

*Волкова Ризиды Анверовна*



## ПРОГРАММИРОВАНИЕ В СРЕДЕ ЛОГО МИРЫ

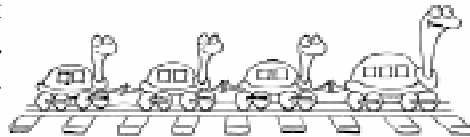
### Часть 4. Многократное выполнение набора

#### КОМАНДЫ ЛОГО перебор, скажи, много\_раз

##### Задача 1. Поезд едет

Написать процедуру, чтобы поезд поехал (одновременно перемещались паровоз и три вагона).

*Подсказка:* используй команду Лого – перебор.



##### Решение:

```
это поезд  
всегда [  
перебор [м [паровоз вагон1 вагон2 вагон3]] [для :м нк 270 вп 5]]  
конец
```

##### Задача 2. Танцующие черепашки

Три черепашки меняют цвета случайным образом и поворачиваются синхронно.

*Подсказка:* используй команды: скажи, нц сл 9999

##### Решение:

```
это танец  
скажи [ч1 ч2 ч3] [нк 0]  
повтори 100 [скажи [ч1 ч2 ч3] [нц 1 + сл 9999 пр 10 жди 1]]  
конец
```

##### Задача 3. Рисуем болото

Зарисовать весь лист проекта камышами (форма черепашки «озеро»).

*Подсказка:* используй команды Лого: штамп

##### Решение:

```
это болото  
перо, нф "озеро нц 92 крась нм [-330 180]  
повтори 10 [  
повтори 6 [штамп нк 180 вп 70 ]  
нов_u 180 нк 90 вп 70  
]  
конец
```



##### Задача 4. Матрешки

Показать содержимое матрешки (Из одной матрешки появляется 5 поменьше). Использовать только одну черепашку.

*Подсказка:* используй команды Лого: **штамп**, **нрз X**.

**Решение:**

```
это матрешки
  пусть "М 80
  перо, нф 61   нм [-260 -70]
  повтори 6 [
    нрз :М штамп нк 99   вп 70 жди 2
    пусть "М :М - 10
  ]
нд 70
конец
```

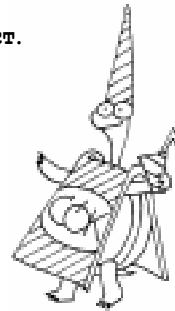
**Задача 5. Рекламные огни**

Сделать рекламный щит с трехбуквенным словом. Для этого: создайте три текстовых окна с именами «текст1», «текст2», «текст3». В каждом окне напишите по большой букве. Текстовые окна должны по очереди то появляться, то исчезать.

*Подсказка:* используй команды Лого: **перебор**, **спрячь\_текст**, **покажи\_текст**.

**Решение:**

```
это огни
повтори 10 [
  перебор [Т [текст1 текст3 текст2]]
  [для :Т спрячь_текст жди 5 покажи_текст жди 2]
]
конец
```



**Задача 6. Падающие буквы**

Сверху вниз по листу проекта перемещаются буквы.

*Подсказка:* используй команды Лого: **установи "текст "место [ ]**, **список :X :У**

**Решение:**

```
это буквопад
  пусть "У 180
  пусть "Х 200
  повтори 9 [перебор [Т [текст1 текст3 текст2]]
    [ установи :Т "место (список :Х :У) пусть "У :У - 15 жди 5]]
конец
```

**Задача 7. Цифры**

Написать процедуру, которая выводит в текстовое окно цифры от 0 до 9.

*Подсказка:* код пробела – 32, **вставь символ 32** – выводит пробел в текстовое окно.

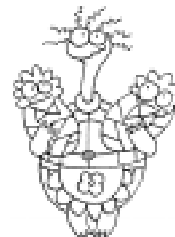
**Решение:**

```
это цифры
все_цифры, ст
много_раз [м 10] [ все_цифры, вставь :м вставь символ 32]
конец
```

**Задача 8. Рождение черепашек**

На листе проекта появляется 15 черепашек с именами – дом0, дом2, ..., дом15. Черепашки принимают форму дома и выстраиваются в линейку. Затем все черепашки удаляются.

