

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)


УТВЕРЖДАЮ
 Проректор по УМР _____
 _____ А.А.Панфилов
 « 17 » _____ 2016 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ПРОГРАММИРОВАНИЕ

Направление подготовки 44.03.05 – Педагогическое образование

Профили подготовки Информатика. Математика

Уровень высшего образования бакалавриат

Форма обучения очная

Семестр	Трудоёмкость зач. ед./ час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	СРС, час.	Форма промежуточного контроля (экз./зачет)
1	9 / 324	36	36	54	153	экзамен (45)
2	8 / 288	18	18	54	153	экзамен (45)
Итого	17 / 612	54	54	108	306	2 экзамена (90)

Владимир 2016

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целями освоения дисциплины «Программирование» являются:

- развитие логического и алгоритмического мышления у студентов;
- формирование и развитие умений планирования действий и структурирования информации, необходимых для решения поставленной задачи, при помощи фиксированного набора средств;
- изучение формального языка как средства управления компьютером;
- формирование способности отображать реальные объекты и явления в компьютерные информационные структуры;
- использование языка программирования как средства автоматизации вычислений, необходимых в учебной деятельности.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина «Программирование» относится к обязательным дисциплинам вариативной части, изучаемых по первому профилю направления.

В ходе изучения дисциплины студенты применяют знания и умения, сформированные при обучении предмету «Информатика и ИКТ» в общеобразовательной школе (содержательная линия «Алгоритмизация и программирование»). Учебный план дисциплины ориентирован на студентов, которые имеют низкий уровень знаний о языках и системах программирования. Материал, изучаемый в ходе освоения дисциплины «Программирование» является базовым для дальнейшей работы с языками программирования.

Знания и умения, практические навыки, приобретенные студентами в результате изучения дисциплины «Программирование», будут использоваться при освоении дисциплин «Численные методы и исследование операций», «Современные языки программирования», «Компьютерное моделирование», «Практикум по решению задач на ЭВМ», «Методика обучения информатике», а также при прохождении педагогической практики.

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Дисциплина «Программирование» нацелена на формирование у выпускника следующих компетенций:

- способность к самоорганизации и к самообразованию (ОК-6);

- готовность реализовывать образовательные программы по учебным предметам в соответствии с требованиями профессиональных стандартов (ПК-1);
- способность руководить учебно-исследовательской деятельностью обучающихся (ПК-12).

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

Знать:

- понятие алгоритма, его свойства, способы записи, исполнителей алгоритмов в объеме, не меньшем, чем данный раздел представлен в программе школьного курса «Информатики и ИКТ» (ОК-6, ПК-1);
- язык программирования как исполнитель алгоритма (ОК-6, ПК-1);
- назначение систем программирования (ОК-6, ПК-1);
- этапы решения задач с использованием компьютера (ОК-6, ПК-1);
- этапы выполнения компьютером программ, написанных на языке высокого уровня (ОК-6, ПК-1);
- основные понятия модульного и объектно-ориентированного программирования (ПК-1, ПК-12);
- методы построения и отладки программ;

Уметь:

- составлять и записывать в различных формах алгоритмы решения практических и прикладных задач (ОК-6, ПК-1);
- организовывать необходимую для решения задач информацию средствами выбранного языка программирования (ОК-6, ПК-12);
- организовывать действия созданного алгоритма с использованием выбранного языка программирования (ОК-6, ПК-1);
- разрабатывать внешний интерфейс реализации алгоритма решения задачи в выбранной среде программирования (ПК-12);
- анализировать разработанный алгоритм с точки зрения его работоспособности (ПК-1);
- анализировать структуру программы с целью ее дальнейшей оптимизации (ПК-12);
- реализовывать ручное исполнение алгоритма (ОК-6, ПК-1);
- оценивать эффективность работы программы (ПК-12).

Владеть:

- основными методами проектирования и отладки алгоритмов (ОК-6, ПК-1);
- навыками проектирования программ на языке программирования высокого уровня (ОК-6, ПК-1).

Изучение дисциплины «Программирование» способствует формированию следующих компетенций, зафиксированных в профессиональном стандарте педагога:

- анализ предлагаемого рассуждения с результатом: подтверждение его правильности или нахождение ошибки и анализ причин ее возникновения; помощь в самостоятельной локализации ошибки, ее исправлении; оказание помощи в улучшении (обобщении, сокращении, более ясном изложении) рассуждения;
- владение основными математическими компьютерными инструментами (визуализации данных, зависимостей; численные вычисления; экспериментальные лаборатории).

