Лицей 393

Проект по программированию

Методические материалы

Санкт-Петербург
2008

Проект по программированию. Методические материалы/ Сост. Зеленина С.Б., Лебедева Е.В.: ГОУ Лицей 393. СПб., 2008. 32с.

Методические указания разработаны в помощь учащимся девятых и десятых классов лицея, а также родителям, для облегчения контроля за успешной учебой.
Содержат все основные требования к проекту, перечислен сстав проекта, даны сроки выполнения и сдачи, критерии оценивания различных частей проекта, рекомендации по выполнению и защите. В сборнике приведен фрагмент реального проекта и много небольших программ с примерами алгоритмов, которые могут пригодиться при работе над собственным проектом.

```
Dop.txt
История колли
История точного происхождения колли потеряна в туманах
времени. Эта порода была выведена в жестких условиях
окружающей среды: суровый климат и относительная бедность
животноводов вынуждала их проводить строгий отбор, фермер
не мог позволить себе содержать бесполезных животных.
...
(весь текст занимает ровно экран и полностью здесь не
приводится)

test.txt
1) На что похожа морда, если смотреть спереди или
сбоку?
3
2
1) на штык
2) на затупленный клин
3) на морду бульдога
2) Какими кажутся колли при первой встрече?
3
3
1) неуклюжими
2) с ужасными пропорциями тела
3) сильными и активными
(всего вопросов 12, здесь полностью они не приводятся)
```

Оглавление

Введение .....

Требования к проекту .....

Дневник проекта .....

Этапы выполнения проекта .....

Критерии оценивания .....
- Защита .....
- Программа .....
- Документация .....

Документация .....
- Руководство оператора .....
- Руководство программиста .....
- Требования к оформлению документов .....

Как организовать работу над проектом .....
- Выбираем тему .....
- Планируем работу .....
- Структура проекта .....
- Несколько полезных советов .....
- Подготовка к защите или
  Как выразить уважение к аудитории .....

В помощь при написании программы на языке Паскаль .....
- Работа со стрелками .....
- «Самостоятельное» движение объекта .....
- Меню .....
- Время .....
- Пишем по-русски .....
- Мышь .....
- Шаблон теста .....

Фрагмент проекта .....
- Выбор темы .....
- Структура проекта .....
- Программа .....
- Текстовые файлы проекта .....

Введение

Проект даст Вам многое, Вы научитесь:

→ самостоятельно формулировать цели и задачи;
→ получать удовлетворение от творчества;
→ мыслить структурно;
→ планировать деятельность;
→ доводить дело до конца;
→ представлять свою работу на публике;
→ преодолевать проблемы;
→ и, наконец, писать программы на языке Паскаль!

Мы надеемся, что эти указания помогут Вам преодолеть все трудности, защитить свой ПРОЕКТ и получить глубокое удовлетворение от того, что ВСЕ у ВАС ПОЛУЧИЛОСЬ!!!

Требования к проекту

Проект по программированию состоит из программы на языке Паскаль и документации к ней. Тема проекта — любая, но согласованная с преподавателем. Сроки выполнения: декабрь-март для учащихся 9 классов и декабрь-апрель для учащихся 10 классов. Этапы выполнения проекта документируются в дневнике проекта. Проверка выполнения этапов проекта производится в установленные для каждого этапа сроки (при задержке сдачи очередного этапа может быть выставлена неудовлетворительная текущая оценка). Проект должен быть защищен перед комиссией.

Дневник проекта

Тетрадь (любая), подписанная фамилией автора проекта (в случае если авторов двое — дневник один на двоих). Первая страница содержит таблицу с перечислением этапов выполнения проекта, сроком сдачи этапа и графой под подпись преподавателя, принявшего этап. Дальше в дневнике должны быть отражены результаты выполнения первых трех этапов.

Этапы выполнения проекта

1. **Выбор темы.** Срок выполнения — конец второй четверти (первого полугодия). В дневнике проекта должно быть написано, какой именно проект собирается реализовать учащийся (учащиеся). Недостаточно указать только название проекта. Например, неприемлемо в качестве темы указать «Игра», или «Тест», или «Игра в крестики-нолики». Указание темы должно


```
   textcolor(red);
   gotoxy(9,12);
   writeln('Мне конечно же жаль Вас,
                но Вам придётся');
   writeln('повторить своё обучение. Sorry!'); readkey;
end;
until mark>=3;

 textbackground(3);
 clrscr;
 textcolor(yellow);
 gotoxy(13,12);
 writeln('Поздравляю! Вы прошли курс на',mark);
 writeln('Спасибо за пользование моей программой! Даня
Егоров');
 readkey;
end.
```

