

**Муниципальное общеобразовательное учреждение
«Средняя школа №56»**

Принято с учетом мнения
педагогического совета школы
протокол № 1
от «28» августа 2018 г

УТВЕРЖДАЮ
Директор школы _____ Т.Н.Озерова
Приказ № 01-014/364 от 30.08.2019



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ
ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
«ПРОГРАММИРОВАНИЕ»**

Техническая направленность

**Возраст обучающихся: 16-18 лет
Срок реализации программы: 1 год**

**Автор-составитель: педагог
дополнительного образования:
Корчевцева Л.И.**

город Ярославль,
2019 год

2. ОГЛАВЛЕНИЕ

№ раздела	Название структурных компонентов	№ страницы
3	Пояснительная записка	3 – 5
4	Учебно-тематический план	6
5	Содержание программы	7
6	Обеспечение программы	8
7	Оценочные материалы	9
8	Список информационных источников	10

3. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Актуальность программы:

Информатизация общества в современных условиях предусматривает обязательное применение компьютеров в школьном образовании, что призвано обеспечить компьютерную грамотность и информационную культуру обучающихся.

Компьютерная грамотность выпускника средней школы складывается из следующих компонентов:

- он должен знать общие принципы устройства, работы ПК и ее логико-функциональной структуры;
- основные направления их использования в народном хозяйстве;
- уметь самостоятельно поставить и решить с помощью ЭВМ простые задачи на вычисление, управление, моделирование, хранение и обработку информации.

Программирование - стержень профильного курса информатики. Изучение основ программирования связано с целым рядом умений и навыков (организация деятельности, планирование ее), которые по праву носят общеинтеллектуальный характер и формирование которых - одна из приоритетных задач современной школы. Очень велика роль изучения программирования для развития мышления школьников, формирования многих приемов умственной деятельности. Здесь роль информатики сродни роли математики в школьном образовании. Поэтому необходимо использовать действительно большие возможности программирования, решения соответствующих задач для развития мышления школьников, формирования многих общеучебных умений и навыков.

Учитывая быстрое развитие индустрии программного обеспечения, интенсивную разработку различных пакетов прикладных программ, можно достаточно обоснованно предположить, что подавляющее большинство будущих пользователей ПК не станет самостоятельно готовить программы для решения собственных производственных задач, а будет использовать уже готовые, разработанные профессиональными программистами. Поэтому неуклонно из курса школьной информатики вытесняется программирование, которое заменяется изучением прикладных программ. С одной стороны это правильно, т.к. из всех выпускников ярко выраженное алгоритмическое мышление имеют всего 3-5% учащихся, но с другой стороны если школа не будет знакомить с основами программирования, то у этих 3-5% не будет шансов реализовать свой потенциал. Да и в вузах преподавание программирования никто не отменял, причем используются различные языки, но самым распространенным является ПАСКАЛЬ:

- Он пригоден для обучения программированию как систематической дисциплине, так как основан на ряде фундаментальных понятий, ясно и естественно отраженных в языке, а также достаточно легок в изучении.
- Он позволяет строить программу-последовательность инструкций (операторов) в виде блоков, что создает условия для так называемого структурного программирования.
- Он содержит полный набор структурных типов данных, а также развитые средства построения из них новых типов данных, позволяющих использовать адекватное представление абстрактных понятий. Это дает возможность формулировать более простые и эффективные алгоритмы.
- Для этого языка созданы программные системы, позволяющие доказать правильность алгоритмов.
- Программы на этом языке обладают повышенной надежностью благодаря избыточности информации, сообщаемой компилятору (например, к избыточным относится требование описывать все переменные). Эта избыточная информация используется при проверке согласованности программы без ее выполнения.

Категория учащихся

Программа предназначена для обучающихся **16-18 лет, в группе 15 человек.** Условия приема – заявление (свободный набор).

Минимально необходимый уровень знаний и технологических умений обучающихся перед прохождением программы:

Обучающиеся:

- владеют языком программирования Pascal;
- умеют применять базовые алгоритмические структуры при программировании решения задач.

Направленность программы – техническая

Цель программы:

Данная программа позволит обучающимся, имеющим склонность к программированию:

- проявить свои творческие возможности при изучении различных тем, решении задач, переходя от простых примеров к сложным,
- развивать алгоритмическое мышление,
- участвовать в различных олимпиадах по программированию.

Задачи:

- понять значение алгоритмизации как метода познания окружающего мира, принципы структурной алгоритмизации;
- овладеть базовыми понятиями теории алгоритмов;
- научиться разрабатывать эффективные алгоритмы и программы;
- приобрести навыки выполнения технологической цепочки разработки программ средствами языка программирования Паскаль,
- формирование образного и теоретического мышления;
- формирование умения планировать свою деятельность.

Ожидаемый (прогнозируемый) результат прохождения курса.

После прохождения курса обучающиеся владеют следующими знаниями, умениями и способами деятельности:

- владеют знаниями об основных алгоритмических конструкциях;
- умеют реализовывать соответствующие алгоритмы на компьютере с помощью языка программирования Pascal;
- знают способы организации алгоритмов с применением циклов и умеют реализовывать их на Pascal;
- знают способы организации алгоритмов с применением файлов и умеют реализовывать их на Pascal;
- знают способы организации алгоритмов обработки строковых величин и умеют реализовывать их на Pascal;
- знают способы организации алгоритмов обработки массивов и умеют реализовывать их на Pascal;
- способны осуществлять рефлексивную деятельность, оценивать свои результаты. (находить и устранять ошибки в программах).

После завершения курса обучающийся должен выполнить индивидуальный **проект**, который он может разработать самостоятельно или взять у педагога.

Такая форма работы позволяет педагогу увидеть и использовать индивидуальные способности каждого обучающегося, привить детям вкус к творчеству и исследовательской деятельности.

