



ПРОГРАММИРОВАНИЕ В СРЕДЕ ЛОГО МИРЫ

Часть 3. Логические операции.

Логические датчики – И, ИЛИ, НЕ

Команды Лого, которые позволяют анализировать различные условия, выглядят так:

если условие [список инструкций]

если_иначе условие [список инструкций1] [список инструкций2]

Условие – это логическое выражение, которое может принимать два значения: ДА (истина, true, 1) и НЕТ (ложь, false, 0).

Логические выражения строятся с помощью логических операций – НЕ, И, ИЛИ.

Пример 1. Проанализировать, является ли введенное пользователем сообщение числом. Если нет, то сообщить «Вводите только числа»

если не число? ответ [сообщи [Вводите только числа]]

Пример 2. Проанализировать, является ли введенное пользователем число больше 7 и меньше 17. Если ДА, то сообщить «Вы школьник»

если и ответ > 7 ответ < 17 [сообщи [Вы школьник]]

Пример 3. Проанализировать, является ли введенное пользователем число меньше 7 или больше 17. Если число меньше 7 или больше 17, то сообщить «Вы не школьник».

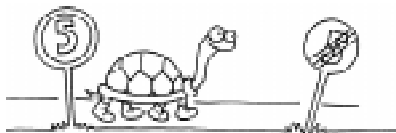
если или ответ > 7 ответ < 17 [сообщи [Вы не школьник]].

Схематическое изображение логических операций изображено на рисунке 1.



ЛОГИЧЕСКИЕ ОПЕРАЦИИ № 1

Задача 1-1. Логическая операция НЕ



Продемонстрируйте работу логической операции НЕ (рисунок 2).

Процедура считывает значение операнда из текстового окна «1-операнд» и выводит результат в текстовое окно «результат».

Решение:

это Задача1

пусть "X 1-операнд

пусть "рез не :X результат, ст вставь :рез

конец

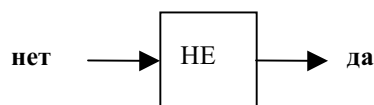


Рисунок 2.

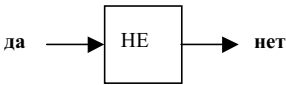
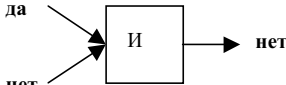

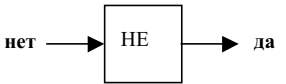
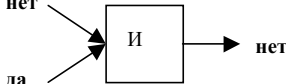

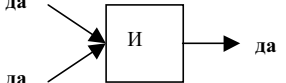

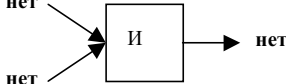
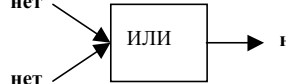
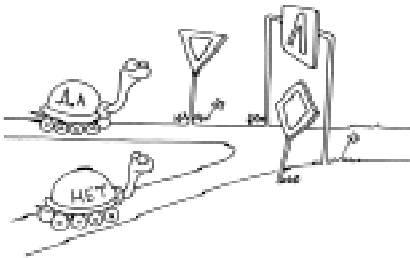
Операция НЕ – Логическое отрицание	Операция И – Логическое умножение	Операция ИЛИ – Логическое сложение
		
		
		
		

Рисунок 1.

Задача 1-2. Логическая операция И



Продемонстрируйте работу логической операции И (рисунок 3).

Процедура считывает значение 1-го операнда из текстового окна «1-операнд», 2-го операнда из текстового окна «2-операнд» и выводит результат в текстовое окно «результат».

Решение:

это Задача2
 пусть "X 1-операнд
 пусть "Y 2-операнд
 пусть "рез и :X :Y результат, ст вставь :рез
 конец

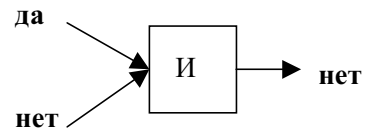
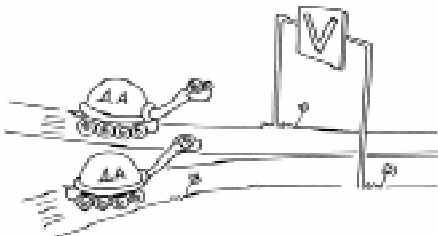


Рисунок 3.

