < 0))
```

при A = False, B = False, X = 3, Y = 2 будет равно

1) -1; 2) False; 3) True; 4) 0; 5) 1.

24. С помощью какой из приведенных серий команд переменной *B* присваивается значение выражения?

$$\left(\frac{x+y}{x-y} - \frac{x-y}{x+y}\right) \cdot \left(\frac{x-y}{x+y} + \frac{x+y}{x-y}\right)$$

- а) $A := (x + y) / (x - y); B := (A - 1 / A) * (1 / A + A);$
 б) $A := (x + y) / (x - y); B := \text{Sqr}(A) - \text{Sqr}(1 / A);$
 в) $A := (x - y) / (x + y); B := \text{Sqr}(1 / A) - \text{Sqr}(A);$

1) а; 2) б; 3) в; 4) всех трех; 5) ни один из ответов 1–4 не является верным.

25. Значения переменных *a* и *b* после выполнения следующих действий

$a := 15 \text{ Div } (16 \text{ Mod } 7); b := 34 \text{ Mod } a * 5 - 29 \text{ Mod } 5 * 2;$

будут равны

1) $a = 1, b = 160;$ 2) $a = 1, b = 4;$ 3) $a = 7, b = 25;$ 4) $a = 7, b = 22;$ 5) $a = 7, b = 28.$

26. Во фрагмент алгоритма

```
For K := 10 To 99 Do
Begin P1 := K Div 10;
      P2 := K Mod 10;
      S := P1 + P2;
      If _____ Then WriteLn(K)
End;
```

печатающий все двузначные числа, в записи которых есть цифра *N* или сумма цифр которых равна самим числам, нужно вписать логическое выражение

1) $(P1 = N) \text{ Or } (P2 = N) \text{ And } (S = K);$ 2) $(P1 = N) \text{ Or } (P2 = N) \text{ Or } (S = K);$ 3) $(P1 = N) \text{ And } (P2 = N) \text{ Or } (S = K);$ 4) $((P1 = N) \text{ Or } (P2 = N)) \text{ And } (S = K);$ 5) $(P1 = N) \text{ And } (P2 = N) \text{ And } (S = N).$

27. Значения переменных *p* и *d* после выполнения фрагмента алгоритма

```
k := 47; Case k Mod 9 Of
5: Begin d := k; p := True End;
0..2: Begin d := 2; p := False End;
8: Begin d := 1; p := False End
Else Begin d := 1; p := True End
End;
```

равны

1) $p = \text{True}, d = 1;$ 2) $p = \text{False}, d = 2;$ 3) $p = \text{False}, d = 3;$ 4) $p = \text{True}, d = 47;$ 5) $p = \text{True}, d = 2.$

28. Тело цикла в программе

$a := 1; b := 1; \text{While } a + b < 8 \text{ Do begin } a := a + 1; b := b + 2 \text{ end};$

выполнится

1) 1 раз; 2) 2 раза; 3) 3 раза; 4) ни разу; 5) бесконечное число раз.

29. Элементы массива $p[1..5]$ равны соответственно 1, -1, 5, 2, 4. Значение выражения

$p[1] * p[3] - p[2 * p[2] + p[p[5] - p[2]]]$

равно

1) 8; 2) -8; 3) 12; 4) -12; 5) 6.

30. Задана строка *St*. Фрагмент алгоритма

$S := 0;$

```

For I := 1 To Length (St) Do
Begin
    Val (St[I], d, k);
    If K = 0 Then S := S + d
End;

```

- 1) определяет количество цифр в строке; 2) подсчитывает количество нулей в строке; 3) определяет сумму номеров позиций в строке, где стоят цифры; 4) подсчитывает сумму цифр в строке; 5) определяет сумму номеров позиций в строке, где стоят нули.

31. Какая из приведенных серий операторов определяет и печатает индекс последнего отрицательного элемента в линейном массиве из n элементов?

- а) $i := n$; While $(i \geq 1)$ And $(m[i] > 0)$ Do Dec (i) ; If $i < 1$ Then WriteLn (' $i = 0$ ') Else WriteLn (' $i =$ ', i);
б) $k := 0$; For $i := 1$ To n Do If $m[i] < 0$ Then $k := i$; WriteLn (' $i =$ ', k);
в) $i := n$; Repeat $i := i - 1$ Until $(m[i] < 0)$; WriteLn (' $i =$ ', i);
1) а, б; 2) б, в; 3) а, б, в; 4) б; 5) ни один из ответов 1–4 не верен.

32. Задан линейный массив $M[1..n]$.