Допустимо, чтобы над проектами обучающиеся работали группами. Опыт работы над проектами позволяет сформулировать некоторые общие методические рекомендации:

1. Проект должен быть небольшим. Ученику может не хватить терпения довести большой проект до конца.
2. Проект должен побуждать к получению новых знаний. Получение знаний правильно мотивируется, и этот мотив выставляет не преподаватель, а сам ученик.
3. Проект должен иметь полезный результат, имеющий общественное признание.
4. При утверждении проекта должны учитываться:
 - успеваемость;
 - личностные качества;
 - творческие способности;
 - объем работы.
5. Проект целесообразно разбить на этапы и последовательно оценивать педагогом.

Задачи для проектов подбирает педагог, но обучающийся и сам может предложить интересную ему тему.

Качество знаний и умений обучающегося оценивается следующими характеристиками:

- знание основных алгоритмических конструкций;
- умение составить и записать алгоритм с использованием соответствующей алгоритмической конструкции;
- умение найти более эффективный способ решения задачи;
- умение тестировать программу.

Отличительные особенности программы:

При изучении данного курса особый акцент делается на приобретение новых знаний, а также на развитие способностей приобретать знания самостоятельно, на умение анализировать ситуацию, выбирать самый удобный способ составления программ для решения задач, аргументировано отстаивать свою точку зрения.

Данный курс раскрыл значение программирования и суть профессии программиста, ознакомил обучающихся со средой и основами программирования на языке Pascal, подготовил обучающихся к практическому использованию полученных знаний при решении учебных задач, а затем профессиональной деятельности.

Обоснование отбора содержания:

Курс построен в виде последовательности практических занятий, занятия имеют строгий порядок, предусматривающий равномерное усвоение материала. Каждое из занятий заканчивается блоком задач для самостоятельного решения, необходимых для закрепления пройденной темы.

Общая характеристика учебного процесса:

- при изучении курса используются лекции и практические работы.
- занятия проводятся по группам 15 человек в кабинете информационных технологий.
- курс обучения заканчивается написанием программы для решения одной из задач из различной предметной области, это может быть тест, модель эксперимента, программа проверки знаний и т.д.

Режим организации занятий: 1 час (45 минут) в неделю.

4. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Тема	Теория	Практика	Всего
1	Понятие алгоритма. Алгоритмизация. Основные алгоритмические конструкции.	1		1
2	Инструктаж по технике безопасности. История возникновения языка Паскаль. Составление алгоритмов. Входной контроль.		1	1
3	История возникновения языка программирования Структура программы на Паскале. Операторы ввода, вывода.	0,5	0,5	1
4	Типы данных, операторы арифметических операций, операторы преобразования типов данных.	1		1
5	Решение вычислительных задач.		1	1
6	Условный оператор, оператор выбора.	1		1
7	Решение задач с использованием условного оператора.		1	1
8	Решение задач с использованием оператора выбора.		1	1
9	Основные виды циклов: цикл с параметром, цикл «до», цикл «пока».	1		1
10	Решение задач с использованием цикла с параметром.		1	1
11	Решение задач с использованием цикла «до»		1	1
12	Решение задач с использованием цикла «пока»		1	1
13	Символьный тип данных, обработка символов.	1		1
14	Решение задач на обработку символов.		1	1
15	Промежуточная аттестация. Практическая работа: Решение задач на обработку символов.		1	1
16	Строковые типы данных, обработка строк.	1		1
17	Инструктаж по технике безопасности. Решение задач на обработку строк.		1	1
18	Решение задач на обработку строк.		1	1
19	Одномерные массивы. Способы заполнения одномерных массивов	0,5	0,5	1
20	Сортировка массива. (Сортировка методом простого выбора. Сортировка методом простого обмена)		1	1
21	Двумерные массивы в Pascal. Заполнение двумерного массива и вывод на экран.	0,5	0,5	1
22	Максимум и минимум в двумерном массиве.		1	1
23	Поиск по условию в двумерном массиве.		1	1
24	Процедуры.	0,5	0,5	1
25	Использование процедур в программах.		1	1
26	Функции.	0,5	0,5	1
27	Использование функций в программах.		1	1
28	Использование процедур и функций в программах.		1	1
29	Файловый тип данных. Общие положения	1		1
30	Текстовые файлы. Обработка текстовых файлов		1	1
31	Работа с файлами: создание.		1	1
32	Работа с файлами: чтение и запись.		1	1
33	Работа с файлами: удаление.		1	1
34-36	Итоговая аттестация. Защита проекта.		3	3

5. СОДЕРЖАНИЕ ПРОГРАММЫ

№	Название темы	Количество часов			Содержание
		Всего	Теория	Практика	
1	Алгоритмизация. Основные алгоритмические конструкции.	2	1	1	Основные алгоритмические конструкции. Составление алгоритмов.
2	История возникновения языка Паскаль, знакомство со средой ABCPascal.	1	0,5	0,5	История возникновения языка программирования Паскаль, версии Паскаля, достоинства, основное окно, работа с меню, работа с окнами.
3	Структура программы на Паскале. Операторы ввода-вывода.	1	0,5	0,5	Структура программы на Паскале, основные разделы программы, знакомство с оператором ввода, вывода.
4	Целочисленные, вещественные и логические типы данных, арифметические операции.	2	1	1	Типы данных, операторы арифметических операций, операторы преобразования типов данных, решение вычислительных задач.
5	Условный оператор, оператор выбора, логический тип данных.	3	1	2	Условный оператор, оператор выбора, решение задач с использованием данных операторов.
6	Программирование циклов.	4	1	3	Основные виды циклов: цикл с параметром, цикл «до», цикл «пока», использование данных структур в программах.
7	Обработка текстовой информации в Pascal. Строковые функции.	3	1	2	Символьный тип данных, обработка символов. Решение задач на обработку символов.
8	Строки. Обработка строк.	3	1	2	Строковые типы данных, обработка строк. Решение задач на обработку строк.
9	Массивы.	5	1	4	Виды массивов: одномерные и двумерные массивы, ввод элементов массивов, поэлементный вывод массивов. Сортировка.
10	Процедуры и функции.	5	1	4	Процедуры и функции. Использование процедур и функций в программах.
11	Файлы, работа с файлами.	4	1	3	Виды файлов, способы доступа к файлам. Работа с файлами: создание, чтение, запись, удаление.
12	Защита проекта.	3		3	Защита проекта.