Задача 1-3. Логическая операция ИЛИ



Продемонстрируйте работу логической операции ИЛИ (рисунок 4).

Процедура считывает значение 1-го операнда из текстового окна «1-операнд», 2-го операнда из текстового окна «2-операнд» и выводит результат в текстовое окно «результат».



Рисунок 4.

Задача 1-4. Логические операции И, ИЛИ

Продемонстрируйте работу логических операций И и ИЛИ (рисунок 5).

Процедура считывает значение 1-го операнда из текстового окна «1-операнд», 2-го операнда из текстового окна «2-операнд», название операции из текстового окна «операция» и выводит результат в текстовое окно «результат».

Решение:

```
это Задача4
если операция = "И [Задача2]
если операция = "Или [Задача3]
конец
```

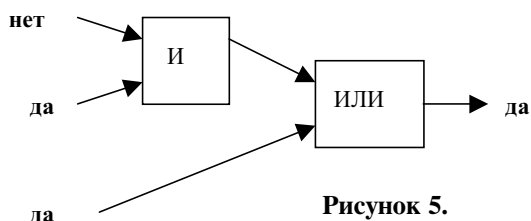


Рисунок 5.

ЛОГИЧЕСКИЕ ОПЕРАЦИИ № 2

Задача 2-1. Логическая схема 1-оп И 2-оп ИЛИ 3-оп

Продемонстрируйте работу логической схемы (рисунок 6). Процедура считывает значение операндов из текстовых окон «1-оп», «2-оп», «3-оп» и выводит результат в текстовое окно «результат».

Решение:

```
это Задача1
пусть "оп1 1-оп
пусть "оп2 2-оп
пусть "оп3 3-оп
пусть "рез или (и :оп1 :оп1) :оп3 результат,
ст вставь :рез
конец
```

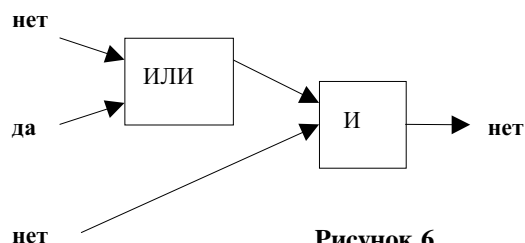


Рисунок 6.

Задача 2-2. Логическая схема (1-оп ИЛИ 2-оп) И 3-оп

Продемонстрируйте работу логической схемы (рисунок 7). Процедура считывает значение операндов из текстовых окон «1-оп», «2-оп», «3-оп» и выводит результат в текстовое окно «результат».

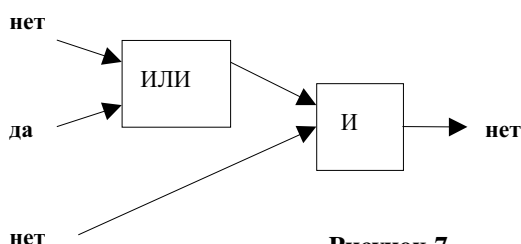


Рисунок 7.

Задача 2-3. Логическая схема (1-оп И 2-оп) ИЛИ (3-оп ИЛИ 4-оп)

Продемонстрируйте работу логической схемы (рисунок 8). Процедура считывает значение операндов из текстовых окон «1-оп», «2-оп», «3-оп», «4-оп» и выводит результат в текстовое окно «результат».

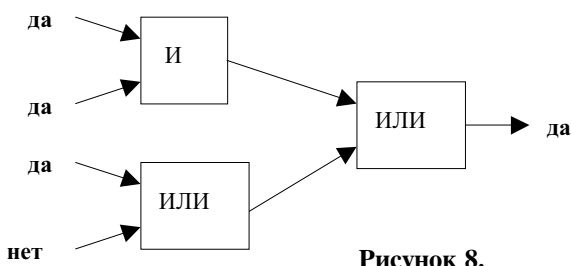


Рисунок 8.

Задача 2-4. Логическая схема НЕ (1-оп И 2-оп ИЛИ 3-оп)

Продемонстрируйте работу логической схемы (рисунок 9). Процедура считывает значение операндов из текстовых окон «1-оп», «2-оп», «3-оп» и выводит результат в текстовое окно «результат».

