

ЛАБОРАТОРНАЯ РАБОТА № 1

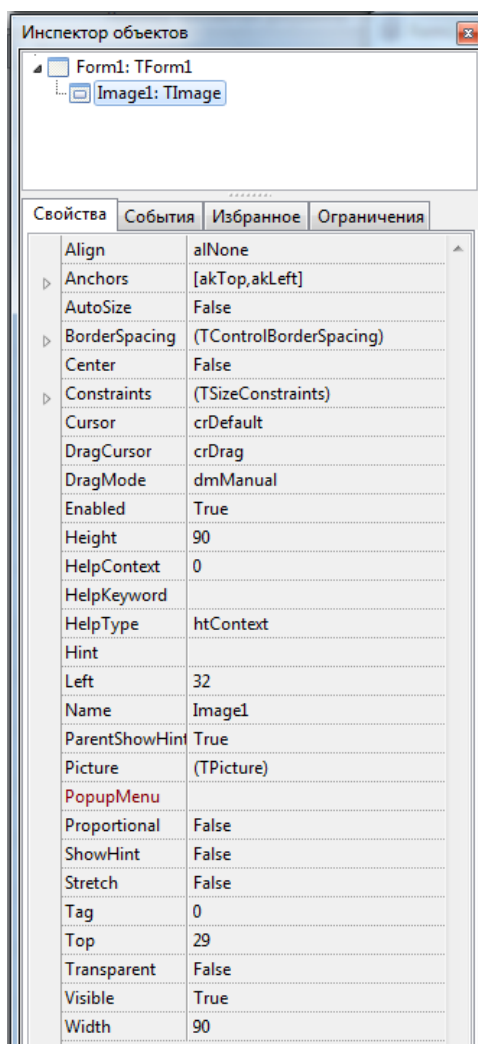
ЗНАКОМСТВО С ОБЪЕКТОМ IMAGE

Для вставки изображений на форму приложения используется объект TImage (на панели инструментов во вкладке Additional -> TImage).



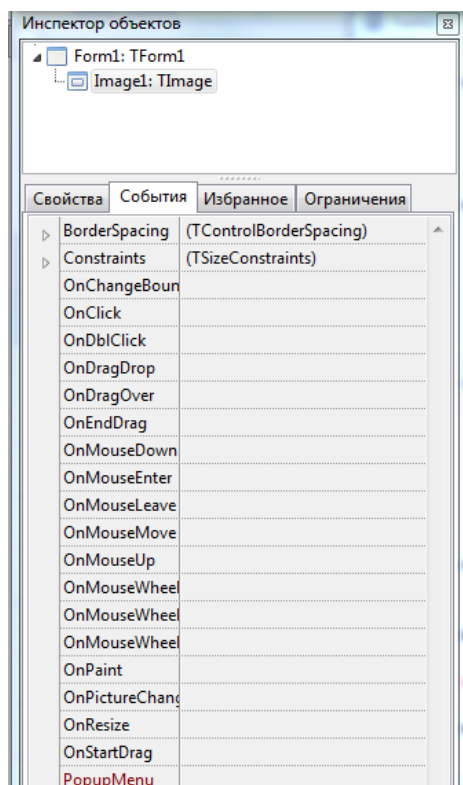
РИСУНОК 1 – ОБЪЕКТ IMAGE

Перечислим некоторые свойства данного объекта и их смысловую нагрузку.



- **Width** – ширина объекта (изображения) [в пикселях];
- **Height** – высота объекта (изображения) [в пикселях];
- **Caption** – заголовок на форме (не путать со свойством Name);
- **Name** – наименование объекта (используется в программном коде);
- **Top** – положение левого верхнего угла объекта относительно формы (отступ сверху) [в пикселях];
- **Left** – положение левого верхнего угла объекта относительно формы (отступ справа) [в пикселях];
- **Proportional** – устанавливает пропорциональность изображения [bool];
- **Stretch** – растягивание изображения до границ объекта [bool];
- **Picture** – файл с графическим изображением, которое подлежит вставке в объект TImage.

Рассмотрим некоторые события, доступные для обработки.