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 17 зачетных единиц, 612 часов.

№	Раздел (тема) дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)						Объем УР с применением интерактивных методов	Формы ТКУ, форма ПА
				Лекции	Практические занятия	Лабораторные работы	Контрольные работы	СРС	КП / КР		
	Семестр	1									
1.	Понятие алгоритма. Свойства алгоритма. Способы записи алгоритма.	1	1	2	2	2		12	2 (33%)		
2.	Структура программы в языке программирования Pascal. Типы данных. Операторы ввода, вывода и присваивания. Линейные алгоритмы	1	1-2	2	2	4		12	4 (50%)		
3.	Команда ветвления в полной и неполной форме. Логический тип и логические выражения. Оператор выбора.	1	3-4	4	4	6		16	6 (43%)		
4.	Циклические алгоритмы. Команды цикла с предусловием и постусловием.	1	5-6	4	4	6		19	4 (29%)	Рейтинг-контроль №1	
5.	Команда цикла с параметром	1	7-8	4	4	6		16	6 (43%)		
6.	Структурированные	1	9-	4	4	8		16	4		

	типы данных. Линейные массивы		10						(25%)	
7.	Алгоритмы поиска и сортировок в линейном массиве.	1	11	2	2	4		14	4 (50%)	Рейтинг-контроль №2
8.	Двумерные массивы.	1	12-13	4	4	6		16	4 (29%)	
9.	Символьный тип данных. Функции для работы с символьным типом	1	14-15	4	4	4		16	4 (33%)	
10	Строковые величины. Функции для работы со строковыми величинами.	1	16-18	6	6	8		16	6 (30%)	Рейтинг-контроль №3
	Итого в 1 семестре			36	36	54		153	44 (35%)	экзамен (45)
	Семестр	2								
11	Подпрограммы. Процедуры. Функции.	2	1-2	2	2	6		6	4 (40%)	
12	Параметры процедур и функций	2	3	2	0	2		8	2 (50%)	
13	Рекурсивные подпрограммы	2	4	0	2	4		12	2 (33%)	
14	Структурированный тип данных записи	2	5-6	2	2	4		12	4 (50%)	Рейтинг-контроль №1
15	Массивы записей	2	7	0	2	4		15	2 (33%)	
16	Тип множество	2	8	2	0	4		12	2 (33%)	
17	Файлы. Чтение и запись информации из файлов.	2	9	2	0	4		12	2 (33%)	
18	Типизированные и текстовые файлы. Операции для работы с файлами	2	10	0	2	6		12	4 (50%)	
19	Графические возможности среды программирования	2	11-12	2	2	4		12	2 (25%)	Рейтинг-контроль №2
20	Процедуры и функции модуля Graph	2	13	2	0	2		12	2 (50%)	
21	Анимация средствами среды программирования	2	14	2	2	4		12	4 (50%)	
22	Указатели. Динамические переменные	2	15-16	2	2	4		14	4 (50%)	
23	Динамические структуры данных (стек, очередь, список, бинарное дерево)	2	17-18	0	2	6		14	4 (50%)	Рейтинг-контроль № 3
	Итого в 2 семестре			18	18	54		153	38	экзамен

									(42%)	(45)
	Итого по курсу		54	54	108		306		82 (38%)	2 экзамена (90)

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Процесс изложения учебного материала на лекции сопровождается демонстрацией компьютерной презентации по рассматриваемой теме, а также выполнением интерактивных заданий при наличии требуемого технического обеспечения (интерактивной доски). Для изучения новых разделов проводятся проблемные лекции.

На практических занятиях проводятся групповые консультации, мозговые штурмы, используются разыгрывание ролей (выступление студента в роли школьного учителя). Решение задач выполняется в двух режимах: с использованием компьютера и последующей отладкой программы и проведением эксперимента по работоспособности программ при различных исходных данных, и без использования компьютера с дальнейшей имитацией работы компьютера при наличии созданной программы.

На лабораторных занятиях могут проводиться тренировки на машинных моделях с использованием компьютерных моделей различной степени сложности.