```

Function Control (M: Myarray): Boolean;
Var I : Integer;
Begin I := 1;
    While (I <= n) And (M[I] > 0) Do Inc(I);
    Control := (I <= n);
End;

```

Если в данном массиве все элементы положительные, приведенная функция возвращает значение

- 1) n ; 2) True; 3) False; 4) $I \leq n$; 5) ни один из ответов 1–4 не верен.

33. Задан двумерный массив $X[1..n, 1..m]$. Процедура

```

Procedure Sub (Var X: Myarray);
Var i, j: Integer;
Begin For i := 1 To n Do
    For j := 1 To m Div 2 Do X[i, 2 * j] :=
X[i, 2 * j] + X[i, 1];
End;

```

- 1) к элементам столбцов в первой половине матрицы прибавляет элементы первого столбца соответствующей строки; 2) добавляет к матрице еще M столбцов с элементами, равными соответствующим элементам первого столбца; 3) к элементам четных столбцов прибавляет элементы первого столбца соответствующей строки; 4) к элементам четных строк прибавляет элементы первой строки соответствующего столбца; 5) меняет порядок столбцов таблицы.

34. Задан двумерный массив $X[1..n, 1..m]$. Функция

```

Function Check (X: Myarray): Boolean;
Var i, j : Integer; t : Boolean;
Begin t := True; i := 1;
    While t And (i <= n) Do
    Begin j:= 1; While (j <= m) And (X[i, j] <> 0) Do
Inc (j);
t:= (j = m + 1); Inc (i)

```



```

        End;
    Check := Not t
End;

```

возвращает значение

- 1) True, если все элементы массива ненулевые; 2) True, если в массиве есть элемент, равный нулю; 3) False, если в массиве есть элемент, равный нулю; 4) Not t; 5) ни один из ответов 1–4 не верен.

35. Среди перечисленных соответствий, которые необходимо соблюдать между формальными и фактическими параметрами

- а) соответствие по типу параметров; б) соответствие по количеству параметров; в) соответствие по типу используемых вспомогательных переменных; г) соответствие по порядку перечисления;

лишним является

- 1) а; 2) б; 3) в; 4) г; 5) ни один из ответов 1–4 не верен.

36. Определите тип выражения (здесь $A : \text{Array}[1..20] \text{ Of Real}$; $B : \text{Boolean}$; $C : \text{Integer}$)

```

C + Ord(Round(A[7]) + Ord(B)) - Trunc(A[1])

```

- 1) Real; 2) Integer; 3) Boolean; 4) Byte; 5) среди ответов 1–4 нет верного.

37. Список объявлен следующим образом

```

Type Ukaz = ^Zveno; Zveno = Record X : String; N :
Ukaz End;

```

```

Var First : Ukaz; {ссылка на начало списка}

```

В следующем фрагменте программы

```

P:= First;
While P^.N <> Nil Do
Begin
    B:= P; M := P;
    While B <> Nil Do Begin If B^.X < M^.X Then
M:= B; B:= B^.N End;
    S:= P^.X; P^.X:= M^.X; M^.X:= S; P:= P^.N
End;

```

выполняется

- 1) перемещение компонента к началу списка; 2) сортировка компонентов списка в порядке возрастания; 3) сортировка компонентов списка в порядке убывания; 4) перестановка соседних компонентов списка; 5) добавление в список нескольких новых компонент.

38. Фрагмент программы

```

S := A; A := B; B := S;

```

выполняет

- 1) обмен значений переменных A , B ; 2) присваивание переменным A , B значения S ; 3) замена значения переменной A значением переменной B ; 4) во фрагменте не выполняется никаких действий; 5) замена значения переменной B значением переменной A .

39. Имеется следующее описание