*Подсказка:* используй команды Лого: **слово "часть1 "часть2**



**Решение:**

```
это город
много_раз [м 15] [ нч слово "дом :м пч нк 90 вп 60 * :м нф 26] жди 15
много_раз [м 15] [ удали слово "дом :м ]
конец
```

**ЦИКЛИЧЕСКИЙ АЛГОРИТМ**

**ЦИКЛИЧЕСКИЙ АЛГОРИТМ № 1**

**Задача 1-1. Поросенок вырос**

Поросенок кушает и растет до тех пор, пока размер его формы не изменится с 10 до 40.

*Подсказка:* используйте команду Лого: `нрз X`.

**Решение:**

```
это поросенок
пусть "к 10
манька, нрз :к нм [-30 -80]
всегда [
пусть "к :к + 5
нрз :к нф "свин2 жди 3 нф "свин1 жди 3
нк 0 вп 7
если :к > 40 [автостоп]
]
конец
```



**Задача 1-2. Собака в лужу попала**

Собака бежит пока не попадет в лужу.

*Подсказка:* используйте команду Лого `цп - цвет поля`.

**Решение:**

```
это собака
всегда [
пес, нк 270 нф 17 вп 5 жди 2 нф 18 вп 5 жди 2
если цп = 105 [автостоп]
]
конец
```



*2 способ.* Запускается два параллельных процесса

```
это собака2
когда [цп = 105] [останов]
всегда [ пес, нк 270 нф 17 вп 5 жди 2 нф 18 вп 5 жди 2]
конец
```

**Задача 1-3. Котенок делает зарядку**

Котенок делает зарядку, пока не пройдет 100 единиц времени.

*Подсказка:* используйте команды Лого `нов_счет`, `счетчик`.

**Решение:**

```
это зарядка
нов_счет
всегда [
котик, нф 61 жди 5 нф 62 жди 5
если счетчик > 100 [автостоп]
]
конец
```

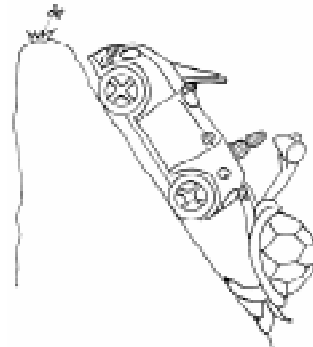


**Задача 1-4. Управляем движением черепашки с пульта**

Автомобиль перемещается по листу проекта. Скорость перемещения автомобиля определяется значением бегунка **скорость**. Направление движения меняется при нажатии мышкой на черепашек в форме стрелок, указывающих направление. Движение начинается при нажатии на кнопку **газ**. Движение прекращается при нажатии на кнопку **тормоз**.

**Решение:**

```
это газ
авто, всегда [ вп скорость жди 2]
конец
это тормоз
останов
конец
```



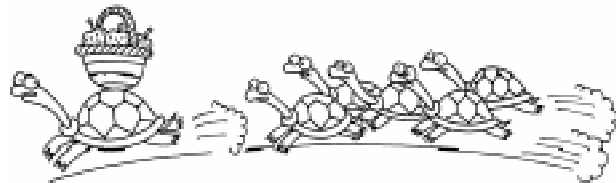
**ЦИКЛИЧЕСКИЙ АЛГОРИТМ № 2**

**Задача 2-1. Проверка делимости на 5**

Проанализировать все числа от 1 до 50 на делимость на 5. В текстовые окна вывести числа, которые делятся на 5 без остатка.

**Решение:**

```
это задача1
пусть "x 1
на_5, ст
всегда [
    если (остаток :x 5) = 0 [на_5, пиши :x]
    пусть "x :x + 1
    если :x = 51 [автостоп]
]
конец
```

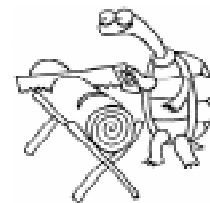


**Задача 2-2. Проверка делимости на 3**

В текстовые окна вывести числа от 1 до 25. Числа кратные 3 должны быть красного цвета, остальные – синего.