В рамках изучения дисциплины осуществляется поддержка студентов с использованием электронных образовательных технологий (размещение учебных материалов на сайте кафедры), организуются консультации с использованием дистанционных образовательных технологий

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Текущий контроль успеваемости проводится в форме рейтинг-контроля в соответствии с планом работы три раза в семестр. В рейтинг-контроль включаются следующие аспекты работы студента:

- выполнение и защита лабораторных работ;
- участие в устных опросах и коллоквиумах;
- выполнение письменных работ;
- выполнение дополнительных заданий в рамках аудиторной или самостоятельной работы.

Примерные задания рейтинг-контролей

1 семестр

Рейтинг контроль №1

1. Известно, что переменная $a=8$. Соотнести оператор вывода с результатом, отображающимся на экране

`writeln('a');`

`writeln(a);`

`writeln('a=', a);`

a=5
5
a

2. Определите значения переменных a, b, c после выполнения фрагмента алгоритма

`a:=5;`

`b:=16;`

`a:=2*b-4;`

`a:=a mod 6;`

```

c:=sqrt(a)+3;
a:=a+b div 3;

```

3. Запишите формулу в линейном виде по правилам языка Pascal

4. Восстановите формулу по её линейной записи

5. Выберите фрагменты, где представлена команда ветвления в полной форме

- | | |
|---|---|
| a) if x>y then x:=0
else y:=0; | c) if x>y then x:=0; |
| b) if x>y then
begin
x:=0;
y:=0;
end; | d) if x>y then
begin
x:=0; y:=0;
end
else y:=110; |

6. Определите значение переменных a и b в конце выполнения фрагмента алгоритма. Запишите, какая информация будет отображена на экране после выполнения фрагмента.

```

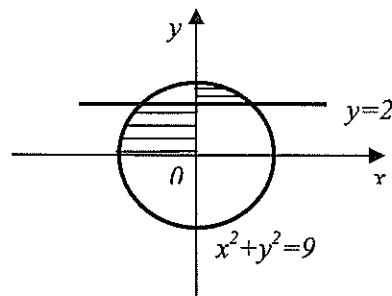
var a, b: integer;
begin
  a:=7;
  b:=12;
  a:=b mod a;
  if a>=b then
    begin
      a:=a-b;
      b:=b+2*a;
    end
  else
    a:=sqr(a);
    b:=b div a;
  writeln(b, ' \', a);
end.

```

7. Определить логические значения сложных условий

- | | |
|-------------------------|---------------------|
| a) (5<7) and not (8>11) | c) (2>6) or (7<6) |
| b) not((6>5) or (3>1)) | d) (2<4) and (5>11) |

4. Записать условие попадания точки с координатами в закрашенную область

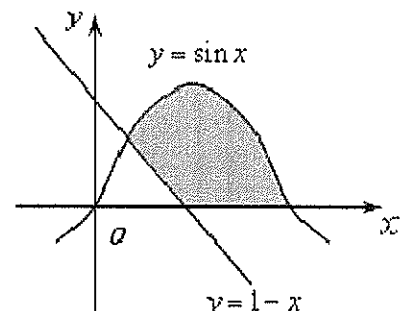


8. Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области. Программист торопился и написал программу неправильно.

```

var x, y: real;
begin
  readln(x, y);
  if y <= sin(x) then
    if y >= 1 - x then

```



```

    if y >= 0 then
        write('принадлежит')
    else
        write('не принадлежит')
    end.

```

Укажите пару чисел (x,y), при которых программа будет работать неправильно. Напишите, как исправить работу программы (напишите фрагмент верного кода).

9. Выполните вручную алгоритм и определите, что будет напечатано в результате работы следующего фрагмента программы:

```

var n, s: integer;
begin
    n := 1;
    s := 0;
    while n <= 101 do begin
        s := s + 7;
        n := n + 1
    end;
    write(s)
end.

```

10. Записать похожие характеристики в циклах с предусловием и постусловием.

11. Ниже записан алгоритм. Получив на вход число x, этот алгоритм печатает два числа: a и b. Укажите наименьшее из таких чисел x, при вводе которых алгоритм печатает сначала 2, а потом 15.

```

var x, a, b: integer;
begin
    readln(x);
    a:=0; b:=1;
    while x>0 do
        begin
            a:=a+1;
            b:=b*(x mod 10);
            x:= x div 10
        end;
    writeln(a); write(b)
end.