```

Type U = ^Zveno; Zveno = Record X, Y : Boolean;
Pred, Next : U End;

```

```

Var Logic : Boolean; A, B : Pointer; X, Y : U;

```

К ошибке компиляции "Несовместимость типов" приведет следующее присваивание

1) $A := X^{Next}.Next$; 2) $X := Y$; 3) $Logic := X.X$; 4) $X^{Next} := A$; 5) $X^{Next} := Y^{Next}$.

40. При присваивании изменяется

1) алгоритм; 2) имя переменной; 3) тип переменной; 4) значение переменной; 5) значение константы.

Зачет

1. Этапы решения задач с использованием ЭВМ.
2. Понятие алгоритма. Подходы к определению алгоритма. Свойства алгоритма. Способы записи алгоритма.
3. Понятие алгоритма. Понятие исполнителя. Система команд исполнителя.
4. Понятие величины. Типы величин. Присваивание величин. Совместимость по присваиванию.
5. Понятие о структурном программировании. Другие парадигмы программирования: сравнительная характеристика.
6. Языки программирования. Алгоритмические языки (алфавит, синтаксис, семантика). Способы описания синтаксиса (язык металингвистических формул, синтаксические диаграммы).
7. Система программирования Pascal ABC.
8. Структура программы, элементы языка (алфавит). Понятие типа данных.
9. Операции (арифметические, логические) на типах. Стандартные функции. Выражения.
10. Процедуры консольного ввода и вывода, управление вводом-выводом. Оператор присваивания. Совместимость по присваиванию.
11. Условный оператор. Оператор множественного ветвления (выбора).
12. Циклы в Pascal ABC: с предусловием, с постусловием. Связь с другими циклами.
13. Циклы в Pascal ABC: с параметром. Связь с другими циклами.
14. Структурированные типы данных. Линейные массивы. Примеры задач.
15. Структурированные типы данных. Двумерные массивы. Примеры задач.
16. Сортировка массивов. Метод выбора. Двоичный поиск в массиве.
17. Сортировка массивов. Метод обмена.
18. Сортировка массивов. Метод вставок.
19. Подпрограммы в Pascal ABC. Основные способы передачи параметров в подпрограмму, их сравнение.
20. Подпрограммы в Pascal ABC. Область видимости. Локальные и глобальные идентификаторы.
21. Процедуры. Организация и вызов. Примеры.
22. Функции. Организация и вызов. Примеры.
23. Простые типы данных в Pascal ABC.
24. Структурированные типы данных. Строковый тип данных в Pascal ABC: основные процедуры и функции, примеры.
25. Рекурсия. Механизм рекурсии. Примеры.
26. Сортировка массивов. Метод быстрой сортировки.
27. Множества в PascalABC. Примеры.

28. Комбинированный тип данных (записи). Оператор присоединения. Записи с вариантами. Программирование типовых алгоритмов обработки записей.
29. Файловые типы в PascalABC. Общие процедуры для работы с файлами.
30. Компонентные (типизированные) файлы

Экзамен

1. Текстовые файлы. Текст-ориентированные процедуры и функции. Типовые задачи.
2. Прямой и последовательный доступ к компонентам файла. Процедуры и функции, ориентированные на прямой доступ к компонентам файла.
3. Поиск в типизированных файлах. Сортировка файлов (на примере одного из методов).
4. Типизированные файлы. Файлы записей. Типовые алгоритмы обработки.
5. Статическая и динамически распределяемая память. Пример использования указателей.
6. Динамические структуры данных. Однонаправленный список. Процедуры обработки списка.
7. Динамические структуры данных. Двухнаправленный список. Процедуры обработки списка.
8. Динамические структуры данных. Кольцевой список (однонаправленный или двухнаправленный). Процедуры обработки списка.
9. Стек. Процедуры обработки.
10. Очередь. Процедуры обработки.
11. Двоичное дерево. Добавление в дерево и поиск в дереве.
12. Двоичное дерево. Удаление элемента из дерева.
13. Модуль. Общая структура модуля. Компиляция и подключение модуля.
14. Объектно-ориентированное программирование.
15. Пример реализации задачи на ООП в PascalABC
16. Концепция и технологии .NET
17. Парадигма объектно-ориентированного программирования и ее предшественники
18. Терминология объектно-ориентированного программирования: класс, объект, переменные экземпляра, метод, интерфейс, реализация, поведение, etc.
19. Три базовых понятия парадигмы объектно-ориентированного программирования
20. Инкапсуляция: абстракция, интерфейс и реализация