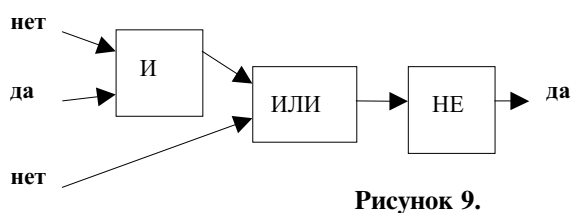


Рисунок 9.

РАЗВЕТВЛЯЮЩИЙСЯ АЛГОРИТМ

РАЗВЕТВЛЯЮЩИЙСЯ АЛГОРИТМ № 1

Задача 1-1. Анализ чисел на четность

1) Программа запрашивает у пользователя число и осуществляет контроль на число, если введено не число, сообщает «Вводите только числа» и завершает свою работу.

2) Анализирует введенное число на четность. Если число четное сообщает «Вы ввели четное число». Если число нечетное, сообщает «Вы ввели нечетное число».

Подсказка: используйте команды Лого – не число?, остаток.



Решение:

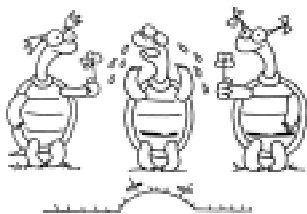
это задача1

спроси [введите число -] пусть "ч ответ

если не число? :ч [сообщи [вводите только числа] автостоп]

если_иначе (остаток :ч 2) = 0 [сообщи [число четное]] [сообщи [число нечетное]]

конец



Задача 1-2. Анализ чисел на делимость на 3

1) Программа запрашивает у пользователя число и осуществляет контроль на число, если введено не число, сообщает «Вводите только числа» и завершает свою работу.

2) Анализирует введенное число на делимость на 3. Если число делится на 3, то сообщает «Ваше число делится на 3». Если число не делится на 3, сообщает «Ваше число не делится на 3». Используйте команды Лого – не число?, остаток.

Задача 1-3. Анализ дроби на правильность

1) Программа запрашивает у пользователя числитель дроби и помещает его в текстовое окно «числитель», затем запрашивает у пользователя знаменатель дроби и помещает его в текстовое окно «знаменатель».

2) Анализирует введенную дробь и в текстовом окне дробь пишет «правильная», если дробь правильная, и пишет «неправильная», если дробь неправильная.

Решение:

это задача3

спроси [введите числитель -]

пусть "ч ответ числитель, ст вставь :ч

спроси [введите знаменатель -]

пусть "з ответ

знаменатель, ст вставь :з

если_иначе :з > :ч [дробь, ст вставь "правильная]

[дробь, ст вставь "неправильная]

конец

Задача 1-4. Анализ величины угла

Программа запрашивает у пользователя значение угла (от 0 до 180). Анализирует введенное число и

если введенное число < 90, сообщает «Угол острый»,

если введенное число = 90, сообщает «Угол прямой»,

если введенное число < 180 и > 90 , сообщает «Угол тупой»,
 если введенное число $= 180$, сообщает «Угол развернутый».
 Черепашка рисует угол.

РАЗВЕТВЛЯЮЩИЙСЯ АЛГОРИТМ № 2

Задача 2-1. Тест на знание курсов черепашки

Напишите процедуру, которая проверяет знание курсов черепашки.

В диалоговое окно выдается задание «Измените курс так, чтобы черепашка смотрела на дом». Затем черепашка меняет курс на значение, введенное пользователем. Анализирует курс черепашки. Если он верен (от 75 до 105), то выдается сообщение «Верно», в противном случае выдается сообщение «Ошибка!».