6. ОБЕСПЕЧЕНИЕ ПРОГРАММЫ

6.1. Материально-техническое обеспечение

Для занятий требуется просторное светлое помещение, отвечающее требованиям СанПиНов.

Для успешной реализации учебно-педагогического процесса требуется наличие материально-технической базы:

- Кабинет информационных технологий 12 ученических + 1 ПК педагога с установленной средой Pascal ABC.

6.2. Организационное обеспечение

Начало учебного года с 1 сентября 2019 г., окончание учебного года - 31 мая 2020 года. Регламентирование образовательного процесса на учебный год производится согласно графику:

Название месяца	сентябрь	октябрь	ноябрь	декабрь	январь	февраль	март	апрель	май
Кол-во занятий	4	4	4	4	4	4	4	4	4

Количество учебных недель – 36. Занятия в объединении проводятся в соответствии с расписанием, утвержденным директором школы. Занятия проводятся во второй половине дня. Между основной занятостью обучающихся и занятиями в объединении соблюдается перерыв не менее часа.

Занятия в объединении проводятся в каникулярное время и не проводятся в праздничные дни. Промежуточная аттестация учащихся проводится в декабре, мае. Родительские собрания по дополнительному образованию детей проводятся в начале учебного года (сентябрь) и в конце учебного года (май).

6.3. Кадровое обеспечение

1. Педагог дополнительного образования.
2. Педагог-психолог.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Примеры заданий для практических работ находятся в [ПРИЛОЖЕНИИ В](#).
Примерные задания для контрольного тестирования в [ПРИЛОЖЕНИИ Г](#).

Возможные проекты данного курса:

Необходимо изобразить:

- сову, хлопающую глазами;
- распускающийся цветок;
- танцующую балерину;
- планету, вращающуюся вокруг солнца;
- сечение шара, параллельное осевому сечению;
- сечение прямоугольного параллелепипеда по трем точкам, расположенным на ребрах параллелепипеда;
- сечение усеченной треугольной пирамиды, расположенное под углом 45° к основанию;
- любой тест, выполненный в графическом режиме, и т.д.

Пример итогового проекта приведен в:

[ПРИЛОЖЕНИИ В](#).

[ПРИЛОЖЕНИЕ А](#).

[ПРИЛОЖЕНИЕ Б](#).

[ПРЕЗЕНТАЦИЯ](#)

8. СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

Список для педагога:

1. Н.Кульгин. Turbo Pascal в примерах и задачах. С-Пб, «БХВ-Петербург»,
2. Д.М.Ушаков, Т.А.Юркова. Паскаль для школьников. С-Пб, «Питер»,
3. Информатика. Конспекты уроков 9-11 кл. Волгоград, «Учитель»,
4. Катицкая Н. Ю., Бельчусов А.А. Учебное пособие «ПАСКАЛЬ. Конспект для учителя», издательство Чувашского республиканского института образования.
5. www.college.ru Статья “Типология элективных курсов и их роль в организации профильного обучения”, Орлов Владимир Алексеевич, канд. пед. наук, зав. лабораторией физики ИОСО РАО.
6. Лапчик М.П., Семакин М.П., Хеннер Е.К. Методика преподавания информатики: Учеб. пособие для студ. пед. вузов / под общей ред. М.П. Лапчика. 2-е изд. М.: Издательский центр Академия, 2005. 624 с.

Список для обучающихся:

1. Turbo Pascal в задачах и примерах, Н.Кульгин, СПб.: БХВ-Петербург, 2005. -256 с.: ил.; Адаменко А. Pascal на примерах и задачах из математики – “БХВ-Петербург”, 2005;
2. Алексеев Е. Р., Чеснокова О. В., Турбо Паскаль 7.0, NT Press Москва, 2005;
3. Информатика. Задачник-практикум в 2 т. / Л.А.Залогова, М.А.Плаксин, С.В.Русаков и др. Под ред. И.Г.Семакина, Е.К.Хеннера: Том 1., М.:БИНОМ. Лаборатория знаний, 2006.
4. www.intuit.ru Интернет – Университет Информационных технологий;
5. Зюзьков В.М. Программирование на языке высокого уровня: Учебное пособие. - Томск: Томский межвузовский центр дистанционного образования, 2003. - 217 с.;
6. Кузнецов А. А. Профильное обучение: проблемы, перспективы развития // Народное образование, 2003, 4;
7. Кузнецов А. А. Элективные курсы в профильном обучении. // Учительская газета, 2004. с. 49-50.

Список для родителей:

1. Кузнецов А.А. Элективные курсы образовательной области Информатика // Элективные курсы в профильном обучении: Образовательная область Информатика // Министерство образования РФ Национальный фонд подготовки кадров. М.: Вита-Пресс, 2004. С. 5-20;
2. Кузнецов А. А., Захарова Т. Б. Принципы дифференциации содержания обучения информатике// Информатика и образование, 1997 – 4;
3. Концепция модернизации российского образования на период до 2010 г. // Вестник образования, 2002, 6. с. 11-40.