Текстовые файлы проекта

```
Kolly.txt
Физическое строение тела колли основано на силе и
активности, без неуклюжести и каких-либо следов
грубости. Этот вид достигается совершенными пропорциями
тела. Качество головы исключительно важно, она должна
рассматриваться в соотношении с размером собаки. При
взгляде спереди или сбоку голова напоминает хорошо
затупленный, чистый клин с гладкими очертаниями. ...
(весь текст занимает ровно экран и полностью здесь не
приводится)

Kolly2.txt
Корпус слегка длинный относительно высоты в холке,
спина крепкая с легким  подъемом в пояснице. Ребра
хорошо округленные. Грудная клетка глубокая, достаточно
широкая за лопатками. Задние ноги мускулистые в бедрах,
сухие и  жилистые ниже, с хорошо выраженным коленом.
Скакательные суставы мощные, хорошо опущенные. Лапы
овальные, с упругими подушечками. ...
(весь текст занимает ровно экран и полностью здесь не
приводится)
```

```
  repeat
    writeln('Теперь понятно? (Y/N)');
    readln(ans14);
    if (ans14='L')or(ans14='l')or(ans14='Д')
               or(ans14='д')
        then begin writeln('Тогда пройдите тест!');
               ans14:='y'; end
    else
    begin
         writeln('Хоть Вы и не понимаете,
                        но придется пройти тест!');
          ans14:='n';
    end;
  until (ans14='Y')or(ans14='y')or(ans14='H')
         or(ans14='н')or(ans14='N')or(ans14='n')
         or(ans14='т')or(ans14='T');
end;

if (ans13='y')or(ans14='y')or(ans14='n') then begin
  clrscr;
  reset(t4);
  for k:=1 to 12 do begin
    readln(t4,s); writeln(s);
    readln(t4,n); readln(t4,nr);
    for i:=1 to n do begin
      readln(t4,s);
      writeln(s);
    end;
    repeat
      readln(num);
    until (num>=1)and(num<=n);
    if num=nr then r:=r+1;
  end;
end;

if (r>=10) then mark:=5;
if (r>=7)and(r<=9) then mark:=4;
if (r>=5)and(r<=6) then mark:=3;
if (r<5) then mark:=2;
if mark=2 then begin
  textbackground(black);
  clrscr;
```


давать достаточно полное представление о задуманном, например: «Игра в крестики-нолики с компьютером на поле три на три с использованием мыши и графических возможностей языка Паскаль» или «Игра в крестики-нолики на поле 20 на 20 для двух игроков в текстовом режиме с сохранением результатов и проведением турнира в 100 партий».

2. **Описание функционирования программы** — подробное описание предполагаемой работы программы, включающее в себя описание всех режимов работы и всех способов управления программой с точки зрения пользователя (всех управляющих клавиш и клавиатурных комбинаций и т.д.). Описание уместно начать словами: «После запуска программы Вы увидите...», далее уместны обороты: «Управление курсором осуществляется курсорными стрелками вниз-вверх... выбор пункта меню осуществляется нажатием клавиши Enter...» и т.п. Срок сдачи — вторая неделя третьей четверти (второго полугодия).

3. **Описание структуры программы** — перечисление всех блоков программы (подпрограмм) с описанием основных структур данных и порядка выполнения блоков. Срок сдачи — третья неделя третьей четверти.

4. **Предварительная сдача программы.** Программа должна быть завершена более чем наполовину (допустимо наличие ошибок, неустойчивость в работе, отсутствие некоторых вариантов работы). Срок сдачи — для девятиклассников третья неделя февраля, для десятиклассников — конец февраля.

5. **Оформление документации.** Девятиклассники оформляют «Руководство пользователя», десятиклассники — «Руководство пользователя» и «Руководство программиста». Требования к документам приведены далее. Срок — до дня защиты.

6. **Защита проекта.** Девятиклассники — на предпоследней неделе третьей четверти. Десятиклассники — на первой неделе четвертой. На защиту необходимо представить: исходный текст программы, все необходимые для работы программы файлы (модули, текстовые файлы, шрифты и пр.), документацию проекта (дневник проекта и «Руководства»). Для защиты проекта ученику предлагается выступить в течение 5 минут и доказать четыре основных положения:

программа выполняет именно то, что было задумано;
программа работает устойчиво;
программа полезна обществу;
программа написана ЛИЧНО тем, кто ее защищает.

После выступления автор отвечает на вопросы комиссии и зрителей.

ВНИМАНИЕ! Во время защиты не увлекайтесь демонстрацией программы — это неэффективная трата времени. Как подготовиться к защите — см. далее.

По результатам выполнения проекта ученик получает ТРИ оценки: за саму программу, за документацию к ней и за защиту.

Критерии оценивания

Защита

«Неудовлетворительно» — 2	Отказ от защиты в установленные сроки или неспособность объяснить существенные аспекты работы программы.
«Удовлетворительно» — 3	Затруднения с ответом на вопросы; отсутствие логики выступления; неграмотная речь.
«Хорошо» — 4	Неполное соответствие требованиям на «Отлично».
«Отлично» — 5	Грамотная речь с правильным использованием терминологии; заранее продуманная логика выступления; полнота освещения проекта (не путать с демонстрацией программы, для освещения проекта нужно доказать ЧЕТЫРЕ основных положения, а не показывать все варианты работы программы); соблюдение регламента; свободный ответ на вопросы. Приветствуется наличие презентации.