Решение:

```

это Задача1
спроси [Поверните черепашку к домику. Введи курс]
Чер1, нк ответ
Чер1, пусть "к курс
если_иначе и :к > 74 :к < 106 [сообщи "Верно] [сообщи "Ошибка!]
конец
    
```



Задача 2-2. Рыбка в море (анализ y-координаты)

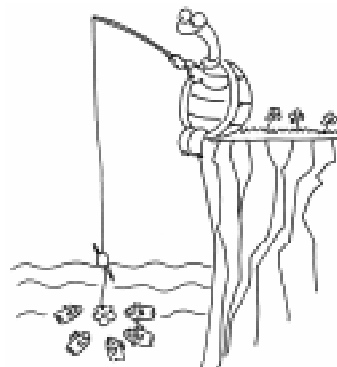
Проверьте местоположение рыбки (анализ y-координаты черепашки). Если она больше -200 и меньше -100 , вывести сообщение «Рыбка в море», в противном случае – вывести сообщение «Рыбку поймали»

Подсказка: используйте команду Лого – `y_коор`

Решение:

```

это Задача2
рыбка, пусть "y y_коор
если_иначе и :y > -200 :y < -100 [сообщи [Рыбка в море]]
[сообщи [Рыбку поймали]]
конец
    
```



Задача 2-3. Рыбки в озере (анализ цвета поля)

Рыбки плавают в озере, не выходя за его границы.

Подсказка: используйте команды Лого – `цп` (цвет поля),

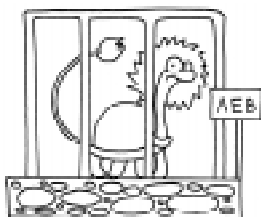
`перебор [к[ч1 ч2 ч3]] [для :к]`

Решение:

```

это Задача3
всегда [
перебор [к [рыбка1 рыбка2 рыбка3]] [для :к нк сл 360 вп 5 если цп = 55 [нд 5] жди 1 ]
]
конец
    
```

Задача 2-4. Лев в клетке (анализ координат x и y)



Проверьте местоположение льва (анализ координат x и y черепашки). Если x больше 0 и меньше 100 , а y больше -150 и меньше -50 , вывести сообщение «Лев в клетке», в противном случае – вывести сообщение «Лев сбежал»

Подсказка: используйте команды Лого – `x_коор`, `y_коор`

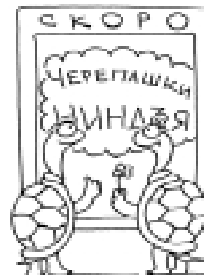
РАЗВЕТВЛЯЮЩИЙСЯ АЛГОРИТМ № 3

Задача 3-1. Поход в кино

Процедура запрашивает у пользователя сумму денег, которая у него при себе, и наличие свободного времени в ближайшие 3 часа. Если есть свободное время и сумма денег составляет больше 100 рублей, процедура выдает сообщение «Пойдем в кино». В противном случае выдается сообщение «УВЫ!»

Решение:

```
это Задача1
спроси [Введите сумму денег]
пусть "д ответ
спроси [вы свободны ближайшие 3 часа (1 - да, 0-нет)]
пусть "с ответ
если иначе и :д > 100 :с = 1 [сообщи [Пойдем в кино]] [сообщи [УВЫ!]]
конец
```



Задача 3-2. Призывники

Напишите процедуру, которая анализирует пригодность человека к службе в армии. Процедура анализирует возраст и пол человека. Если пол мужской, а возраст от 18 до 27, процедура выдает сообщение «Вы – призывник». В противном случае – сообщение «Вы не годитесь для армии».

Задача 3-3. Проверка – двузначное число

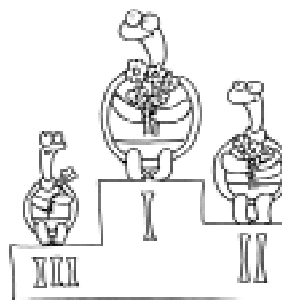
Процедура запрашивает у пользователя «Введите двузначное число». Если пользователь ошибся, то выдать сообщение «Вводите только двузначные числа»

Задача 3-4. Поиск максимального из 3-х чисел

Напишите процедуру, которая ищет максимальное число из 3-х чисел, введенных пользователем. Сами числа выводятся в текстовое окно «Все_числа», а максимальное число выводится в текстовое окно «Самое_большое».