```

12. Изменить приведенный алгоритм, записав вместо цикла while цикл repeat так, чтобы результат работы алгоритма не изменился.

```

s:=0; readln(a);
while a>0 do
    begin
        if a mod 2 =0 then s:=s+a;
        readln(a);
    end;

```

5. Записать фрагмент алгоритма, который вводит в клавиатуры числа, а затем каждое число умножает на 2 и выводит на экран. Ввод последовательности оканчивается нулем.

Рейтинг-контроль №2

1. Выполните вручную фрагмент алгоритма. Вычислите значение переменной s.

s:=5;					
for i:=3 to 6 do					
s:=s+i*2;					

Укажите: параметр цикла, начальное значение, конечное значение, количество повторов цикла, значение переменной s.

2. Перепишите фрагмент алгоритма, заменив цикл for циклом while так, чтобы результат работы алгоритма не изменился

```
k:=10;
for j:=10 to 100 do
  begin
    x:=j*j;
    k:=k+2*x;
  end;
write(k);
```

3. Выполните фрагмент алгоритма. В ответе укажите, что будет выведено на экране после его выполнения.

```
a:=150;
for i:=8 downto 6 do
  begin
    a:=a-sqr(i);
    write(a);
  end;
```

4. Описать массив b, состоящий из 15 элементов. Элементы массива являются вещественными числами.

5. Написать фрагмент кода, задающий для элементов описанного в задании 1 массива b случайные значения в диапазоне [-20; 20].

6. Написать фрагмент программы для задания элементов массива по следующему правилу: каждый элемент массива равен квадрату своего индекса, увеличенному на 1.

7. В таблице Dat хранятся оценки (по десятибалльной шкале) студента по программированию за 8 прошедших с начала учёбы семестров. Определите, что будет напечатано в результате выполнения следующего алгоритма.

```
Var k, m, term: integer;
Dat: array[1..8] of integer;
Begin
Dat[1]:=7;          Dat[2]:=7; Dat[3]:=6;   Dat[4]:=8;
Dat[5]:=4;          Dat[6]:=5; Dat[7]:=8;   Dat[8]:=7;
term:=1; m := Dat[1];
for k :=2 to 8 do
if Dat[k] > m then
  begin
    m:= Dat[k];
    term := k;
  end;
write(term);
End.
```

8. В программе описан одномерный целочисленный массив с индексами от 0 до 10. В приведенном ниже фрагменте программы массив сначала заполняется, а потом изменяется. Чему будут равны элементы данного массива в конце выполнения алгоритма.

```
for i:=0 to 10 do
  A[i]:=i;
for i:=0 to 4 do
  begin
    k:=A[2*i];
    A[2*i]:=A[2*i+1];
    A[2*i+1]:=k;
  end;
```

Рейтинг-контроль №3

1. Напишите фрагмент программы, который вводит с клавиатуры матрицу размером 3×4 , а затем выводит её на экран в табличной форме.
2. Напишите программу, которая задает квадратную матрицу размера n так, что каждый элемент равен удвоенной сумме своих индексов.
3. Дан фрагмент программы, обрабатывающей двумерный массив $A[1..5, 1..4]$:

```
k=4;
for m:=1 to m=4 do
  begin
    k:=k+1;
    for n:=1 to 5 do
      begin
        k:=m-k;
        A[n,m]:=n*n+m*m-2*k;
      end;
    end;
```

Чему будет равно значение $A[3,1]$?

4. Элементы двумерного массива A размером 10×10 первоначально были равны 1. Затем значения некоторых из них меняют с помощью следующего фрагмента программы:

```
for n:=1 to 4 do
  for k:=1 to n+1 do
    begin
      A[n,k]:=A[n,k]-1;
      A[n,k+1]:=A[n,k]-1;
    end;
```

Запишите получившийся двумерный массив и укажите, сколько элементов массива в результате будут равны 0?

5. Выполнить последовательность операторов для строки a . Определить значение переменных a , n , c , p после выполнения этих операторов.

```
c:=copy(a,3,5);
delete(a,4,2);
n:=Length(a);
p:=pos('o',a);
a[n-p]:=c[2];
```

6. Опишите функции для работы со строковыми величинами (назначение функции, имя функции, необходимые параметры, способы вызова функции, пример): `delete`, `insert`, `inttostr`, `strtoint`;
7. Напишите программу, которая меняет местами трети слова из 12 букв следующим образом: первую треть размещает на месте третьей, вторую – на месте первой, третью – на месте второй.
8. Дана строка, состоящая из слов, разделенных пробелами. Вывести на экран в столбик те слова, которые имеют четную длину.