Программа

«Неудовлетворительно» — 2	Программа заимствована более чем на 75% или неработоспособна в принципе.
«Удовлетворительно» — 3	Программа заимствована более чем на 25%; уровень сложности не соответствует отведенному под выполнение проекта времени.


```
 assign(t3,'dop.txt'); assign(t4,'test.txt');
 textbackground(2); clrscr; textcolor(yellow);
 gotoxy(24,4); writeln('$  $    $$       $$$      $$$   $     $');
 gotoxy(24,5); writeln('$ $    $  $     $  $     $  $   $    $$');
 gotoxy(24,6); writeln('$$    $    $    $  $     $  $   $   $ $');
 gotoxy(24,7); writeln('$$    $    $    $  $     $  $   $   $ $');
 gotoxy(24,8); writeln('$ $   $    $    $  $     $  $   $  $  $');
 gotoxy(24,9); writeln('$  $   $  $    $    $    $    $ $   $');
 gotoxy(24,10);writeln('$   $   $$    $    $   $    $   $$    $');
 textcolor(blue);
 gotoxy(27,14);  write('Здраствуйте!');
 gotoxy(8,16);  write('Эта программа, сделанная учеником
7-1 класса Егоровым Даниилом,');
 gotoxy(11,17);  write('расскажет Вам о колли.');
 readkey;

repeat
 clrscr; reset(t1);
 repeat
   readln(t1,s);
   writeln(' ',s);
 until Eof(t1);{ первый текст }
 writeln('Чтобы читать далее нажмите Enter)'); readln;
 clrscr; reset(t2);
  repeat readln(t2,s); writeln(' ',s); until Eof(t2);
readkey;  { второй текст }
 repeat
   writeln('Понятно??? (Y/N)');
   readln(ans13);
   if (ans13='L')or(ans13='l')or(ans13='Д')
           or(ans13='д')then ans13:='y';
 until (ans13='Y')or(ans13='y')or(ans13='H')
           or(ans13='н') or(ans13='N')
           or(ans13='n')or(ans13='т')or(ans13='T');

 if (ans13<>'y') then begin
   clrscr;
   reset(t3);
   repeat  { третий текст - для непонятливых }
     readln(t3,s);
     writeln(s);
   until Eof(t3);
```

переменная (S), для общения с пользователем — символьные переменные (ans14,ans13).
Алгоритм работы программы и основные составные части:
1. Вывод заставки.
2. Вывод первой половины текста о колли на экран.
3. После нажатия клавиши Enter — вывод второй половины текста.
4. Вопрос к пользователю: «Понятно ли ему?». Если ответ — «да» (несколько вариантов, например «Yes» и т.п.), то переход прямо к тесту, если любой другой ответ, то сначала будет показан дополнительный текст, после которого снова будет задан вопрос «Понятно или нет?», но от ответа на него ничего не будет зависеть, т.к. больше информации в программе нет.
5. Тест — 12 вопросов, каждый прочитывается из файла, выводится на экран, спрашивается номер правильного ответа. Если ответ пользователя совпадает с правильным — счетчик правильных ответов (R) увеличивается на единицу.
6. Выставляется оценка (меньше 5 правильных ответов «Два», 5-6 — «Три», 7-9 — «Четыре», 10 и больше — «Пять»). Оценка сообщается пользователю и, если оценка больше «Двойки», программа завершает работу, если нет — повторяется с пункта 2.
{Данный проект писал ученик СЕДЬМОГО класса... для более старших учеников ОЧЕНЬ желательно использование подпрограмм и описание порядка их вызова, а не просто порядок выполнения частей программы}

Альтернативный вариант описания структуры

Также структуру программы можно описать с помошью укрупненной блок-схемы (правда описание основных переменных и файлов все равно необходимо). Примерную блок-схему можно видеть на рисунке.

Программа

```
program Kolly;
uses crt;
var k,v,n,nr,i,mark,num,r,x,y:integer;
    s:string;
    ans14,ans13:char;
    c:char;
    t1,t2,t3,t4:text;
begin
 mark:=0;
 assign(t1,'kolly.txt'); assign(t2,'kolly2.txt');
```


«Хорошо» — 4	Авторская разработка, основанная на материалах уроков.
«Отлично» — 5	Полностью самостоятельная разработка (использование отдельных модулей и библиотек допустимо с указанием источника кода) соответствующего уровня сложности; использование дополнительных материалов по алгоритмам и технологиям программирования; соблюдение принципов структурного программирования, грамотное оформление текста программы (в том числе — использование комментариев); грамотное использование структур данных.

Документация

«Неудовлетворительно» — 2	Отсутствует на момент защиты, заимствована или полностью не соответствует требованиям по оформлению и содержанию.
«Удовлетворительно» — 3	Не соответствует требованиям по оформлению; не содержит существенной части информации о проекте.
«Хорошо» — 4	Частично соблюдены требования по оформлению; информация о проекте изложена неполно или неграмотным языком.
«Отлично» — 5	Полное соответствие требованиям по содержанию и оформлению документов; информация изложена корректным языком и полностью освещает все требуемые аспекты проекта.

Документация

Руководство оператора

1. Титульный лист (наименование учреждения — вверху, название документа, название проекта, имена разработчиков, руководителя, Санкт-Петербург, год).

2. Аннотация (что можно найти в данном документе, ОЧЕНЬ кратко – назначение программы).
3. Содержание (оглавление с указанием номеров страниц).
4. Возможности использования программы (примерно соответствует разделу «Выбор темы» в дневнике проекта).
5. Описание интерфейса (примерно соответствует разделу «Описание функционирования» в дневнике проекта).
6. Описание размещения (полный перечень файлов, входящих в состав проекта, а также указание папки и диска, если это влияет на работу программы).
7. Требования к программным и аппаратным средствам (указать язык программирования и версию среды разработки программ, с помощью которой разрабатывался проект, версию офисных программ, с помощью которых подготовлена документация, операционную систему и краткие характеристики компьютера, на котором проект ТОЧНО работает).