**Решение:**

```
это задача2
пусть "x 1
ттт, ст
всегда [
    если_иначе (остаток :x 3) = 0 [нцт 15] [ нцт 105]
    вставь :x вставь символ 32
    если :x > 25 [автостоп]
    пусть "x :x + 1
]
конец
```

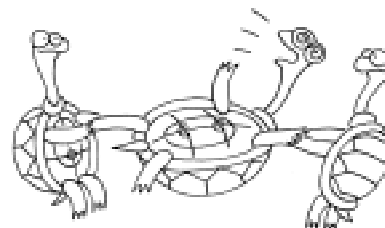


**Задача 2-3. Проверка делимости на 2, 3, 5, 7**

Проанализировать все числа от 1 до 25 на делимость на 2, 3, 5, 7. В текстовые окна вывести соответствующие числа.

**Решение:**

```
это задача3
пусть "x 1
скажи [на_2 на_3 на_5 на_7] [ст]
всегда [
    если (остаток :x 2) = 0 [на_2, пиши :x]
```



```

если (остаток :x 3) = 0 [на_3, пиши :x]
если (остаток :x 5) = 0 [на_5, пиши :x]
если (остаток :x 7) = 0 [на_7, пиши :x]
пусть "x :x + 1
если :x > 25 [автостоп]
]
конец

```

#### Задача 2-4. Код буквы

Написать процедуру, которая в текстовом окне «буква» пишет букву и выводит код этой буквы. Буквы вводятся с клавиатуры. Количество букв 10.

*Подсказка:* используйте команды Лого: жди\_символ, код.

#### Решение:

```

это задача4
пусть "x 0
коды_букв, ст
всегда [
пусть "с жди_символ
коды_букв, пиши (предложение :с «- код :с)
пусть "x :x + 1
если :x > 10 [автостоп]
]
конец

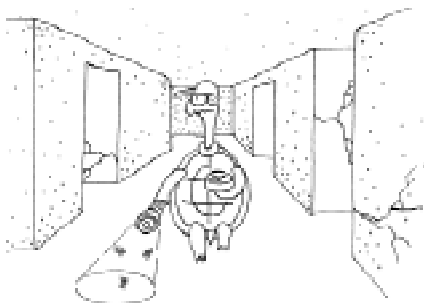
```



## ПРОГРАММИРОВАНИЕ ИГР

### Игра 1. Лабиринт

*Содержание игры:* по лабиринту перемещается черепашка, ее задача – не выходить за границы лабиринта и достичь какой-либо цели, например, зеленого поля.



Для этого:

1. На листе проекта:
  - нарисуйте лабиринт,
  - создайте черепашку,
  - создайте бегунок, который будет определять скорость перемещения черепашки,
  - создайте кнопку, которая будет запускать игру.
2. Напишите процедуру, которая управляет движением черепашки с клавиатуры.

3. Напишите процедуру, которая анализирует цвет поля под черепашкой. Если черепашка вышла за границы лабиринта, она возвращается на начало лабиринта и начинает свое путешествие сначала.

4. Напишите инструкцию к игре.

Можно сделать игру более сложной.

- сделать несколько уровней,
- подсчитывать количество очков или количество жизней,
- сделать тест: при попадании на какой-нибудь цвет, задавать вопросы, анализировать ответ; в зависимости от правильности ответа, пропускать черепашку дальше либо возвращать назад.

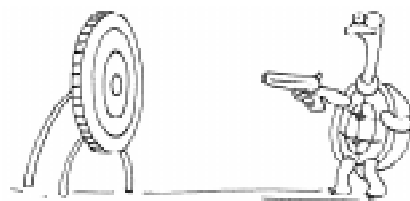
### Игра 2. Мишень

*Содержание игры:* тест на знание курсов черепашки и расстояний на листе проекта. Игрок настраивает бегунки (расстояние до центра мишени и курс черепашки) и произво-

дит выстрел. Пуля летит по заданному курсу и на выбранное расстояние. При попадании в мишень прибавляются очки. После пяти выстрелов выдается результат.

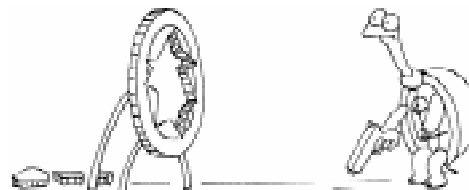
Для реализации этой игры сделайте следующее:

1. Нарисуйте мишень – четыре разноцветных вложенных друг в друга круга.
2. Создайте
  - бегунок – **угол** с диапазоном от 0 до 90,
  - бегунок **дистанция** с диапазоном от 0 до 800,
  - две кнопки «Старт» и «Огонь»,
  - два текстовых окна с именами «Ваши\_очки» и «№попытки»,
  - черепашку с именем **пуля**.
3. В поле форм нарисуйте новую форму для пули.
4. На листе процедур напишите процедуру **старт**, которая будет
  - очищать текстовое окно,
  - определять 2 переменные – **попытки** и **очки** ( начальное значение 0),
  - будет ставить черепашку **пуля** на место и задавать ей форму пули (место выбирается случайным образом; надо только исключить попадание пули на поле мишени).