2 семестр

Рейтинг-контроль №1

1. Определение функции. Отличие функции от процедуры.
2. Передача параметров по значению.
3. Перечислите номера верных высказываний:
 - a. Порядок следования фактических параметров подпрограммы не имеет значения
 - b. Глобальные параметры доступны в любом месте программы, включая подпрограммы

- c. При передаче параметров по значению их значение в основной программе изменится после возвращения из подпрограммы
 - d. В подпрограмме-функции должен быть оператор, присваивающий некоторое значение имени функции
 - e. При вызове функции необходимо имя функции (с фактическими параметрами) указывать внутри некоторого оператора.
 - f. Фактические параметры процедуры и их типы перечисляются при описании процедуры, а формальные параметры – при вызове.
4. Определить значения переменной p в конце выполнения программы. Указать, какие значения будут выведены в конце выполнения программы (порядок вывода сохранить).

```

program example;
  var m,n,p:integer;
  procedure task(var a:integer; b:integer);
    var i:byte; s:integer;
  begin
    for i:=a to b do
      s:=s+i;
    a:=s;
    b:=a+b;
    writeln(a, ' ', b);
  end;
begin
  m:=3; n:=7;
  task(m,n);
  writeln(m, ' ', n);
  p:=m+n;
end.

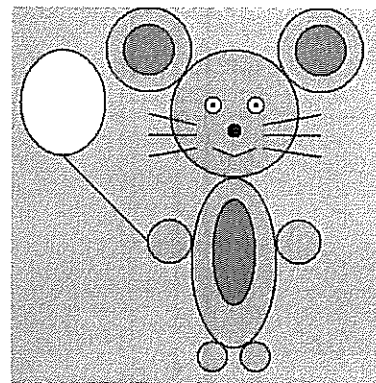
```

5. Дан массив слов. Для каждого слова необходимо вычислить и вывести количество слогов. Для подсчета количества слогов написать соответствующую подпрограмму.
6. С клавиатуры вводится матрица размера $n \times 3$. Для тройки элементов из каждой строки проверить. Образуют ли они пифагорову тройку, написав для этого соответствующую подпрограмму (числа a, b, c образуют пифагорову тройку, если $a^2 + b^2 = c^2$).
7. Описать тип `avto`, содержащий информацию о марке автомобиля, годе выпуска, пробеге и регистрационном номере. Описать две переменных данного типа.
8. Описать массив из 10 элементов, состоящий из записей, представленных в задании 7.
9. Написать фрагмент программы, который осуществляет ввод массива из задания 8 с клавиатуры.
10. Написать оператор, который для 8-го элемента описанного в задании 8 массива задает год выпуска 2008.

Рейтинг-контроль №2

1. Напишите фрагмент кода, который создает на диске C новый файл с именем `digite.txt` и заполняет его числами от 0 до 9.
2. Опишите назначение и синтаксис функции `assign`.
3. Опишите назначение и синтаксис функции `reset`.
4. Опишите назначение и синтаксис функции `append`.
5. Опишите назначение и синтаксис функции `rewrite`.
6. Дан типизированный файл целых чисел. Оставить в файле наибольшее из чисел, удалив остальные.
7. Охарактеризуйте инструмент `Pen` (назначение, характеристики). Приведите примеры изменения характеристик данного инструмента.

8. Опишите методы для рисования следующих графических примитивов (с указанием параметров):
 - a. прямоугольник
 - b. эллипс
 - c. вывод текста в графическом режиме.
9. Составить программу, рисует три треугольника: равносторонний, равнобедренный и разносторонний (координаты треугольников выберите самостоятельно). Цвет контура и заливки у каждого треугольника должен быть отличным от остальных. Под каждым рисунком должна быть подпись с указанием типа треугольника.
10. Создать изображение средствами графики в Pascal