Руководство программиста

1. Титульный лист.
2. Аннотация.
3. Содержание.
4. Постановка задачи (возможности использования программы + описание интерфейса, но не с точки зрения пользователя, а с точки зрения программиста — четко, полно и конкретно).
5. Формализация алгоритма (перечень подпрограмм с описанием их назначения, блок-схема или подробное словесное описание алгоритма), , примерно соответствует разделу «Структура проекта» в дневнике проекта.
6. Листинг программы (текст программы).
7. Тестовые примеры (различные режимы функционирования программы — копии экрана, можно фотографии).
8. Описание размещения.
9. Требования к программным и аппаратным средствам.

Требования к оформлению документов

1. Единство шрифтового и стилевого оформления всех разделов.
2. Соблюдение правил верстки:
   a) отсутствие «лишних» пробелов и «лишних» абзацев,
   b) правильное использование знаков препинания, сокращений и т.п.,
   c) отсутствие «висячих» строк,
   d) использование стилей для выделения заголовков,

текст, рассказывающий о колли, потом он сможет при желании прочесть дополнительный текст, а затем ему придется ответить на тест. Если результат теста будет неудовлетворительным, пользователю ПРИДЕТСЯ повторить все сначала, и так до тех пор, пока он не сможет пройти тест.

Структура проекта

Основные данные проекта хранятся в текстовых файлах:
kolly.txt — первая половина текста о породе колли;
kolly2.txt — вторая половина текста о породе (текст разбит на две части для удобства вывода на экран);
dop.txt — дополнительный текст для тех, кто не уверен, что все понял;
test.txt — тестовые вопросы и ответы. Вопросы хранятся в формате: первая строка — сам вопрос, вторая строка — целое число не больше 4 — количество вариантов ответов, третья строка — номер правильного ответа, затем в отдельных строках варианты ответов.
Перечислим ОСНОВНЫЕ переменные программы.
Для работы с файлами необходимы файловые переменные (t1,t2,t3,t4: text)
Для работы программы необходимо подсчитывать количество правильных ответов (целочисленная переменная R), для работы с вопросами понадобится строковая

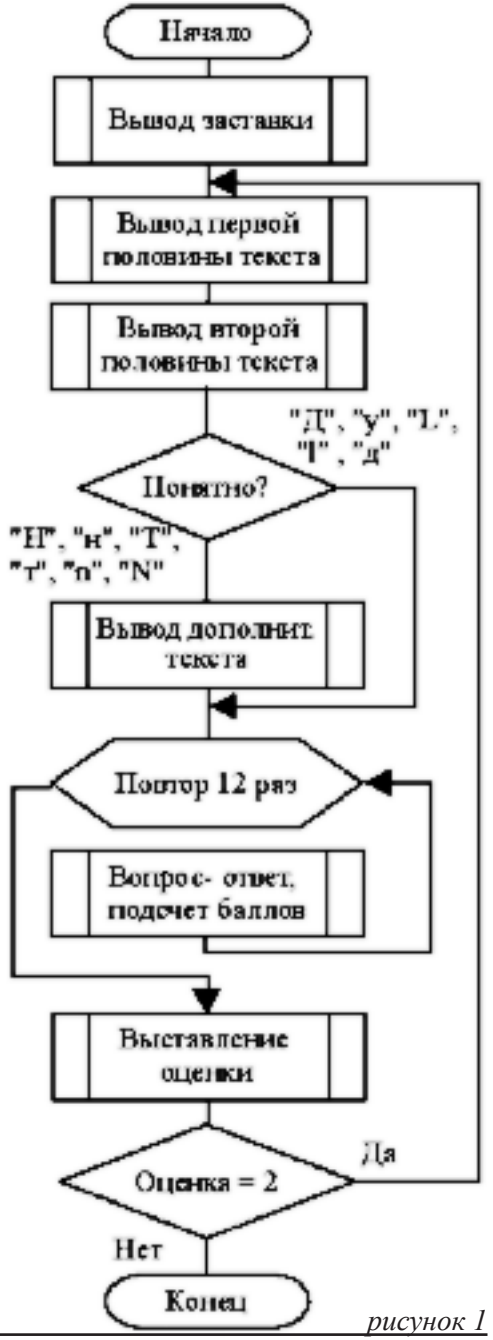

рисунок 1

```
      gotoxy(5,9);
      write('3. Английском языке');
      gotoxy(5,10);
      write('4. Нет правильного ответа');
      gotoxy(15,15);
      repeat
         readln(num);
      until (num>=1) and (num<=4);
      if num=3 then right:=right+1;
      if right=2 then mark:=5
      else if right=1 then mark:=3
           else mark:=2;
      if mark=2 then begin
         textbackground(red);
         clrscr;
         textcolor(black);
         gotoxy(15,10);
        write('Сожалеем, но Вам придется повторить курс
обучения... :(((');
         readkey;
      end;
    until mark>=3;
    textbackground(green);
    clrscr;
    textcolor(black);
    gotoxy(15,10);
             write('Поздравляем,   Вы   прошли   курс,
                    Ваша оценка ',mark);
    readkey;
end.
```

Фрагмент проекта

Далее приводятся фрагменты реального проекта, выполненного в 2004 году учеником 7 класса. Мы не приводим описание функционирования и документацию, потому что считаем каждого учащегося способным написать их самостоятельно.