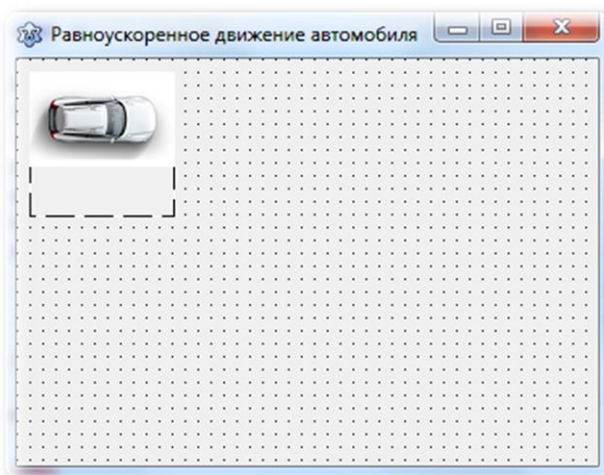


- **onClick** – при клике на объекте;
- **onDblClick** – при двойном клике на объекте;
- **onMouseMove** – при движении мыши по объекту;
- **onMouseEnter** – при входе курсора мыши в объект;
- **onMouseLeave** – при выходе курсора мыши из объекта;

ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Запустите IDE Lazarus. Создайте новый проект (Файл – Создать – Проект – Приложение)
2. Поменяйте заголовок формы приложения на «Равноускоренное движение автомобиля».
3. Поместите на форму объект TImage.
4. Установите ширину и высоту объекта TImage в значение 90 (пикселей).
5. Найдите (нарисуйте) изображение автомобиля видом сверху.
6. Вставьте изображение в свойство Picture объекта TImage.
7. Используя необходимые свойства добейтесь того, чтобы изображение автомобиля было вписано в объект TImage пропорционально.

Пример формы с правильно выполненным заданием:



ЛАБОРАТОРНАЯ РАБОТА №2

ЗНАКОМСТВО С ОБЪЕКТОМ TIMER

Для выполнения некоторых действий через определенный интервал времени используют объект TTimer (на панели инструментов во вкладке System -> TTimer).

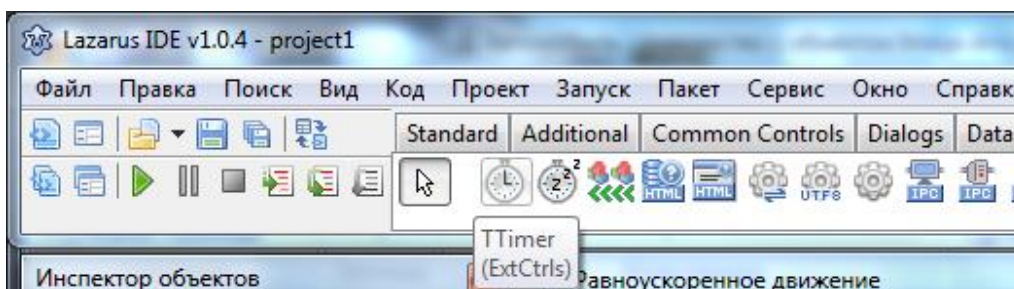


РИСУНОК 2 – ОБЪЕКТ TIMER

Объект TTimer – не отображаемый, то есть в процессе работы приложения этот компонент не виден пользователю. В окне редактора формы данный объект отображается в виде иконки.

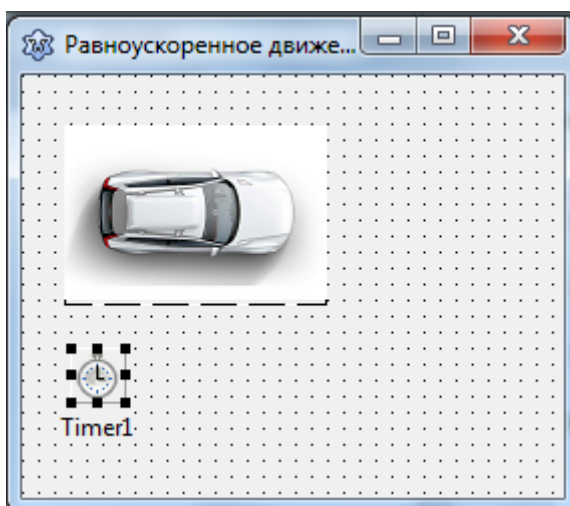
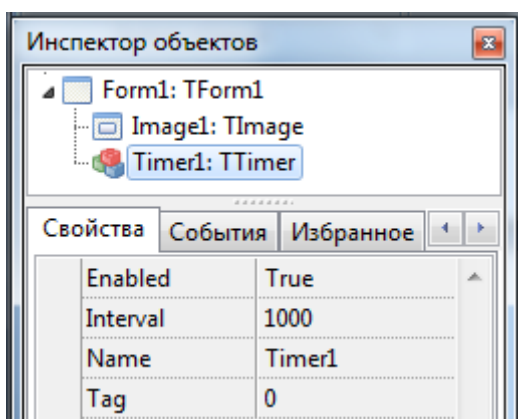