ПРИЛОЖЕНИЕ А

(Обязательное)

Примеры движения объектов

{Программа “Летающая тарелка”}

```
program nlo;
uses Crt, Graph;
var
    x, y, i, dx, dy, gd, gm: integer;
    Tarelka: array [1..600] of byte;
begin
    x:=320;
    y:=240;
    gd:=0;
    initgraph (gd, gm, ‘ ‘);
    randomize;
{Построение летающей тарелки}
    setfillstyle (SolidFill, 4);
    fillellipse (100, 50, 20, 8);
    ellipse (100, 46, 190, 357, 20, 6);
    line (107, 44, 110, 38);
    circle (110, 38, 2);
    line (93,44, 90, 38);
    circle (90, 38, 2);
{Запомнили изображение тарелки и стерли его}
    getimage (79, 36, 121, 59, Tarelka);
    putimage (79, 36, Tarelka, XORput);
{Построение звездного неба}
    for i:=0 to 1000 do
        putpixel (random(639), random(479), random(15)+1);
    repeat {Бесконечный цикл до нажатия клавиши}
        putimage (x, y, Tarelka, XORput); {Вывод тарелки}
        delay (6000); {Задержка}
        putimage (x, y, Tarelka, XORput); {стирание тарелки}
{Перемещение тарелки}
        dx:=random (60);
        if odd (dx) then dx:=-dx;
        x:=x+dx;
        if x>590 then x:=590;
        if x<0 then x:=0;
        dy:=random (40);
        if odd (dy) then dy:=-dy;
        y:=y+dy;
        if y>450 then y:=450;
        if y<0 then y:=0;
        until KeyPressed;
        closegraph;
end.
```

{Программа «Часы»}

```
program time;
uses Crt, Dos, Graph;
```

```

var
  gd, gm, k: integer;
  h1, m1, s1, h2, m2, s2, hs2: word;
procedure cifra(x, y, n: integer);
type
  a4 = array [0..4] of integer;
  a6 = array [1..7] of byte;
const
  a = 4; b = 20; c = a + a + b;
  d1: a4 = (a, b, a, -a, -b);
  d2: a4 = (-a, 0, a, a, 0);
  dx: a6 = (0, 0, 0, 0, c, 0, c);
  dy: a6 = (0, c, 2 * c, 0, 0, c, c);
  q: array [0..9] of byte =
    ($5F, $5, $76, $75, $2D, $79, $7B, $45, $7F, $7D);
var
  xy: array [0..5] of PointType;
  j, k, d: byte;
begin
  setfillstyle(0, 0);
  bar(x - a, y - a, x + (c + a + a), y + 2 * (c + 2 * a));
  d := q[n];
  for j := 1 to 7 do
    begin
      if ((d and ($80 shr j)) = 0) then continue;
      xy[0].x := x + dx[j];
      xy[0].y := y + dy[j];
      for k := 1 to 5 do
        if j < 4 then begin
          xy[k].x := xy[k - 1].x + d1[k - 1];
          xy[k].y := xy[k - 1].y + d2[k - 1]; end
        else begin
          xy[k].x := xy[k - 1].x - d2[k - 1];
          xy[k].y := xy[k - 1].y + d1[k - 1]; end;
        setfillstyle(1, 14);
        fillpoly(6, xy);
      end;
    end;
end;

begin
  gd := 0;
  initgraph(gd, gm, "");
  settxtstyle(0, 0, 4);
  setcolor(14);
  outtextxy(136, 44, ':');
  outtextxy(256, 44, ':');
  setcolor(4);
  h1 := 100;
  m1 := 100;
  s1 := 100;
  repeat
    gettime(h2, m2, s2, hs2);
    if h1 <> h2 then begin
      k := h2 div 10; cifra(50, 30, k);
    end;
  until h1 = h2;
end;

```

```
k:=h2 mod 10; cifra(100,30,k);
h1:=h2;
end;
if m1<>m2 then begin
k:=m2 div 10; cifra(170,30,k);
k:=m2 mod 10; cifra(220,30,k);
m1:=m2;
end;
if s1<>s2 then begin
k:=s2 div 10; cifra(290,30,k);
k:=s2 mod 10; cifra(340,30,k);    s1:=s2;
end;
until KeyPressed;
closegraph;
end.
```

(Обязательное)
Пример пользовательского интерфейса

Приведенный фрагмент программы является реализацией меню с возможностью перехода по кнопкам с помощью клавиш стрелок и «Enter». Фрагмент реализован с помощью процедур, но в принципе можно обойтись и без них, включив необходимые блоки в тело основной программы.

Меню состоит из пяти работающих кнопок.

При запуске программы, включается процедура `beginning`. Эта процедура запускает процедуру `menu`. В процедуре меню прорисованы управляющие кнопки программы. Далее управление передается процедуре `doing`, находящейся в цикле, который ожидает нажатия кнопки `ENTER`. Эта процедура действует, в зависимости, от выбора кнопки, после нажатия `ENTER`.