5. На листе процедур напишите процедуру **огонь**, которая будет

- к переменной **попытки** прибавлять 1,
- черепашка **пуля** будет вставать на начальное место, принимать форму пули, устанавливать курс **угол** (значение бегунка), перемещаться на расстояние **дистанция** (значение бегунка),
- анализировать значение переменной **попытки** и в случае, если значение переменной больше 5, сообщать «Вы исчерпали все попытки!»; затем анализировать количество заработанных очков и, в зависимости от результата, выдавать разные сообщения (на ваше усмотрение) и останавливать выполнение процедуры **огонь**,
- анализировать цвет поля под черепашкой; в зависимости от цвета поля, к переменной **очки** прибавлять разные числа,
- в текстовое окно «Ваши\_очки» выводить значение переменной **очки**, в текстовое окно «№попытки» выводить значение переменной **попытки**,
- черепашке **пуля** менять форму на **флаг**; после небольшой задержки возвращаться на начальное место и принимать форму пули.



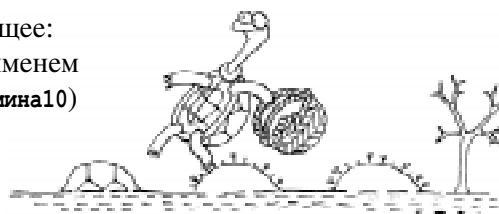
6. Желательно подготовить инструкцию для пользователя на отдельном листе проекта.

### Игра 3. Минное поле

*Содержание игры:* задача игрока – пересечь минное поле, не задев мин. Сапер следует за курсором мышки.

Для реализации этой игры сделайте следующее:

1. На листе проекта создайте черепашку с именем **Сапер** в форме человечка, 10 черепашек (**мина1**, ..., **мина10**) в форме мины.
2. Создайте кнопку «Пошли», которая будет запускать игру.
3. Нарисуйте левую и правую границу минного поля (минное поле и область за границей поля должны быть разных цветов).
4. На листе процедур напишите процедуру **Пошли**, в которой выполняется следующее:
  - a) мины прячутся и размещаются случайным образом на листе проекта;
  - b) черепашка **Сапер** перемещается вслед за указателем мышки;

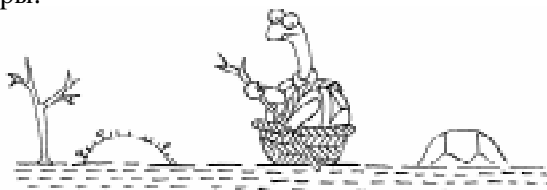


с) анализируются условия окончания игры:

– анализируется цвет поля под черепашкой **Сапер**, в случае, если он равен номеру цвета правой границы минного поля, выдается сообщение «Молодец! Ты победил!» и выполнение процедуры останавливается,

– анализируется расстояние от черепашки **Сапер** до черепашек, изображающих мины; если расстояние меньше 20 единиц, то мины взрываются (показываются черепашки), выдается сообщение «УВЫ! Ты проиграл!» и выполнение процедуры прекращается.

*Подсказка:* Используйте команды Лого: `скажи [ ] [ ]`, `нов_х Х`, `нов_у У`, `сл Х`, `путь "имяЧ, место_мышки`.



#### Игра 4. Прицел

*Цель игры:* проверить знание английских слов. После прочтения задания нужно навести прицел на объект, название которого вы прочитали в задании, и выстрелить по нему.