Выбор темы

Программа должна познакомить пользователя с основными признаками собак породы колли. Пользователю будет предложено прочесть некий

e) отступы в абзаце через формат абзаца и т.д.,
f) разумные поля на странице...

3. Наличие колонтитула и номеров страниц (кроме первого листа).
4. Сборка оглавления автоматически.
5. Кегль шрифта основного текста 11 пунктов,
6. Каждый раздел – на новой странице.
7. Сквозная нумерация рисуков и таблиц

Как организовать работу над проектом

Выбираем тему

Самое сложное — найти оригинальную идею. НО! Не поленитесь потратить силы на ее поиск. Помните, что простая, но неожиданная программа произведет на комиссию большее впечатление, чем 20-й по порядку морской бой (несмотря на то, что комиссия отдает себе отчет в трудоемкости его написания). Оглядитесь вокруг, вспомните различные игры, головоломки, пересмотрите учебники и конспекты, задайте себе вопрос — что еще не умеет делать мой компьютер?

> *ОБЯЗАТЕЛЬНО! Тему согласуйте с преподавателем, чтобы не было разочарования, когда выяснится, что эта задача считается слишком простой и может претендовать только на «тройку» или что в вашем классе уже несколько человек взялись за разработку такой темы. Заодно консультация преподавателя поможет вам избежать чрезмерной сложности проекта — более опытный человек может увидеть проблемы там, где Вы их заметите слишком поздно.*

Планируем работу

→ Представьте себе, что Ваша программа уже написана. ОЧЕНЬ подробно представьте.

→ Опишите, как она выглядит — буквально каждый экран, каждый режим, лучше даже нарисуйте с соблюдением цвета. Ответьте на вопрос — будете ли Вы использовать текстовый режим или графический, или тот и другой по очереди.

→ Опишите, как управляется программа — перечислите буквально каждую кнопочку, на которую реагирует программа в каждом режиме. Решите, будете ли Вы использовать мышь.

→ Перечислите, какую информацию требует для своей работы программа и откуда она ее берет — с клавиатуры, из файла и т.п.

→ Продумайте, какую информацию выдает программа — текстовую, числовую (на экран, в файл и т.п.), графическую, звуковую...

→ Осознайте, нужно ли Вам хранение информации между запусками программы (имена пользователей, рекорды, режимы работы и т.д.). Если да, то в каком виде должна быть сохранена информация.

→ Осознайте, нужны ли вашей программе временные задержки. Если да, то насколько важно точное соблюдение времени.

→ Проанализируйте, нужен ли Вашей программе искусственный интеллект. Если да, то представляете ли Вы алгоритм его работы.

Все, что выше перечислено, необходимо записать в дневник проекта в раздел «Описание функционирования».

→ После того, как Вы все это напишете, ВНИМАТЕЛЬНО прочтите свои записи и осознайте, достаточно ли Ваших знаний для реализации задуманного. Если нет, отметьте места, где Ваших знаний недостает, и примите решение, будете ли Вы пополнять свои знания в этой области (с помощью преподавателя, учебной литературы, материалов Интернет, чьих-либо советов, системы подсказок среды программирования) или пересмотрите свое представление о проекте с учетом недостатка ваших знаний.

→ Если Вы пересматриваете свое представление о проекте, обязательно повторите все действия по планированию работы с самого начала.

Планирование завершено, когда Вы четко осознаете, что Ваших знаний в принципе достаточно, чтобы реализовать все аспекты работы проекта.

Структура проекта

После того как Вы представили себе внешний вид программы — переходим к анализу внутренней структуры.

→ Опишите ВСЕ СУЩЕСТВЕННЫЕ для работы программы ДАННЫЕ (имя, тип). Например, где Вы будете хранить игровое поле или вопросы теста, где подсчитываются очки и фиксируется имя пользователя.

→ Перечислите крупные блоки программы. Например, «Вывод заставки», «Ввод имени пользователя», «Вывод игрового поля», «Проверка правильности ответа» и т.п. По возможности сопоставьте им подпрограммы (процедуры


```
                clrscr;
                textcolor(red);
                gotoxy(10,5);
                 write('hello - читается хело или хелоу -
означает привет по-английски');
                gotoxy(20,7);
                write('Больше добавить нечего :} ');
                textcolor(blue);
                gotoxy(10,22);
                write('Теперь понятно? (y/n) ');
                readln(ans2);
               if (ans2='y') or (ans2='Y') or (ans2='н')
or (ans2='H') then ans2:='y';
                rep:=rep+1;
            until (ans2='y') or (rep>2);
          end;
       until (ans1='y') or (ans2='y');
       textbackground(white);
       right:=0;
       clrscr;
       gotoxy(15,5);
       write(' Hello - это:');
       gotoxy(5,7);
       write('1. Привет');
       gotoxy(5,8);
       write('2. Пока');
       gotoxy(5,9);
       write('3. Извините');
       gotoxy(5,10);
       write('4. Нет правильного ответа');
       gotoxy(15,15);
       repeat
          readln(num);
       until (num>=1) and (num<=4);
       if num=1 then right:=right+1;
       clrscr;
       gotoxy(15,5);
       write(' Hello - это привет на:');
       gotoxy(5,7);
       write('1. Китайском языке');
       gotoxy(5,8);
       write('2. Русском языке');
```