При нажатии на любую из кнопок, процедура `doing` вызывает, определенный в ней фрагмент программы. Управление кнопками осуществляется через процедуру `doing`, вложенную в цикл ожидания нажатия кнопок.

procedure beginning;

```
begin
  setbkcolor(blue);
  setcolor(black);
  setfillstyle(1, cs);
  cleardevice;
  bar(100, 0, 639, 430);
  setfillstyle(1, white);
  setcolor(blue);
  setbkcolor(blue);
  for k:=1 to 4 do menu(k);
end;
```

procedure menu(h: integer);

```
begin
  case h of
  1: begin
    bar(10, 50, 90, 74);
    settextstyle(0, 0, 1);
    outtextxy(15, 52, 'Text');
    end;
  2: begin
    bar(10, 90, 90, 114);
    settextstyle(0, 0, 1);
    outtextxy(15, 92, 'Text');
    end;
```

```

3: begin
  bar(10, 130, 90, 154);
  settextstyle(0, 0, 1);
  outtextxy(15, 132, 'Text');
end;
4: begin
  bar(10, 170, 90, 194);
  settextstyle(0, 0, 1);
  outtextxy(15, 172, 'Text');
end;
5: begin
  bar(10, 210, 90, 234);
  settextstyle(0, 0, 1);
  outtextxy(15, 221, 'Text');
end;
end;

```

procedure doing(i:integer);

```

begin
  case i of
    1: begin
      cleardevice;
      setbkcolor(blue);
      setcolor(white);
      for k:=1 to 4 do menu(k);
      setfillstyle(1, red);
      setcolor(red);
      gd1:=0; gm1:=0; gg1:=0;
      setbkcolor(blue);
      setcolor(white);
      setfillstyle(1, red);
      setcolor(red);
      settextstyle(0, 0, 2);
      outtextxy(150, 100, ', ');
      setcolor(white);
      settextstyle(0,0,0);
      repeat
        bar(170, 150, 330, 180);
        outtextxy(175, 165, 'ИГ-М');
        gotoxy(27, 11); read(gd1);
        until (gd1<=31) and (gd1>0);
        repeat
          bar(170, 210, 330, 245);
          outtextxy(175, 225, 'ИГПЖ');
          gotoxy(29, 15); read(gm1);
          until (gm1<=12) and (gm1>0);
        repeat
          bar(170, 275, 330, 305);
          outtextxy(175, 290, 'J@');
          gotoxy(28, 19); read(gg1);
          until (gg1>1900);
          setcolor(blue);
          setfillstyle(1, white);
          for k:=1 to 5 do menu(k);

```

```

end;

2: begin
  ras4et(gd1, gm1, gg1);
  setcolor(blue);
  setfillstyle(1, white);
  for k:=1 to 5 do menu(k);
end;

3: begin
  helps;
  setcolor(blue);
  setfillstyle(1, white);
  for k:=1 to 4 do menu(k);
end;

4: halt(1);{Stop run programe and do in OS}

5: begin
  ras1(gd1, gm1, gg1);
  setcolor(blue);
  setfillstyle(1, white);
  for k:=1 to 4 do menu(k);
end;
end;
end;

u:=readkey;
begining;
setcolor(white);
setfillstyle(1, red);
menu(1); mn:=1;
while uuuu<>#27 do
  begin
    nm:=mn;
    repeat until keypressed;
    uuuu:=readkey;
    case uuuu of
      #0: begin
        uuuu:=readkey;
        case uuuu of
          #72: if mn<>1 then {strelka vverx}
            mn:=mn-1
            else
            mn:=4;
          #80: if mn<>5 then {strelka vniz}
            mn:=mn+1
            else
            mn:=1;
        end;
      end;
    end;
  #13: begin
    setfillstyle(1, blue);
    setcolor(white);
    menu(mn);
  end;
end;

```



```
doing(mn);
setfillstyle(1, white);
setcolor(blue);
menu(mn);
setfillstyle(1, white);
setcolor(blue);
menu(mn);
end;
end;
setcolor(blue);
setfillstyle(1, white);
menu (nm); setfillstyle(1, red);
setcolor(white); menu(mn);
end;
```

(Обязательное)

Примеры заданий для практических работ

Практическая работа №1 «Изображение геометрических фигур»

Пример задания:

Нарисовать паровоз.

Текст программы:

```
uses graph;
var
dx, dy, x0, y0:integer;
tr: array[1..15] of PointType;
begin
i:=detect;
  initgraph(i,j,"");
  if graphresult<>grok then halt;
x0:=100; y0:=100;
dx:=5; dy:=5;
tr[1].x:=x0+0*dx;    tr[1].y:=y0+7*dy;
tr[2].x:=x0+0*dx;    tr[2].y:=y0+6*dy;
tr[3].x:=x0+1*dx;    tr[3].y:=y0+6*dy;
tr[4].x:=x0+1*dx;    tr[4].y:=y0+3*dy;
tr[5].x:=x0+2*dx;    tr[5].y:=y0+3*dy;
tr[6].x:=x0+2*dx;    tr[6].y:=y0+0*dy;
tr[7].x:=x0+3*dx;    tr[7].y:=y0+0*dy;
tr[8].x:=x0+3*dx;    tr[8].y:=y0+3*dy;
tr[9].x:=x0+7*dx;    tr[9].y:=y0+3*dy;
tr[10].x:=x0+7*dx;   tr[10].y:=y0+1*dy;
tr[11].x:=x0+13*dx;  tr[11].y:=y0+1*dy;
tr[12].x:=x0+13*dx;  tr[12].y:=y0+2*dy;
tr[13].x:=x0+12*dx;  tr[13].y:=y0+2*dy;
tr[14].x:=x0+12*dx;  tr[14].y:=y0+7*dy;
tr[15].x:=x0+0*dx;   tr[15].y:=y0+7*dy;

Rectangle (x0+8*dx, y0+2*dy, x0+10*dx, y0+4*dy);
Setfillstyle(SolidFill, Red);
SetColor(Red);
PieSlise(x0+3*dx, y0+7*dy, 0, 360, 1+dx);
PieSlise(x0+6*dx, y0+7*dy, 0, 360, 1+dx);
PieSlise(x0+9*dx, y0+7*dy, 0, 360, 1+dx);
SetColor(White);
Circle(x0+3*dx, y0+7*dy, 1*dx);
Circle(x0+6*dx, y0+7*dy, 1*dx);
Circle(x0+9*dx, y0+7*dy, 1*dx);
Readln;
End;
closeGraph;
end.
```

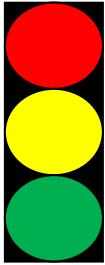
Практическая работа №2 «Раскрашивание рисунков»

Пример задания:

Написать программу, которая выводит на экран изображение светофора.