Для реализации этой игры сделайте следующее:

1. На листе проекта создайте
  - черепашку с именами **прицел** в форме прицела орудия,
  - текстовое окно для вывода заданий (назовите его, как хотите),
  - кнопку «Задание1» (затем «Задание2», «Задание3», ...).
2. На листе процедур напишите процедуру **стрелки**, которая будет управлять прицелом с клавиатуры и задавать цвет поля под прицелом (**ц0**).
3. На листе процедур напишите процедуру **задание1**, в которой выполняется следующее:
  - черепашка **прицел** становится подальше от объектов на экране,
  - определяется переменная **ц0** со значением 0 (цвет поля под черепашкой),
  - в текстовое окно выводится задание (например, «наведи прицел на «ROOF»»),
  - запускается выполнение процедуры **стрелки**,
    - в цикле анализируется значение переменной **ц0**; если значение переменной больше 0, то опять анализируется значение переменной **ц0**; если оно равно номеру цвета поля правильно выбранного объекта, то сообщается «YES», в противном случае сообщается «NO».
4. Оформите программу звуковыми и видеоэффектами, добавьте подсчет очков, выставление отметки – по желанию разработчика.

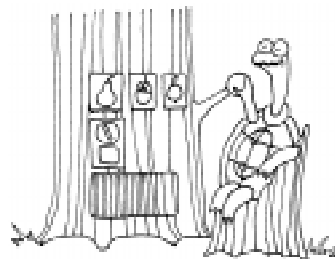


#### Игра 5. Казино – игровой автомат

*Содержание игры:* задача игрока – выиграть крупную сумму денег на игровом автомате. Если во всех трех окошках автомата появились одинаковые картинки – игрок выигрывает 1000 рублей. Если две из трех картинок совпали – игрок получает 20 рублей. Программа также контролирует количество денег у игрока (в начале игры запрашивается сумма денег, имеющаяся у игрока в наличии).

Для реализации этой игры сделайте следующее:

1. На листе проекта
  - нарисуйте игровой автомат с тремя окошками,



- создайте три черепашки с именами  $\phi 1$ ,  $\phi 2$ ,  $\phi 3$  и разместите их в окошках автомата,
- создайте черепашку в форме ручки автомата.

2. Напишите процедуру **начнем**, в которой запрашивается сумма денег, имеющаяся в наличии у игрока.

3. Напишите процедуру **включение\_автомата** (вызывается при нажатии на ручку автомата), в которой выполняется следующее:

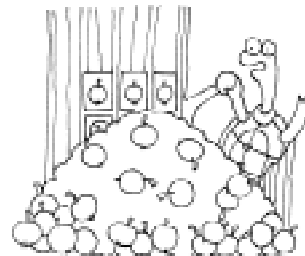
- все три черепашки меняют форму от 1 до 10-го номера случайным образом,
- из суммы денег игрока вычитается 10 рублей (стоимость одного запуска автомата),
- запускается процедура **проверка**.

4. Напишите процедуру **проверка**, в которой выполняется следующее:

- определяются три переменные  $n1$ ,  $n2$ ,  $n3$ , которым присваиваются номера форм черепашек  $\phi 1$ ,  $\phi 2$ ,  $\phi 3$ ,
- сравниваются друг с другом переменные  $n1$ ,  $n2$ ,  $n3$ ;

в случае, если они равны друг другу, выдается сообщение «Вы выиграли 1000 рублей»; если совпадут две переменные из трех, выдается сообщение «Вы выиграли 20 рублей»; в противном случае выдается сообщение «Вы проиграли»,

- к сумме денег игрока прибавляется выигранная сумма.



### Игра 6. Казино – бросаем кости

*Содержание игры:* два игрока по очереди бросают кости. Выигрывает тот, кто раньше наберет 50 очков (но не более). Тот игрок, у которого оказывается более 50 очков, проигрывает.

Для реализации этой игры сделайте следующее:

1. На листе проекта создайте

а) текстовые окна:

– «бр1», «бр2» – в эти окна будут выводиться суммы очков, набранные за 1 бросок;

– «сумма1», «сумма2» – в эти окна будут выводиться общие суммы очков;

б) четыре кнопки «новая\_игра», «финиш», «бросок1», «бросок2».

2. Создайте две черепашки с именами **куб1**, **куб2**. Для этих черепашек создайте 6 новых форм с точками (от одной до шести). Номера форм – от 40 до 45 (можно любые другие, главное – чтобы они шли по порядку).

3. На листе процедур напишите процедуру **новая\_игра**, в которой:

а) определяются переменные **сум1**, **сум2** со значением 0;

б) чистятся все текстовые окна;

с) запускается процесс анализа значений переменных **сум1**, **сум2**: если значение переменной **сум1** или **сум2** становится больше 50-ти, то игрок 1 или 2 (соответственно) проигрывает.