Шаблон теста

```
{Жирным выделены фрагменты программы }
{МИНИМАЛЬНО необходимые для ее работы}
program primer_test;
uses crt;
var
     ans1, ans2 : char;
     num,rep : integer;
     mark,right : integer;
     x,y : integer;
begin
    textbackground(black);
    clrscr;
    for y:=1 to 24 do begin
       x:=3*y;
       textcolor(240+y mod 15);
       gotoxy(x,y);
       write('hello');
    end;
    textcolor(red);
    gotoxy(50,5);
    write('Привет, однако!!!');
    readkey;
    repeat
      repeat
        textbackground(black);
        clrscr;
        textcolor(green);
        gotoxy(20,5);
        write('hello - привет (англ.)');
        gotoxy(20,7);
        write('Это, собственно, и весь
                учебный материал :) ');
        textcolor(blue);
        gotoxy(10,22);
        write('Понятно? (y/n) ');
        readln(ans1);
        if (ans1='y') or (ans1='Y') or (ans1='н')
            or (ans1='H') then ans1:='y';
        rep:=0;
        if (ans1<>'y') then begin
           repeat
```


или функции), дайте им имена, перечислите необходимые параметры.

→ Напишите или, еще лучше, нарисуйте с помощью стрелок последовательность выполнения блоков.

→ Укажите все условия ветвлений, выполнения и прекращения циклов. (Имеются в виду «большие» циклы, заметные с точки зрения всей программы. Например, ходы игроков выполняются в цикле до достижения победы или ничьей — такой цикл важен с точки зрения структуры программы, а рисование 20 линий в цикле для изображения игрового поля — не важно на данном этапе. Рисование 20 линий для построения игрового поля с точки зрения структуры программы — это блок «Изображение игрового поля»).

→ Найдите все точки общения с пользователем, все точки прекращения работы программы. Убедитесь, что ни одна линия не ведет «в пустоту», ни один цикл не зацикливается «навсегда».

→ Постарайтесь сделать структуру максимально понятной и простой, избегайте пересечения стрелок, «скачков мысли» при словесном описании. Чем проще будет Ваша структура, тем вероятнее Вы ее правильно запишете на языке программирования.

> *Как Вы уже поняли — структура проекта также должна быть отражена в дневнике проекта.*

Несколько полезных советов

→ Делайте ВСЕ ЗАРАНЕЕ. Те, кто откладывает дело на последний день, лишает себя или результата, или здоровья и душевного спокойствия.

→ Решайте проблемы ПО ОДНОЙ. Не пытайтесь сделать все сразу. При ликвидации пробелов на этапе планирования — разбирайтесь с каждым вопросом отдельно и переходите к следующему, когда будете уверены в достаточном освоении предыдущего.

→ При написании программы — сначала напишите общую структуру, заменяя отдельные фрагменты «заглушками» (например, вместо изображения игрового персонажа — кружок, вместо сохранения таблицы рекордов — сообщение «Вы обязательно войдете в историю, когда я допишу этот фрагмент программы»).

→ СОБЛЮДАЙТЕ СРОКИ сдачи проекта — так Вы избавляете себя от ненужных неудовлетворительных оценок (за задержку сдачи этапов Вам могут поставить «двойку») и заодно — от ненужных подозрений в заимствовании чужих идей и программ.

УЧТИТЕ! Заимствуя в УЧЕБНЫХ целях фрагменты чужих программ (выложенных в открытом доступе в сети Интернет или опубликованных иным способом без указания об ограничении на использование), Вы поступаете приемлемым образом. НО! Всякая попытка выдать чужую работу за свою неэтична и подрывает доверие и уважение к человеку, совершающему такое. Поэтому ОБЯЗАТЕЛЬНО указывайте ИСТОЧНИКИ каждого заимствованного Вами фрагмента.

Подготовка к защите или Как выразить уважение к аудитории

→ Продумайте выступление заранее.

→ Разместите основные тезисы выступления на презентации, НО не читайте с экрана (ибо слушатели тоже умеют читать и им обидно, когда выступающий в этом сомневается) — развивайте излагаемые в тезисах мысли.

→ Избегайте намеков и недомолвок, которые могут обидеть слушателей (например, оборотов «в отличие от некоторых я...» и т.п.). Избегайте самоуничижения и самовосхваления — рассказывайте о своей работе просто и прямо, НЕ СРАВНИВАЯ С ДРУГИМИ.

→ Соблюдайте регламент.

→ Придерживайтесь делового стиля речи, избегайте использования жаргонизмов и просторечных оборотов.

→ При выполнении проекта в паре распределите роли так, чтобы при защите не возникло сомнений в равном участии обоих авторов.

→ Не увлекайтесь демонстрацией работы программы — лучше заготовьте заранее фотографии экрана в самых впечатляющих местах и сразу же — фрагменты кода, реализующие самое интересное и поместите их на презентацию... Так вы сбережете время и избежите лишних вопросов.