21. Инкапсуляция: средства защиты и доступа

[Оформление письменной работы согласно МИ 4.2-5/47-01-2013 Общие требования к построению и оформлению учебной текстовой документации](#)

**Учебно-методическое и информационное обеспечение дисциплины
Основная литература**

1. Фаронов Валерий Васильевич. Turbo Pascal : учеб. пособие / Фаронов Валерий Васильевич. - Санкт-Петербург : Питер, 2009. - 367 с. : ил. - (Учебное пособие). - ISBN 978-5-469-01295-5 : 260-00.
2. Сеницын Сергей Владимирович. Программирование на языке высокого уровня : учебник / Сеницын Сергей Владимирович, Михайлов Александр Сергеевич, Хлытчиев Олег Игоревич. - Москва : Академия, 2010. - 400 с. - (Высшее профессиональное образование). - ISBN 978-5-7695-6673-8 : 405-90.
3. Долинский М.С. Алгоритмизация и программирование на TURBO PASCAL от простых до олимпиадных задач: учеб. пособие/ М. С. Долинский. – 2005
4. Меньшиков Ф.В. Олимпиадные задачи по программированию: научное издание/ Ф. В. Меньшиков. - 2006
5. Фаронов В.В. Delphi. Программирование на языке высокого уровня.: учебник для студентов вузов, обучающихся по направлению подготовки дипломированных специалистов «Информатика и вычислительная техника»/ В. В. Фаронов. – 2009
6. Фаронов В.В. Turbo Pascal 7.0. Практика программирования: учебное пособие/ В. В. Фаронов. – 2009
7. Павловская Т.А. Паскаль. Программирование на языке высокого уровня: практикум: учеб. пособие для студентов вузов, обучающихся по направлению «Информатика и вычислит. Техника»/ Т. А. Павловская. – 2007
8. А.Н. Маслобоев Программирование на языке Object Pascal Учебно-методическое пособие, 2014г.(PDF)
9. А.Н. Маслобоев Операции с файлами и нестандартными типами данных в Turbo Pascal Учебно-методическое пособие, 2013г.(PDF)
10. А.Н. Маслобоев Практикум по алгоритмическому программированию в системе Turbo Pascal Часть II Учебно-методическое пособие, 2011г.(PDF)

Дополнительная литература

1. В.М. Пестриков, Г.А. Петров, с.в. Тихов основы алгоритмизации и программирования Учебно-методическое пособие для студентов всех специальностей заочной формы обучения, 2009г.(PDF)
2. В.М. Пестриков, Г.А. Петров, С.В. Тихов Основы Алгоритмизации и Программирования Учебно-методическое пособие , 2008г.(PDF)
3. Яковлева Лидия Леонидовна. Информатика и программирование : учеб. пособие. В 2 ч. Ч. 1 / Яковлева Лидия Леонидовна. - Чита : ЗабГУ, 2014. - 213 с. : ил. - ISBN 978-5-9293-0992-2. - ISBN 978-5-9293-0993-9 : 150-00.
4. Культин Н.Б. Turbo Pascal в задачах и примерах / Н. Б. Культин. - СПб. : БХВ-Петербург, 2008. - 256с. : ил. - ISBN 978-5-8206-0061-6 : 124-72
5. Павловская Татьяна Александровна. **Паскаль**. Программирование на языке высокого уровня: практикум : учеб. пособие / Павловская Татьяна Александровна. - Москва ; Нижний Новгород ; Воронеж : Питер, 2007. - 317 с. - (Учебное пособие). - ISBN 978-5-94723-008-6 : 131-00.

6. Кузан Дмитрий Ярославович.
 Программирование Win32 API в **DELPHI** / Кузан Дмитрий Ярославович,
 Шапоров Владимир Николаевич. - Санкт-Петербург: БХВ-Петербург, 2013. -
 368 с.: ил. + CD. - ISBN 978-5-94157-535-0 : 274-00.

7. Архангельский Алексей Яковлевич.
Delphi 2006: справ. пособие: Язык **Delphi**. классы, функции Win32 и . NET /
 Архангельский Алексей Яковлевич. - Москва : Бином, 2006. - 1152 с. - ISBN
 5951801389: 480-00.

Базы данных, информационно-справочные и поисковые системы*

1. H:\FMF\STUDENTS\STUDENT\...ПСУН «Изучаем Delphi»
2. H:\FMF\STUDENTS\STUDENT\...ПСУН «Изучаем С#»
3. www.zabspu.ru. - ➤ Ресурсы ➤ Дипломы ПСУН «Изучаем Delphi»,
 ПСУН «Изучаем С#»

Интернет-ресурсы:

| <i>№ n/n</i> | <i>Название сайта</i> | <i>Электронный адрес</i> |
|------------------|---|---|
| 1 | Все о программировании | http://forum.chertenok.ru/ |
| 2 | Форум программистов | http://forum.developing.ru/ |
| 3 | Форум по программированию на Delphi | http://www.delphisources.ru/forum/ |
| 4 | Видео-уроки по программированию | http://www.videourokov.ru/program/page/3/ |
| 5 | Большой список книг по программированию | http://www.rutracker.org/forum/viewtopic.php |
| 6 | Основы программирования для начинающих | http://www.opita.net/node/490 |

Ведущий преподаватель

Заведующий кафедрой