Решение:

```
это Задача4
спроси [Введите 1-ое число]
пусть "ч1 ответ
Все_числа, ст пиши :ч1
спроси [Введите 2-ое число]
пусть "ч2 ответ
Все_числа, пиши :ч2
спроси [Введите 3-ое число]
пусть "ч3 ответ
Все_числа, пиши :ч3
пусть "макс :ч1
если (и :ч2 > :ч1 :ч2 > :ч3) [пусть "макс :ч2]
если (и :ч3 > :ч2 :ч3 > :ч1) [пусть "макс :ч3]
Самое_большое, ст вставь :макс
конец
```



РАЗВЕТВЛЯЮЩИЙСЯ АЛГОРИТМ № 4

Задача 4-1. Загадка с вариантами ответов

Процедура задает загадку пользователю, например, «Зимой и летом одним цветом». У загадки несколько верных ответов – ель, елка, ёлка, сосна. Во всех этих случаях выдается сообщение – «Отгадал!». В противном случае – сообщение – «Подумай еще».



Решение:

это загадка
спроси [Зимой и летом одним цветом]
пусть "от ответ
если_иначе (или :от = "ель :от = "елка :от = "ёлка :от = "сосна) [сообщи [Отгадал!]]
[сообщи [Подумай еще!]]
конец

Задача 4-2. Мальвина идет в кино

Мальвина собралась пойти в кино с Буратино и Пьеро. Но она готова пойти и с одним кавалером. Напишите процедуру, в которой Мальвина спрашивает у Буратино и Пьеро, готовы ли они пойти с ней в кино. В случае, если хотя бы один из них готов, Мальвина сообщает «Ура! Идем!». В противном случае она говорит «Очень жаль!»



Задача 4-3. «Да» и «нет» не говорить

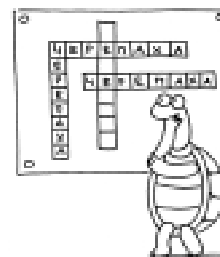
Запрограммируйте известную детскую игру «Да и нет не говорить, черное с белым не носить».

Задача 4-4. Анализ количества букв в текстовых окнах

Напишите 2 процедуры.

1-ая: Задание. Процедура выдает задание «Введите 2-х, 3-х и 4-х буквенные слова в соответствующие текстовые окна».

2-ая: Проверка. Анализирует количество букв в каждом текстовом окне. Если пользователь ввел правильные слова, то выдается сообщение «Верно!», в противном случае выдается сообщение «Ошибка!».



Решение:

это Задание
скажи [Две_буквы Три_буквы Четыре_буквы] [ст]
сообщи [Введите 2-х, 3-х и 4-х буквенные слова в соответствующие текстовые окна]
конец
это Проверка1
Две_буквы, внт выделяй вкт копируй пусть "б2 сколько буфер
Три_буквы, внт выделяй вкт копируй пусть "б3 сколько буфер
Четыре_буквы, внт выделяй вкт копируй пусть "б4 сколько буфер
если_иначе (и :б2 = 2 :б3 = 3 :б4 = 4) [сообщи [Верно!]] [сообщи [Ошибка!]]
конец
это Проверка2 ; 2-ой способ решения.
пусть "б2 сколько Две_буквы
пусть "б3 сколько Три_буквы
пусть "б4 сколько Четыре_буквы
если_иначе (и :б2 = 2 :б3 = 3 :б4 = 4) [сообщи [Верно!]] [сообщи [Ошибка!]]
конец

РАЗВЕТВЛЯЮЩИЙСЯ АЛГОРИТМ № 5

Задача 5-1. Управляем движением черепашки с клавиатуры

Напишите процедуру, которая управляет движением черепашки с цифровой клавиатуры.

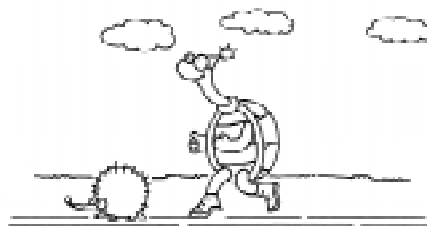
Подсказка: Пользуйтесь командами Лого: жди_символ

Решение:

это стрелки
ч1,
всегда [

```

пусть "х жди символ
если :х = 4 [нк 270]
если :х = 8 [нк 0]
если :х = 6 [нк 90]
если :х = 2 [нк 180]
если :х = 5 [отмени [вп 5 жди 1]]
если :х = 0 [всегда [вп 5 жди 1]]
]
конец
    
```



Задача 5-2. Управляем движением черепашки в диалоге

Напишите процедуру, в которой черепашка в форме бегуна бежит по листу проекта слева направо, при попадании на красный цвет появляется вопрос «Бежать дальше?». Если пользователь напишет «да», бегун отвечает «Хорошо!» и продолжает бег, в противном случае бегун останавливается. На листочке нарисуйте блок-схему.