Рейтинг-контроль №3

1. Написать программу для упорядочения массива методом быстрой сортировки.
2. Написать программу работы с базой данных отдела кадров предприятия. База хранится в текстовом файле, его размер может быть произвольным. Каждая строка файла содержит запись об одном сотруднике. Формат записи: фамилия и инициалы (30 позиций), фамилия должна начинаться с первой позиции), год рождения (5 поз.), оклад (10 позиций). Программа должна обеспечивать поиск в базе по заданным критериям, корректировку и дополнение данных.
3. На междугородной телефонной станции картотека абонентов, содержащая сведения о телефонах и их владельцах, организована в виде двоичного дерева. Написать программу, которая обеспечивает начальное формирование картотеки в виде двоичного дерева; производит вывод всей картотеки; вводит номер телефона и время разговора; выводит извещение на оплату телефонного разговора;

Примерные задания для самостоятельной работы студентов

1 семестр

1. Составьте блок-схемы для решения задач лабораторных работ.
2. Опишите компоненты структуры программы в Pascal (их назначение, синтаксис, примеры использования).
3. Составьте задания, для решения которых необходимо использовать команду ветвления в полной или неполной форме. Напишите программу для решения составленной задачи.
4. Составьте задания, для решения которых необходимо использовать циклический алгоритм. Напишите программу для решения составленной задачи с использованием различных видов циклов.
5. Составьте задания, для решения которых необходимо использовать циклический алгоритм со вложенным ветвлением. Напишите программу для решения составленной задачи.
6. Приведите пример необходимости использования структурированного типа данных – массива. Составьте и решите задачу, требующую организации данных в виде массива.
7. Приведите пример необходимости использования структурированного типа данных – двумерного массива. Составьте и решите задачу, требующую организации данных в виде двумерного массива.
8. Изучите дополнительные виды сортировок (пирамидальная, быстрая и т.д.). Напишите программы, реализующие данные алгоритмы сортировок.
9. Создайте программу для осуществления шифровки и дешифровки сообщения выбранным вами способом.

2 семестр

1. Реализуйте программы ранее решенных задач с использованием подпрограмм.

2. Рассмотрите примеры задач, для решения которых необходима реализация рекурсивных подпрограмм (игра Баше, Ханойская башня и т.д.). Составьте программы для решения этих задач.
3. Составьте задачи, для решения которых необходимо ввести структурированный тип запись. Напишите программу для решения задачи.
4. Для составленных ранее задач организуйте ввод и вывод информации с использованием файлов.
5. Создайте и выполните с помощью языка программирования изображение, включающее в себя как можно большее количество графических примитивов.
6. Составьте программу, анимирующую один из физических процессов (на выбор студента).

Вопросы к экзамену по дисциплине (1 семестр)

1. Алгоритм, свойства алгоритма.
2. Способы записи алгоритма.
3. Исполнители алгоритмов.
4. Языки программирования и среды программирования.
5. Способы отладки программ.
6. Среда программирования Pascal. Структура программы. Описание констант и переменных.
7. Объявление переменных в программе. Схема типов данных в Pascal.
8. Числовые целые и вещественные типы данных. Логический тип.
9. Линейные алгоритмы. Операторы ввода и вывода в Pascal. Оператор присваивания.
10. Команда ветвления в полной и неполной форме (применение, блок-схема, синтаксис в Pascal).
11. Цикл с параметром (применение, блок-схема, синтаксис в Pascal).
12. Цикл с предусловием (применение, блок-схема, синтаксис в Pascal).
13. Цикл с постусловием (применение, блок-схема, синтаксис в Pascal).
14. Вложенные циклы.
15. Массивы в Pascal (определение, описание массивов, ввод и вывод массивов).
16. Массивы в Pascal. Сортировка методом простого выбора.
17. Массивы в Pascal. Сортировка пузырьковым методом.
18. Массивы в Pascal. Алгоритмы линейного и бинарного поиска элемента в массиве.
19. Двумерные массивы – матрицы (назначение, описание матриц, ввод и вывод матриц, примеры использования).
20. Символьный тип. Функции для работы с символами.
21. Строковые величины. Функции для работы со строками.

Вопросы к экзамену по дисциплине (2 семестр)

1. Подпрограммы. Функции.
2. Подпрограммы. Процедуры.
3. Подпрограммы. Локальные и глобальные параметры. Передача параметров структурированных типов.
4. Подпрограммы. Передача параметров по имени и по значению.
5. Рекурсивные подпрограммы.
6. Тип запись. Назначение, описание, ввод-вывод, обращение к записям.
7. Оператор with.
8. Массивы записей.
9. Файлы. Описание файлов, операции для работы с файлами.
10. Файлы. Запись информации в файл, чтение информации из файла.
11. Текстовые файлы. Описание, операции для работы с текстовыми файлами.
12. Типизированные файлы. Функции для работы с типизированными файлами.
13. Графические возможности модуля crt.
14. Модуль GraphABC. Окно модуля. Основные принципы работы.
15. Графические примитивы модуля GraphABC (точка, прямая, эллипс, прямоугольник, дуга).