Текст программы:

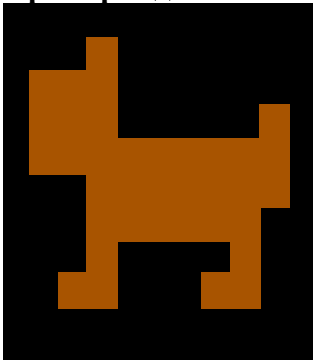
```
Program Svetofor;
Uses Crt, Graph;
Var
  GrDriver, GrMode, GrError, i, j : Integer;
BEGIN
  GrDriver := Detect; InitGraph(GrDriver, GrMode,
```



```
'f:\BP\BGI');  
GrError := GraphResult; If GrError <> GrOk then  
Halt;  
rectangle (88,88,112,152);  
setfillstyle(solidfill, red);  
setcolor(red);  
PieSlice(100,100,0,360,10);  
setcolor(yellow);  
setfillstyle(solidfill, yellow);  
PieSlice(100,120,0,360,10);  
setcolor(green);  
setfillstyle(solidfill, green);  
PieSlice(100,140,0,360,10);  
end.
```

Практическая работа №3 «Создание рисунка в конкретных координатах»

Пример задания:



Текст программы:

```
uses graph, dos, crt;  
var i,j: integer;  
{4}  
{5}  
begin  
i:=detect;  
initgraph(i,j,"");  
if graphresult<>grok then halt;  
Setfillstyle(1,6);  
Bar(10,20,40,50);  
Bar(30,10,40,90);  
Bar(30,40,90,70);  
Bar(90,30,100,60);  
Bar(20,80,40,90);  
Bar(80,70,90,90);  
Bar(70,80,90,90);  
{*****}  
readln;  
closegraph;  
end.
```

Практическая работа №4 «Вывод объекта в относительных координатах»

Пример задания:

Составить программу, выводящую в центре экрана в графическом режиме прямоугольник, который перемещается

Текст программы:

```
uses crt,graph;  
var  
drv,mode:integer;  
  
PROCEDURE move_rect;  
var  
flag_exit,flag_move:boolean;  
k:char;  
x,x_old,y,y_old:word;  
c1,c2:byte;
```

```

BEGIN
x:=100; {начальные координаты прямоугольника}
y:=100;
c1:=9; {цвет прямоугольника}
c2:=0; {цвет фона}
flag_move:=true;
flag_exit:=false;
repeat
if keypressed then
begin
k:=readkey;
if k=#27 then flag_exit:=true; {выход}
if k=#80 then {кнопка вниз}
begin
y_old:=y; {сохранить старые координаты}
x_old:=x;
y:=y+5; {изменить текущие координаты}
flag_move:=true; {показать прямоугольник с
изменениями}
end;
if k=#72 then {кнопка вверх}
begin
y_old:=y;
x_old:=x;
y:=y-5;
flag_move:=true;
end;
{другие кнопки}
end;
{координаты изменились, показать изменения}
if flag_move then
begin
setcolor(c2);
rectangle(x_old,y_old,x_old+100,y_old+50);
setcolor(c1);
rectangle(x,y,x+100,y+50);
flag_move:=false;
end;
until flag_exit;
END;

```

```

BEGIN
drv:=detect;
initgraph(drv,mode,'C:\bp\bgi\');

move_rect;

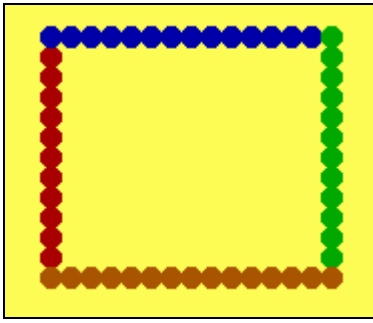
closegraph;
END.

```

Практическая работа №5 «Оператор повторения»

Пример задания:

Текст программы:
PROGRAM gr4;
uses graph, dos, crt;



```

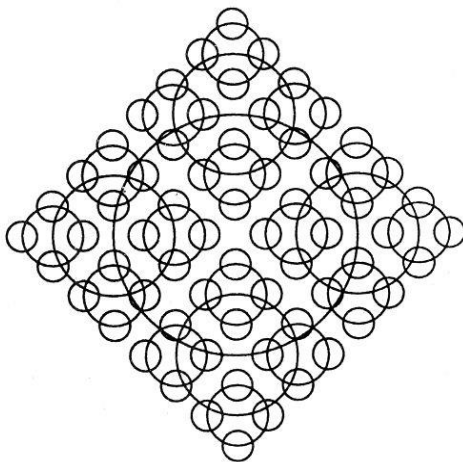
var x,y,i,j: integer;
begin
  i:=detect;
  initgraph(i,j,"");
  if graphresult<>gok then halt;
  Setbkcolor(14);
  x:=40;y:=30;
  Setcolor(4);Setfillstyle(1,4);
  Repeat
    y:=y+10;
    Fillellipse(x,y,5,5);
  Until y>=160;
  x:=30;y:=40;
  Setcolor(1);Setfillstyle(1,1);
  Repeat
    x:=x+10;
    Fillellipse(x,y,5,5);
  Until x>=180;
  x:=180;y:=30;
  Setcolor(2);Setfillstyle(1,2);
  Repeat
    y:=y+10;
    Fillellipse(x,y,5,5);
  Until y>=160;
  x:=30;y:=160;
  Setcolor(6);Setfillstyle(1,6);
  Repeat
    x:=x+10;
    Fillellipse(x,y,5,5);
  Until x>=180;
  readln;
  closegraph;
  end.