4. На листе процедур напишите процедуру **бросок1**, в которой:

а) определяется переменная **с** со значением 0. В эту переменную будет помещаться сумма очков, которые будут выпадать на двух кубиках;

б) для каждого кубика выполняется следующее:

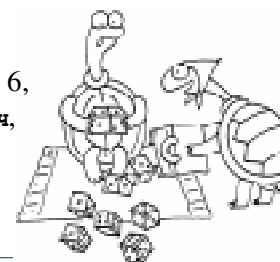
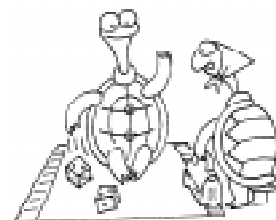
– определяется переменная **ч** со случайным значением от 1 до 6,

– кубик меняет форму на номер 39 плюс значение переменной **ч**,

– к переменной **с** прибавляется значение переменной **ч**,

– к переменной **сум1** прибавляется значение переменной **с**,

– значение переменной **сум1** выводится в текстовое окно **бр1**,

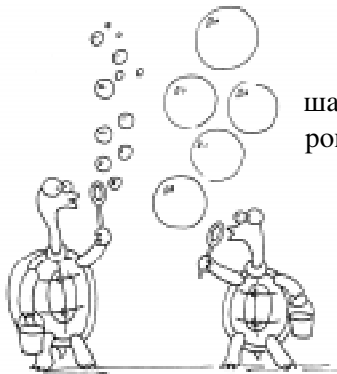




– в текстовое окно «сумма1» выводится значение переменной `сум1`.

5. На листе процедур напишите процедуру `бросок2`, в которой выполняется то же, что в процедуре `бросок1`, но только для других переменных (вместо `с` – `сс`, вместо `сум1` – `сум2`) и для других текстовых окон (вместо `бр1` – `бр2`, вместо `сумма1` – `сумма2`).

6. На листе процедур напишите процедуру `финиш`, в которой будут сравниваться переменные `сум1` и `сум2` и выдаваться сообщение о победителе.



### Игра 7. Кто надует больший шар

*Содержание игры:* два игрока по очереди дуют каждый в свой шар. Выигрывает тот, кто надует больший шар. Тот игрок, у которого шар лопается (размер оказывается более 100), проигрывает.

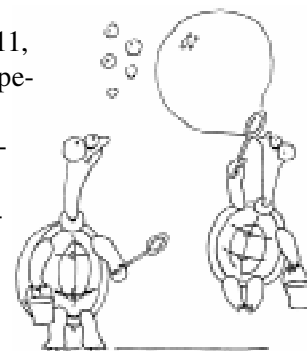
Для реализации этой игры сделайте следующее:

1. На листе проекта создайте
  - а) две черепашки **Красный** и **Синий** в форме воздушных шаров разного цвета;
  - б) два текстовых окна с именами «красный\_шар» и «синий\_шар»;
  - в) три кнопки «начало», «пуф\_красный», «пуф\_синий».
2. На листе процедур напишите процедуру `Начало`, в которой выполняется следующее:
  - а) чистятся текстовые окна;
  - б) определяются переменные `р_с_ш` и `р_к_ш` с начальными значениями 60 (начальный размер шаров);
  - в) черепашки-шары приобретают начальные формы и размеры.

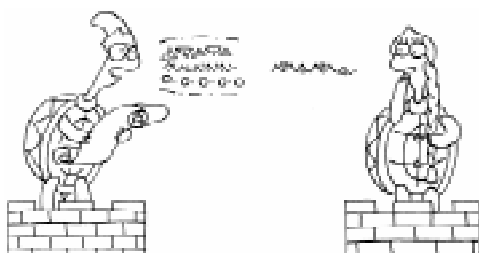
3. На листе процедур напишите процедуру `пуф_синий`, в которой выполняется следующее:

- определяется переменная `х` со случайным значением от 3 до 11,
- к значению переменной `р_с_ш` прибавляется значение переменной `х`,
- в текстовое окно «синий\_шар» выводится значение переменной `р_с_ш`,
- анализируется значение переменной `р_с_ш`; если оно больше 100, то синий шар лопается (меняет форму).