Решение:

```

это крас          ; имя процедуры пишется в диалоговом окне красного цвета
спроси [Бежать дальше?]
если_иначе ответ = "да [сообщи "Хорошо! ] [останов]
конец

это бег
бегун, нк 90 всегда [нф 42 вп 3 жди 1 нф 43 вп 3 жди 1 нф 44 вп 3 жди 1]
конец
    
```

Задача 5-3. Столкновение черепашек

Две черепашки бесцельно перемещаются по листу проекта (либо в определенной части листа проекта). При столкновении появляется сообщение «Ой!», и черепашки перестают перемещаться.

Подсказка: Пользуйтесь командами Лого: коснулись? "ч1 "ч2, нов_х, нов_у, сл Х.

Решение:

```

это общение
всегда [ он, нов_х -150 + сл 300 нов_у -200 + сл 200 жди 2
она, нов_х -150 + сл 300 нов_у -200 + сл 200 жди 2
если (коснулись? "он "она ) = "да [ сообщи "Ой! автостоп]
]
конец
    
```



Задача 5-4. Гром гремит

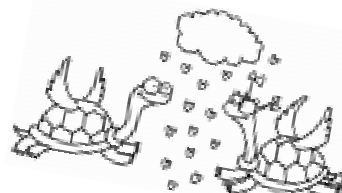
Два облака (черепашки) перемещаются по голубому небу. При столкновении сверкает молния (появляется и исчезает черепашка) и гремит гром (выдается сообщение или звук).

Подсказка: облака не должны улетать за пределы неба (для этого запрограммируйте цвета).

Решение:

```

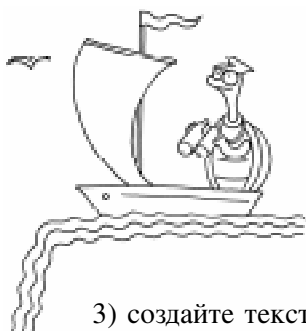
это гром
молния, пп сч
всегда [ облако1, нов_курс сл 360 вп сл 10 жди 1
облако2, нов_курс сл 360 вп сл 10 жди 1
если (коснулись? "облако1 "облако2 ) = "да [ пусть "mmm место молния,
nm :mmm пч жди 10 сч сообщи "бах! автостоп]
]
конец
    
```



ДОПОЛНИТЕЛЬНЫЕ ЗАДАЧИ НА РАЗВЕТВЛЯЮЩИЙСЯ АЛГОРИТМ

Задача 1. Движение корабля по течению и против

Создать проект «Движение корабля». Корабль перемещается на расстояние в направлении, которое зависит от условия задачи (скорость корабля, скорость реки, направление движения и время задаются бегунками).



Для этого

- 1) на листе проекта нарисуйте реку, стрелкой укажите направление течения реки;
- 2) создайте 4 бегунка:
 - a) бегунок с именем **Ск_кораб**, с диапазоном от 0 до 40,
 - b) бегунок с именем **Ск_реки**, с диапазоном от 0 до 10,
 - c) бегунок с именем **Время**, с диапазоном от 0 до 100,
 - d) бегунок с именем **Течение**, с диапазоном от 0 до 1
 (0 – корабль идет против течения, 1 – корабль идет по течению);
- 3) создайте текстовое окно «Расстояние» (для вывода результата расчета);
- 4) на листе процедур напишите процедуру «Решение», в которой выполняется следующее:
 - a) корабль помещается к месту старта (любое место на ваше усмотрение),
 - b) анализируется значение бегунка **Течение**. Если оно равно 1, то относительная скорость (переменная **Ск_отн**) рассчитывается как сумма скорости корабля и скорости реки. Если оно равно 0, то относительная скорость рассчитывается как разность скорости корабля и скорости реки.
 - c) рассчитывается расстояние (переменную назовите **расст**), которое пройдет корабль за время, указанное в бегунке **Время**. (Надеюсь, вы знаете, как рассчитать расстояние, зная время и относительную скорость).
 - d) в текстовое окно «Расстояние» выводится полученное значение,
 - 5) корабль плавно перемещается в нужном направлении на рассчитанное расстояние с относительной скоростью.