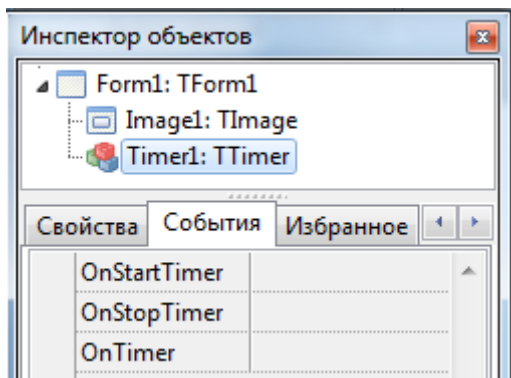
РИСУНОК 3 – ИКОНКА ОБЪЕКТА TIMER

Перечислим некоторые свойства данного объекта и их смысловую нагрузку.



- **Enabled** – определяет, включен ли таймер [bool];
- **Interval** – определяет интервал времени, через который выполняются заданные действия [в миллисекундах];
- **Name** – имя объекта, которое используется в редакторе кода [текст];

Рассмотрим некоторые события, доступные для обработки.



- **onStartTimer** – при запуске (включении) таймера;
- **onStopTimer** – при остановке (выключении) таймера;
- **onTimer** – при срабатывании таймера (с интервалом, установленным в свойстве Interval);

ПРАКТИЧЕСКАЯ ЧАСТЬ

Описание задачи. В работе необходимо смоделировать равноускоренное движение автомобиля. Один из способов реализации идеи – через использование таймера. Если взять достаточно маленький интервал пересчета координат изображения автомобиля, то человеку будет казаться, что автомобиль движется плавно. Поэтому, через равный и достаточно малый интервал времени будет пересчитываться координату расположения объекта с изображением.

Для решения задачи, нужно, как минимум, вспомнить закон равноускоренного движения из курса физики. В одномерном случае движение по оси OX будет выглядеть так:

$$x = x_0 + v_0 t + \frac{at^2}{2},$$

где (для нашей задачи): x – текущая координата, x_0 – координата в предыдущий момент времени, v_0 – скорость в предыдущий момент времени, a – ускорение, t – время.

Из данного уравнения движения нам будет дано только значение ускорения, а время будет является значением интервала таймера.

Рассмотрим, чему равна начальная скорость автомобиля:

$$v_0 = v_{\text{текущая начальная}} = v_{\text{предыдущая}} + at$$

Итак, мы имеет следующие значения:

Наим.	Комментарий	Значение
a	Ускорение	Дано по условию задачи
t	Время	Свойство Interval объекта Timer
x_0	Начальное положение изображения	Свойство Left объекта Image
v_0	Начальная скорость для данного интервала	$v_{\text{предыдущая}} + at$

Так как при каждом следующем срабатывании таймера автомобиль набирает всё большую скорость, необходимо «запоминать» ее величину. При этом следует помнить, что если указать переменную v_0 в секции var для обработчика события таймера, то это значение будет сохранено только пока выполняется код события (**локальная переменная**), а нам необходимо «помнить» значение скорости даже в те моменты, когда таймер не сработал (между интервалами). Для этого, переменную v_0 следует поместить в секцию var всего проекта, как показано на рисунке (такая **переменная будет называться глобальной**):

```
1  unit Unit1;
.
.  { $mode objfpc } { $H+ }
.
5  interface
.
.  uses
.  Classes, SysUtils, FileUtil, Forms, Controls, Graphics,
.
.  type
10 type
.
.  { TForm1 }
.
.  TForm1 = class(TForm)
15     Image1: TImage;
.     Timer1: TTimer;
.     private
.     { private declarations }
.     public
20     { public declarations }
.     end;
.
.  var
.     Form1: TForm1;
25     v0: real; //переменная для хранения скорости
.
.  implementation
```

РИСУНОК 4 - ПЕРЕМЕННАЯ V0

Порядок выполнения работы.