4. На листе процедур напишите процедуру `пуф_красный`, в которой выполняется то же, что и в процедуре – `пуф_синий`, но только для красного шара.



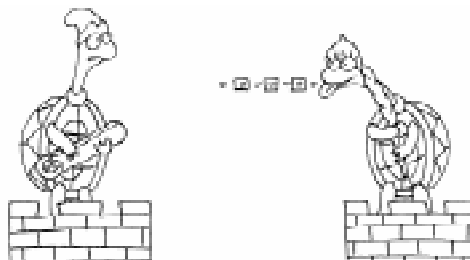
### Игра 8. Тарабарская грамота



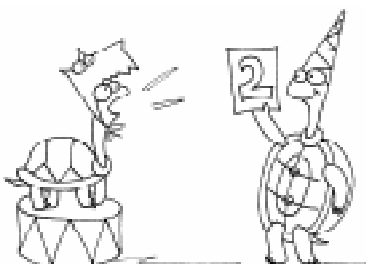
В XIV–XV веке на Руси для составления секретных бумаг в переписке царского двора с русскими послами использовалась шифровка, называемая «тарабарской грамотой».

Способ такой: *гласные* буквы и буквы *й, ь, ъ* – писались без изменений. А согласные заменялись согласно следующей схеме:

б	в	г	д	ж	з	к	л	м	н
щ	ш	ч	ц	х	ф	т	с	р	п



При шифровке согласные взаимно заменялись. Напишите процедуру, которая любой тест шифрует по этой методике.

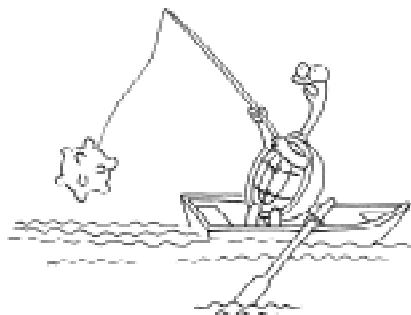


### Игра 9. Отгадай число

*Содержание игры:* компьютер загадывает число и просит пользователя отгадать его за три попытки; в зависимости от ответа пользователя, выдаются сообщения: «Задуманное число больше», «Задуманное число меньше», «Вы угадали», «Вы исчерпали свои попытки».

### Игра 10. Найди мину

*Содержание игры:* игрок должен обезвредить мину, спрятанную в реке (отгадать X-координату мины); программа задумывает число от  $-200$  до  $200$ , запрашивает у пользователя X-координату мины, перемещает ракету на координату, введенную пользователем, и сообщает: «Мина ближе», «Мина дальше», «Вы попали», в зависимости от ситуации. В случае попадания мина взрывается.

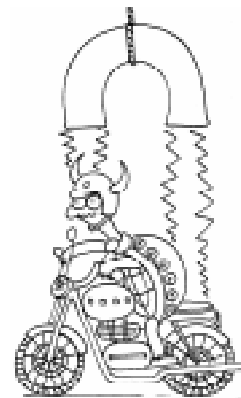


### Игра 11. Магнит

*Содержание игры:* игрок должен собрать магнитом все предметы, сделанные из железа.

Для реализации этой игры сделайте следующее:

1. На листе проекта создайте:
  - а) шесть черепашек **ч1**, ..., **ч6**, для некоторых из них выберите формы предметов, сделанных из железа, в диалоговых окнах этих черепашек в поле «инструкция» напишите «железо»,
  - б) кнопку «действие\_магнита».
2. На листе процедур напишите процедуру **действие\_магнита**, в которой выполняется следующее:
  - а) черепашки случайным образом распределяются на листе проекта;
  - б) для магнита пишутся команды, которые будут всегда перемещать его за курсором мыши;
  - в) в цикле повторяется следующее:
    - анализируются два условия: 1 – коснулся ли магнит черепашки, 2 – правило черепашки – **запусти [магнит]**; если оба условия *истинны*, то черепашка начинает перемещаться вслед за магнитом.
3. Напишите процедуру **магнит**, в которой будет выдаваться сообщение «Железо притягивается магнитом».



Наши авторы, 2004.  
Our authors, 2004.

Волкова Ризиды Анверовны,  
руководитель группы  
информационных технологий  
общеобразовательного учебного  
заведения Тольяттинской  
академии управления.