```

Практическая работа №6 «Повторяющиеся изображения, рекурсия»

Пример задания:

Вывести на экран узор:



Текст программы:

```

Uses graph, crt;
Procedure Elem(x, y, r, p: integer);
Begin
  If p>=0 then
  Begin
    Circler(x, y, r);
    Delay (100);
    Elem (x+r, y, round(r/2), p-1);
    Elem (x, y-r, round(r/2), p-1);
    Elem (x-r, y, round(r/2), p-1);
    Elem (x, y+r, round(r/2), p-1);
  End;
End;

Begin
gd:= detect;
initgraph( gd, gm, 'c/ bp');
Elem (320, 240, 60, 3);

```

```
Readln;  
End.
```

Практическая работа №7 «Построение графиков функций»

Пример задания:

Построить график функции $y=x^2$

Текст программы:

```
uses graph;  
var gd, gm: integer;  
x, y: real;  
begin  
gd:= detect;  
initgraph( gd, gm, 'c/bp');  
SetColor(5);  
Line(0, 240, 640, 240);  
Line(320, 0, 320, 480);  
x:= -20;  
While x<=20 do  
begin  
y:= -Sqr(x);  
PutPixel( x *5 + 320, y*5 +240, 15);  
x:=x + 0.01;  
end;  
end.
```

Практическая работа №8 «Построение графиков тригонометрических функций»

Пример задания:

Построить график функции $Y=X*\sin(X)$ в интервале от $X=0$ до $X=10$.

Текст программы:

```
uses Graph;  
var  
Gd, Gm: Integer;  
x, y: real;  
begin  
Gd := Detect;  
InitGraph(Gd, Gm, "");  
SetBkColor(1);  
line(50,0,50,480); line(50,200,640,200);  
moveto(50,200);  
x:=0;  
repeat  
y:=x*sin(x);  
lineto(50+round(x*50),100+(100-round(y*20))) ;  
x:=x+0.02;  
until x>10;  
Readln;  
CloseGraph;  
end.
```

Практическая работа №9 «Использование случайных величин при рисовании»

Пример задания:

Вывести на экране в произвольном месте экрана окружности различных диаметра и цвета.

Текст программы:

```
Program Picture_Circle;  
Uses Crt,Graph;  
Const  
  
X_max=600;Y_max=480,Radius_max=50;Color_max=  
15;  
Var
```

```

GrDr,GrMod :integer; {описание переменных граф.
драйвера и режима}
X,Y,R:integer;
begin
GrDr:=Vga; GrMod:=VgaHi;
InitGraph(GrDr,GrMod,'');
Randomize;
While Not KeyPressed do
begin
SetColor(Random(Color_max));
X:= Random(X_max);
Y:=Random(Y_max);
R:= Random(Radius_max);
circle(X,Y,R);
end;
CloseGraph; {закрытие графического режима}
end.

```

Практическая работа №10 «Построение трехмерного изображения»

Пример задания:

Вывести на экран каркасную модель тетраэдра. Тетраэдр задается координатами своих вершин, соединенных отрезками прямых. Вывести тетраэдр в изометрической, косоугольной и диметрической проекции.

Текст программы:

```

x, Z, y, xstep, ystep, theta : Real;
i, j, xold, yold, xnew, ynew : Integer;
P : projection;
procedure draw_tetra:
var
xp, yp : array[1, .4] of Integer;
begin
project(P, 0, 0, 0, 0, 0, xp[1], yp[1]);
project(P, 0, 0, 50, 0, 0, xp[2], yp[2]);
project(P, 0, 50, 0, 0, 0, xp[3], yp[3]);
project(P, 50, 0, 0, 0, 0, xp[4], yp[4]);
norm_line(xp[1], yp[1], xp[2], yp[2], White);
norm_line(xp[1], yp[1], xp[3], yp[3], White);
norm_line(xp[1], yp[1], xp[4], yp[4], White);
norm_line(xp[3], yp[3], xp[4], yp[4], White);
norm_line(xp[2], yp[2], xp[3], yp[3], White);
norm_line(xp[2], yp[2], xp[4], yp[4], White);
end;
begin
abs_x_center := 0; abs_y_center := 0;
open_graph;
compute_isometric_matrix(P);
ClearViewPort;
out_text_XY('Isometric Projection'. -150, 150.
Yellow): draw_tetra;
ReadLn;
theta := 45;
compute_oblique_matrix(theta, P);
ClearViewPort;
out_text_XY('Oblique Projection'. -150, 150.
Yellow): draw_tetra;
ReadLn;

```

```

theta := 45;
repeat
  compute_oblique_matrix(theta. P);
  ClearViewPort;
  draw_tetra;
  theta := theta + 1;
until keypressed;
ReadLn;
theta := 15;
compute_dimetric_matrix(theta. P);
ClearViewPort;
out_text_XY('Dimetric Projection'. -150. 150.
Yellow); draw_tetra;
ReadLn;
theta := 15;
repeat
  compute_dimetric_matrix(theta. P);
  ClearViewPort;
  draw_tetra;
  theta := theta + 1;
until theta >= 45;
ReadLn;
close_graph;
end.

```

Практическая работа №11 «Движение объекта»

Пример задания:

Нарисовать человечка, делающего зарядку.