1. Разместите на произвольном участке формы объект TImage.
2. Установите интервал таймера в значение 10 миллисекунд.
3. Добавьте переменную для хранения скорости *v0* в секцию *var* приложения (Рисунок 4).
4. Напишите обработчик события при срабатывании таймера с подсчетом координат автомобиля:
 - a. Создайте локальную переменную *a* (в секции *var* обработчика события).
 - b. Установите значение *a:=1*.
 - c. Посчитайте координату автомобиля.
 - d. Посчитайте новое значение скорости и запишите его в переменную *v0*.
5. Ограничьте максимальную скорость (*v0 = 1000*) через конструкцию «Если *v0>1000*, то *v0* присвоить значение 1000»
6. Запустите проект и проверьте корректную работу приложения.

Возможные проблемы при выполнении работы.

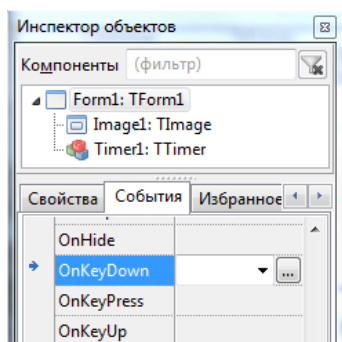
1. **Проблема.** При запуске автомобиль пропадает.
Причина. Изображение выходит за границы формы слишком быстро, т.к. интервал задан в миллисекундах, а при подсчетах координат вы не поделили на 1000, чтобы перевести в секунды. При первом же запуске интервала, координата будет слишком большой и заставит изображение с автомобилем выйти за границы формы.
Решение. При вычислении поделите каждый интервал на 1000 для перевода в секунды.
2. **Проблема.** Ошибка «Error: Incompatible type... Got "Real", expected "LongInt"»
Причина. Координата изображения – целое число (integer), а при вычислении координаты *x* может получиться дробное число (real). Невозможно целому числу присвоить дробное значение.
Решение. Округлите, используя функцию *round*. Пример применения функции *round*:

b:= round (100/3); //в переменную b запишется округленное значение 100/3

ЛАБОРАТОРНАЯ РАБОТА №3

ЗНАКОМСТВО С ОБРАБОТЧИКАМИ НАЖАТИЯ КЛАВИШ

В данной работе мы сделаем так, чтобы автомобиль перемещался вверх/вниз при нажатии на стрелочки на клавиатуре. Для реализации управления движением автомобиля, рассмотрим некоторые обработчики формы, которые отвечают за нажатие клавиш.



- **OnKeyDown** – при нажатии клавиши (в момент нажатия);
- **OnKeyUp** – в момент отжатия клавиши;

ПРАКТИЧЕСКАЯ ЧАСТЬ

Описание задачи. В работе необходимо обеспечить управление автомобилем (передвижение вверх и вниз при равноускоренном движении слева направо). Пользователь регулирует движение клавишами «вверх» и «вниз». Так как в компьютере сигнал с клавиатуры приходит в виде кода символа, то нам необходимо уметь определять какой именно символ пришел и в зависимости от этого либо двигать автомобиль вверх, либо вниз.

Порядок выполнения работы.