Решение:

это расчет

```
если_иначе течение = 1 [ пусть "Ск_отн Ск_корабля + Ск_реки кораблик, нк 90]
    [ пусть "Ск_отн Ск_корабля - Ск_реки кораблик, нк 270 ]
    пусть "Расст :Ск_отн * время
    Расстояние, ст вставь :Расст
    кораблик, нм [-23 55] плавно :Расст :Ск_отн
конец
```

Задача 2. Расселяем животных (дикие – домашние) (множества)

Игра – на листе находятся черепашки в форме различных животных (домашних и диких). Задача игрока – разместить отдельно диких и домашних животных (поместить их на разные цветные пятна), затем нажать на кнопку «Готово», чтобы проверить правильность выполнения задания.

Для этого:

- 1) на листе проекта нарисовать два овала разного цвета и подписать их «дикие», «домашние»;
- 2) создать не менее 6 черепашек в форме различных животных и назвать дик1, дик2, ..., дом1, дом2, ...;
- 3) написать процедуру «начнем», которая выставляет всех животных вне этих овалов;
- 4) написать процедуру «готово», которая анализирует правильность выполнения задания пользователем. Неправильно размещенные животные выталкиваются из цветных

овалов. Количество ошибок подсчитывается, и, в зависимости от результата, выдаются сообщения «Молодец!» или «Потрудись_еще!».

Решение:

```
это начнем  
дом1, нм [0 -50]  
дом2, нм [0 40]  
дом3, нм [0 -100]  
дик1, нм [1 100]  
дик2, нм [0 0]  
дик3, нм [0 -200]  
конец
```



```
это готово  
пусть "ош 0
```

```
перебор [к [дом1 дом2 дом3]]
```

```
[для :к если цп = 113 [нов_х сл 20 нов_у сл 200 пусть "ош :ош + 1 ] ]
```

```
перебор [к [дик1 дик2 дик3 ]]
```

```
[для :к если цп = 63 [нов_х сл 20 нов_у сл 200 пусть "ош :ош + 1]]
```

```
если иначе :ош = 0 [Сообщи "Молодец! ] [Сообщи "Потрудись_еще! ]
```

```
конец
```

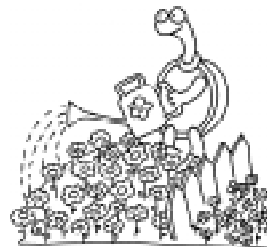
Задача 3. Рассчитываем налоги

Рассчитайте сумму налогов на доходы. Налоги рассчитываются по следующей схеме:

Если доход меньше 5000 руб., то налог составляет 12%, если доход больше 5000 руб., то налог составляет 30%.

Для этого:

- 1) на листе проекта создайте текстовое окно для вывода результата;
- 2) напишите процедуру, выполняющую следующие действия:
 - а) спрашивает сумму доходов за год в рублях,
 - б) выполняет контроль на число,
 - в) сохраняет введенное значение в переменной доход,
 - г) вычисляет налог:
если доход < 5000, то налог – 12 % от дохода;
если доход > 5000, то налог – 30 % от дохода;
 - д) выводит результат в текстовое окно.



Решение:

```
это налоги
```

```
спроси [Введите сумму Вашего дохода в рублях]
```

```
пусть "доход ответ
```

```
если не число? ответ [сообщи[Сумма Вашего годового дохода вводится числом] останов]
```

```
если :доход < 5000 [ пусть "н :доход * 12 / 100 ]
```

```
если :доход > 5000 [ пусть "н :доход * 30 / 100 ]
```

```
ттт, ст вставь [Ваш доход за год составил -] вставь символ 32 вставь :доход вставь символ 32 пиши [руб.]
```

```
вставь [Вы должны заплатить налог в размере -] вставь символ 32 вставь :н вставь символ 32 пиши [руб.]
```

```
конец
```

*Волкова Ризид Анверовна,
руководитель группы
информационных технологий
общеобразовательного учебного
заведения Тольяттинской
академии управления.*



Наши авторы, 2004.
Our authors, 2004.