В помощь при написании программы на языке Паскаль

Работа со стрелками

```
{Перемещение фигуры по экрану с помощью }
{стрелок управления курсором            }
```



```
function my_str( x : integer ) : string;
var m_s : string;
begin
   str(x,m_s);
   my_str := m_s;
end;

var l_b,r_b : boolean;
    m_x,m_y : integer;
    key     : char;
    s1,s2   : string;
begin
   textcolor(green);
   textbackground(white);
   clrscr;
   writeln('Вас приветствует программа
                        минимальной поддержки мыши');
   if query_mouse=-1
     then begin
       gotoxy(5,5);
       writeln('Извините, но у Вас проблемы -
                        мышь не обнаружена...');
       readkey;
       halt(1);
     end
     else begin
       show_mouse;
       repeat
         get_status_mouse(l_b,r_b,m_x,m_y);
         if l_b then s1:='нажата'
                else s1:='не нажата';
         if r_b then s2:='нажата'
                else s2:='не нажата';
         hide_mouse;
         box(20,2,'x='+my_str(m_x)+' y='+my_str(m_y)
   +' левая клавиша '+s1+' правая клавиша '+S2);
         show_mouse;
         delay(1000);
         if keypressed then key:=readkey;
       until key=chr(27);
     end;
end.
```

```
procedure get_status_mouse
 (var left_button,right_button:boolean; var x,y : inte-
ger);
                          {запрос состояния мыши}
var Regs : Registers;  {Запись со значениями регистров
процессора}
begin
   regs.AX:=3;
   Intr($33, Regs);
   { $33 - мышиное прерывание}
   left_button:=(regs.BX and 1) <> 0;
   right_button:=(regs.BX and 2) <> 0;
   x:=regs.CX div 8 + 1;
   y:=regs.DX div 8 + 1;
end;

{Эта процедура не имеет отношения к мыши - }
{так, украшение вывода                     }
procedure box(lx,uy : integer; s : string);
var x,y,rx,dy,l : integer;
begin
   l:=length(s);
   if (80-lx) < (l+2) then lx:= 80-l-2;
   rx:=80;
   dy:=uy+2;
   for x:=lx+1 to rx-1 do begin
      gotoxy(x,uy);   write('═');
      gotoxy(x,uy+1); write(' ');
      gotoxy(x,dy);   write('═');
   end;
   gotoxy(lx,uy);   write('╔');
   gotoxy(lx,dy);   write('╚');
   gotoxy(rx,uy);   write('╗');
   gotoxy(rx,dy);   write('╝');
   gotoxy(lx,uy+1); write('║');
   gotoxy(rx,uy+1); write('║');
   gotoxy(lx+1,uy+1); write(s);
end;
```



```
program move_by_arrows;
uses crt,graph;
var x,y : integer;
    gd,gm : integer;
    c : char;

procedure paint(x,y,cl:integer);
begin
   setcolor(cl);
   circle(x,y,10); circle(x+30,y,10);
   line(x-10,y,x-20,y-10);
   line(x+40,y,x+50,y-10);
   line(x+10,y,x+20,y);
end;

begin
   gd:=DETECT;
   InitGraph(gd,gm,'c:\tp7\bgi');
   x:=50;
   y:=50;
   repeat
      paint(x,y,white);
            c:=readkey;  {посмотрим,  что  нажато  на
клавиатуре}
      if c=chr(0) then begin    {нажата стрелка}
         paint(x,y,black);     {сотрем изображение}
            c:=readkey; {код стрелки состоит из двух
символов}
         if c=chr(72) then y:=y-1;    {вверх}
         if c=chr(75) then x:=x-1;    {влево}
         if c=chr(77) then x:=x+1;    {вправо}
         if c=chr(80) then y:=y+1;    {вниз}
          paint(x,y,white);   {нарисуем снова в новом
месте}
      end;
   until c=chr(27);  {пока не нажат ESC}
   CloseGraph;
end.
```

«Самостоятельное» движение объекта

```
program ball_second;
uses crt;
{Теперь по экрану движется не графический,   }
{а текстовый объект слева-направо и наоборот }
{и, заодно - неограниченно долго             }
var x,y,dx : integer;
    key : char;
begin
   clrscr;
   key:='y';{Это признак того, что надо продолжать}
   x:=5;
   y:=5;  {Мяч встал на  исходную позицию }
   dx:=1; {И будет двигаться вправо       }
   repeat
      gotoXY(x,y);
      write('o');
      delay(100);
      gotoXY(x,y);
      write(' ');
      x:=x+dx;
      if (x=80) or (x=1) then dx:=-dx;
        {Пора идти в другую сторону}
      if keypressed then key:=readkey;
       {Если клавиша нажата, надо посмотреть - какая}
   until key=chr(27); {Код 27 - это клавиша Esc}
end.
```

Меню

```
{Простое меню с управление курсорными стрелками}
program menu;
uses crt;
var
     num : integer;

procedure first;
begin
   textbackground(white);
   textcolor(red);
```