1. Сделайте так, чтобы после того, как автомобиль вышел за границу формы справа, он вернулся на исходную позицию. Для этого воспользуйтесь подсказкой для добавления в код таймера: **«Если Позиция Машины Слева больше Ширины Формы, тогда Позиции Машины Слева присвоить значение 0».**
2. Запустите проект и убедитесь, что машина перемещается в начало. Обратите внимание, что она не выезжает из формы постепенно, а сразу появляется целиком.
*Подумайте и сделайте так, чтобы она постепенно появлялась слева, а не мгновенно.
3. Добавьте глобальную переменную в проект под названием «control» с типом данных целое число.
4. Добавьте обработчик события нажатия на клавишу для формы. Перейдите к его коду и впишите туда код на языке Паскаль: « control := Key; »
5. Добавьте обработчик события отжатия клавиши для формы. Перейдите к его коду и впишите туда код на языке Паскаль: « control := 0; »
6. Вставьте в нужное место обработчика события для Таймера код на языке Паскаль, соответствующий следующему алгоритму:
**«Если control = 40, тогда Позиции Машины Сверху присвоить значение Позиции Машины Сверху + 2
Иначе Если control = 38, тогда Позиции Машины Сверху присвоить значение Позиции Машины Сверху – 2»**
7. Запустите проект и проверьте корректную работу приложения.
8. Внимательно изучите код программы и найдите ответы на вопросы (письменно):
 - 1) Что значит строчка control := Key; ? В чем ее смысл и предназначение? На что влияет наличие / отсутствие этой строчки в программе?
 - 2) Что значит строчка control := 0; ? В чем ее смысл и предназначение? На что влияет наличие / отсутствие этой строчки в программе?
 - 3) Почему мы сравниваем control с цифрами 40 и 38 ? Что значат эти цифры?

ЛАБОРАТОРНАЯ РАБОТА № 4

ОПЕРАТОР ВЫБОРА CASE OF

Данная работа посвящена актуализации знаний об использовании оператора **case of**. Рассмотрим два варианта программного кода, имеющих равносильное значение.

if	case of
<pre>if (a = 1) or (a = 3) or (a=5) or (a=7) or (a=9) then writeln('нечетное') else if (a = 2) or (a=4) or (a=6) or (a=8) then writeln('четное') else writeln('Число вне диапазона от 1 до 9');</pre>	<pre>case a of 1,3,5,7,9: writeln('нечетное'); 2,4,6,8: writeln('четное') else writeln('Число вне диапазона от 1 до 9'); end;</pre>

Внимательно изучите оба примера, проведите аналогию.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Описание задачи. В работе необходимо «научить» автомобиль тормозить и начинать движение только при нажатии стрелочек на клавиатуре. Таким образом, при старте программы автомобиль будет обездвижен, а движение начнется только при нажатии клавиши «вправо». Иными словами, мы должны уметь обрабатывать 4 клавиши (перемещение вверх/вниз и газ/тормоз).

Что такое *торможение* на бытовом уровне? Это отрицательное ускорение!

Таким образом, поведение водителя может быть следующим:

- Нажата педаль акселератора (газ) – значит, ускорение будет положительным (в этом случае в нашей задаче мы возьмем его равным 1)
- Нажата педаль тормоза – значит, ускорение будет отрицательным (в этом случае в нашей задаче мы возьмем его равным -1)
- Никакая клавиша не нажата – значит, движение автомобиля будет равномерное прямолинейное, либо скорость равна нулю (в этом случае ускорение будет равно 0)

Порядок выполнения работы.

1. В коде обработчика события для таймера измените оператор if, который отвечает за выбор направления движения автомобиля (вверх / вниз) на оператор case of.
2. Проверьте, что после замены приложение по-прежнему работает корректно.
3. Уберите строку, в которой ускорению присваивается значение “1”;
4. Убедитесь, что оператор case of в обработчике таймера расположен в начале алгоритма (сразу после **begin**). Если это не так, то переместите его в начало.
5. Добавьте в оператор case обработчик кодов 39 и 37. В качестве **действий** для этих кодов **присвойте ускорению значение 1 и -1 соответственно**.
6. Запустите приложение и проверьте, что сначала автомобиль стоит неподвижно, а с началом управления пользователем он начинает движение.

САМОСТОЯТЕЛЬНАЯ ЧАСТЬ

1. Аналогично проделанным работам, добавьте второй автомобиль, который движется с постоянной скоростью навстречу нашему автомобилю, и после выхода за левую границу формы вновь появляется справа. (Добавьте таймер и изображение 2-ого автомобиля).
2. Добавьте изображение с дорогой (из шаблона) позади всех других элементов формы. Разместите автомобили в соответствии с двусторонним движением.