Текст программы:

```

Program Animation;
Uses Crt, Graph;
Const
  Vert   : Array[1..3] of Integer = (190, 157, 120);
  Horizont : Array[1..3] of Integer = (200, 190, 200);
Var
  GrDriver, GrMode, GrError, i, j : Integer;
BEGIN
  GrDriver := Detect; InitGraph(GrDriver, GrMode,
'C:\TP\BGI');
  GrError := GraphResult; If GrError <> GrOk then
Halt;
  SetColor(LightGray); {установка светло-серого
цвета для рамки}
  Rectangle(20, 20, 480, 400); {рисование рамки}
  SetColor(LightCyan); {установка ярко-голубого
цвета для текста}
  OutTextXY(200, 40, 'П Р И В Е Т !');
  SetColor(LightGray); Circle (250, 130, 20); {голова}
  SetColor(Yellow); Arc(250, 130, 0, 180, 26);
{волосы}
  Arc(250, 130, 0, 180, 24); Arc(250, 130, 0, 180, 22);
  Line(250, 105, 244, 115); Line(250, 105, 250, 116);
{чубчик}
  Line(250, 105, 256, 115);
  SetColor(LightCyan); Circle(241, 125, 4); {левый

```



```

глаз }
  Circle(259, 125, 4); { правый глаз }
  SetColor(LightRed);
  SetFillStyle(SolidFill, LightRed);
  FillEllipse(250, 140, 6, 3); { рот }
  Setcolor(Green);
  Line(250, 152, 250, 220); { туловище }
  Line(250, 220, 210, 290); { левая нога }
  Line(250, 220, 290, 290); { правая нога }
Repeat {цикл прерывается нажатием любой
клавиши}
  For i:= 1 to 3 do {последовательный вывод трех
положений рук;}
    begin {вниз, на уровне плеч, вверх}
      SetColor(LightCyan); Sound(200*i);
      Line(250, 157, Horizont[i], Vert[i]); {левая рука}
      Line(250, 157, 500-Horizont[i], Vert[i]); {правая
рука}
      Delay(300); {задержка}

      SetColor(Black);
      Line(250, 157, Horizont[i], Vert[i]); {левая рука}
      Line(250, 157, 500-Horizont[i], Vert[i]); {правая
рука}
    end
  until Keypressed;

  SetColor(LightCyan);
  Line(250, 157, Horizont[3], Vert[3]); {левая рука
поднята}
  Line(250, 157, 500-Horizont[3], Vert[3]); {правая
рука поднята}

  For i := 1 to 10 do {звуковая трель}
    begin
      Sound(1000);
      Delay(50);
      Sound(1500);
      Delay(50)
    end;

  NoSound; {выключение звука}
  CloseGraph;
END.

```

Практическая работа №12 «Реализация эффектов»

Пример задания:

Нарисовать звезду, которая сверкает и переливается разными цветами.

Текст программы:

```

Program Star;
Uses Crt, Graph;
Const
  TopsStar: Array[1..18] of Integer = (300, 125, 325, 225,
425, 250, 325, 275, 300, 375, 275, 275, 180, 250, 275,
225, 300, 125);

```

```

Var i, j, GrDriver, GrMode : Integer;
BEGIN
  GrDriver := Detect;
  InitGraph(GrDriver, GrMode, 'C:\TP\BGI'); {гбв -
  @ÿС Ја дËзІбЄ@J@aГ|Ë¬ }
  SetTextStyle(DefaultFont, HorizDir, 2);
  OutTextXY(220, 60, 'S T A R ');
  SetTextStyle(DefaultFont, VertDir, 2);
  OutTextXY(140, 150, 'S T A R ');
  SetTextStyle(DefaultFont, VertDir, 2);
  OutTextXY(500, 150, 'S T A R ');

  i:=0;
  Repeat
    j:=i mod 12;
    SetFillStyle(j, Random(13));
    FillPoly(9, TopsStar);
    Inc(i);
    Delay(500)
  until KeyPressed
  CloseGraph
END.

```

Практическая работа №13 «Работа с текстом в графическом режиме»

Пример задания:

Экспериментальная работа с текстом, шрифтами, размерами шрифтов. Создать любую надпись.

Текст программы:

```

Program Svetofor;
  Uses Crt, Graph;
  Var
    GrDriver, GrMode, GrError, i, j : Integer;
  BEGIN
    GrDriver := Detect; InitGraph(GrDriver, GrMode,
    'f:\BP\BGI');
    GrError := GraphResult; If GrError <> GrOk then
    Halt;
    For i:=1 to 10 do
      begin
        SetBkcolor(i);
        SetTextStyle(i, 0, i);
        OutTextXY (100, 100, 'GRAPHIKS');
        OutTextXY (100, 150, 'IN PASCAL');
        Setcolor (1);
        SetTextStyle(0, 0, 1);
        OutTextXY (460, 460, 'Press any key...');
        readln;
        cleardevice;
        end;
    CloseGraph;
  end.

```

Практическая работа №14 «Создание графического интерфейса»

Пример задания:

Создать заставку (общее творческое задание для всех).

Текст программы:

Программа демонстрирует заставку, созданную в графическом режиме для программы, обрабатывающей числа. Здесь появляются

произвольные числа в графическом режиме со случайными координатами и случайным цветом.

```
uses graph, crt;
type
massive=array[1..9] of integer;
var
gd, gm, i, c, x, y, f, p:integer;
s: string;
k:massive;
begin
gd:=vga;
gm:=vgahi;
InitGraph (gd, gm,'d:\bp\bgi');
SetBkcolor(7);
for i:=1 to 9 do k[i]:=i;
randomize;
for i:=1 to 1000 do
begin
c:=random(15); setcolor(c);
x:=random(GetMaxX);
y:=random(GetMaxY);
f:=random(k[9]);
str(f, s);
OutTextXY(x, y, s);
delay(1000);
end;
Setcolor (4);

SetTextStyle(4, 0, 7);
OutTextXY (180, 100, 'GOROSKOP');
OutTextXY (184, 160, 'PIFAGORA');
Setcolor (1);
SetTextStyle(0, 0, 1);
OutTextXY (460, 460, 'Press any key...');
repeat until KeyPressed;
CloseGraph;
end.
```