Мышь

```
{для вызова функции мыши: загрузить регистры,      }
{как записано в описании функции, установить       }
{в регистр AX номер функции, вызвать прерывание $33}
{ $00 - определить наличие мыши}
{ $01 - показать курсор}
{ $02 - спрятать курсор}
{ $03 - спросить состояние мыши (координаты в точках)}

program example_mouse_use;
{Программа не делает ничего полезного - }
{проверяет наличие мыши, выводит ее     }
{координаты и отмечает нажатие клавиш   }
uses DOS, CRT;

function query_mouse : Integer;
var Regs : Registers;
{Запись со значениями регистров процессора}
begin
   regs.AX:=0;
   Intr($33, Regs);
   { $33 - мышиное прерывание}
   if Regs.AX = 0 then query_mouse:=-1
      else query_mouse:=Regs.BX;
{количество клавиш на мыши}
end;

procedure show_mouse; {показать курсор на экране}
var Regs : Registers;
{Запись со значениями регистров процессора}
begin
   regs.AX:=1;
   Intr($33, Regs);   { $33 - мышиное прерывание}
end;

procedure hide_mouse; {скрыть курсор на экране}
var Regs : Registers;
{Запись со значениями регистров процессора}
begin
   regs.AX:=2;
   Intr($33, Regs);   { $33 - мышиное прерывание}
end;
```

коды русских букв строки s и делает возможным вывод на экран текста из сохраненного файла.

```
var f,g:text; s:string; x,k,t,p:byte;
function tr(s:string):string;
var p:byte;
begin
  for p:=1 to length(s) do
    case ord(s[p]) of
      192..239: dec(s[p],64);
      240..255: dec(s[p],16);
      end;
      tr:=s;
end;
begin
  k:=0;
  assign(f,'c:\tp7\data1.txt');
  assign(g,'c:\tp7\data2.txt');
  reset(f); {открываем для чтения файл вопросов}
  reset(g);
{открываем для чтения файл с номерами
                                правильных ответов}
  repeat
    for x:=1 to 4 do
         begin
            readln(f,s);
{читаем из файла 4 строки - вопрос
                                и 3 варианта ответа}
            writeln(tr(s));
{выводим перекодированные строки}
         end;
    readln(t); {вводим с клавиатуры номер ответа}
    readln(g,p);
{вводим из файла номер правильного ответа}
    if t=p then k:=k+1;
  until eof(f) or eof(g);
{продолжаем до конца одного из файлов}
  writeln(k);
{выводим количество правильных ответов}
  readln;
end.
```



```
   clrscr;
   gotoxy(20,10);
   writeln('Это первая из процедур');
   readkey;
end;
procedure second;
begin
   textbackground(white);
   textcolor(blue);
   clrscr;
   gotoxy(20,10);
   writeln('Это вторая из процедур');
   readkey;
end;
procedure by;
begin
   textbackground(black);
   textcolor(yellow);
   clrscr;
   gotoxy(20,10);
   writeln('Bye-bye');
   readkey;
end;

function menu_my : integer;
const ss : array[1..3] of string =
      ('1. Первая процедура','2. Вторая процедура',
       '3. Выход');
var
   m : integer;
   cf,cb : byte;
   x,y : integer;
   c : char;

   procedure out_menu(s : string; x,y : integer;
                              cf,cb : integer);
   begin
      textcolor(cf);
      textbackground(cb);
      gotoxy(x,y);
      write(s);
   end;
```

```
begin
   cf:=green;
   cb:=black;
   x:=10;
   textcolor(cf);
   textbackground(cb);
   clrscr;
   m:=0;
     for y:=11 to 13 do begin
       gotoxy(x,y);
       out_menu(ss[y-10],x,y,cf,cb);
     end;
     y:=13;
     out_menu(ss[y-10],x,y,cb,cf);
   repeat
     c:=readkey; {write(c,'  ',ord(c));readln;}
     if (c >='1') and (c<='3') then begin
        out_menu(ss[y-10],x,y,cf,cb);
        y:=10+ord(c)-ord('0'); {write('y=',y);}
        out_menu(ss[y-10],x,y,cb,cf);
     end
     else if ord(c) = 0 then begin
        c:=readkey;
        out_menu(ss[y-10],x,y,cf,cb);
        case ord(c) of
           72:begin
                 if y=11 then y:=13 else y:=y-1;
              end;
           77:   ;{rigth}
           80:begin
                 if y=13 then y:=11 else y:=y+1;
              end;
           75:   ;{left}
        end; {case}
        out_menu(ss[y-10],x,y,cb,cf);
     end;
     if ord(c) = 13 then m:=y-10;
   until (m > 0) and (m <= 3);
   menu_my:=m;
end;
```



```
begin
  repeat
   num:=menu_my;
   case num of
      1: first;
      2: second;
      3: by;
   end;
  until num = 3;
end.
```

Время

```
{Выжидаем заданое количество секунд - просто так}
uses crt,dos;
var h,h1,m,m1,s,s1,hs,hs1 : word;
    sek : word;
    t0, t1 : longint;
    interval : longint;
begin
  clrscr;
  write('Сколько секунд ждать? ');
  readln(sek);
  gettime(h,m,s,hs);
  writeln('Сейчас ',h,'часов ',m,'минут ',
                s,'секунд ',hs,' сотых секунды');
  t0:=h*3600+m*60+s+round(hs/100);
  repeat
    gettime(h1,m1,s1,hs1);
    t1:=h1*3600+m1*60+s1+round(hs1/100);
    interval := t1-t0;
  until interval >=sek;
  writeln('Сейчас ',h1,'часов ',m1,'минут ',
          s1,'секунд ',hs1,' сотых секунды');
  readkey;
end.
```

Пишем по-русски

Если у вас в Паскале нет русификатора, т.е. Вы не можете печатать русские буквы, рекомендуем текст набрать в Блокноте, сохранить, например, по адресу c:\tp7\data1.txt, а в программу добавить функцию tr(s), которая меняет