16. Вывод текста в окне Graph. Параметры текста.
17. Объект Pen. Назначение. Параметры объекта Pen.
18. Объект Brush. Назначение. Параметры объекта Brush.
19. Построение графиков функции средствами модуля GraphABC.
20. Анимация в среде программирования PascalABC.NET.
21. Событийные процедуры.
22. Указатели в Pascal
23. Динамические переменные
24. Динамические структуры данных. Стек
25. Динамические структуры данных. Очередь
26. Динамические структуры данных. Список

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) основная литература

1. Тишин, В. И. Программирование на Паскале [Электронный ресурс]: практикум / В. И. Тишин. – Эл. изд. – М.: БИНОМ. Лаборатория знаний, 2013. – 364 с.: ил.
<http://znanium.com/bookread2.php?book=500900>
2. Грацианова, Т.Ю. Программирование в примерах и задачах [Электронный ресурс] / Т.Ю. Грацианова. – М.: БИНОМ, 2015. – (ВМК МГУ – школе)
<http://www.studentlibrary.ru/book/ISBN9785996327812.html>
3. Комлев, Н.Ю. Самоучитель игры на Паскале. ABC и немного Турбо. Самоучитель игры на Паскале. ABC и немного Турбо [Электронный ресурс] / Комлев Н.Ю. – М.: СОЛОН-ПРЕСС, 2013. – 256 с.: ил.
<http://www.studentlibrary.ru/book/ISBN9785913591128.html>

б) дополнительная литература:

1. Окулов, С.М. Программирование в алгоритмах [Электронный ресурс] / С.М. Окулов. – М.: БИНОМ, 2014. – Электронное издание на основе: Программирование в алгоритмах [Электронный ресурс] / С.М. Окулов. – 5-е изд. (эл.). – М.: БИНОМ. Лаборатория знаний, 2014. – 383 с.: ил. – (Развитие интеллекта школьников).
<http://www.studentlibrary.ru/book/ISBN9785996323111.html>
2. Канцедал, С.А. Алгоритмизация и программирование: Учебное пособие / С.А. Канцедал. – М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014. – 352 с.: ил.; – (Профессиональное образование).
<http://znanium.com/catalog.php?bookinfo=429576>
3. Давыдова Н.А. Программирование [Электронный ресурс]: учебное пособие/ Давыдова Н.А., Боровская Е.В. – Электрон. текстовые данные. – М.: БИНОМ. Лаборатория знаний, 2015. – 239 с.
<http://www.iprbookshop.ru/6485>

в) интернет-ресурсы:

1. <http://pascalabc.net/>
2. <https://projecteuler.net/>

г) периодические издания:

1. Журнал «Информатика».
<http://информатика.1сентября.рф>
2. Журнал «Информатика и образование».
<http://infojournal.ru/>

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Проведение практических и лекционных занятий осуществляется с использованием мультимедийного комплекса (компьютер +проектор) или интерактивной доски.

Лабораторные занятия проводятся в компьютерном классе с возможностью индивидуальной работы каждого студента за компьютером. Необходимое программное обеспечение – PascalABC.NET (может быть заменено на PascalABC, Free Pascal версии не ниже, чем 2.6, Turbo Pascal версии не ниже, чем 7.0).

Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО по направлению 44.03.05 Педагогическое образование.

Рабочую программу составил

ст. преподаватель кафедры информатики и ИТО  А.А. Мартынова

Рецензент

Заместитель директора ГБПОУ Владимирской области

«Владимирский педагогический колледж»  Н.И. Коршунова

Программа рассмотрена и одобрена на заседании кафедры информатики и ИТО

протокол № 7а от 10.03.2016 года

Заведующий кафедрой

информатики и ИТО  Ю.А. Медведев

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии направления 44.03.05 Педагогическое образование

Протокол № 3 от 17.03.2016 года

Председатель комиссии  М.В. Артамонова