**Уроки программирования в среде GAMBAS**

**Урок № 1: Знакомство со средой.**

**Обучающий игровой проект для малышей «Светофор»**

В прошлом году мы использовали среду программирования Кумир, в которой знакомились с различными исполнителями. Каждый из них – Робот, Черепашка, Рисователь, выполняли наши программы, а мы при создании программ использовали систему команд именно этого исполнителя.

В этом году для создания программ мы будем использовать *объектно*-ориентированный язык программирования Gambas. Программы в нем называются проекты и собираются из объектов, как из кубиков. Рассмотрим пример:



В этом проекте используются следующие объекты:

* **Форма** с заголовком «Приветствие»
* **Надпись** «Введите свое имя»
* **Текстовое поле**, где во время выполнения проекта пользователь написал «Вася»
* **Надпись**, сформированная с учетом введенного имени: «Здравствуй, Вася!»
* **Кнопка** с заголовком «Выход»

***Ход практической работы «Светофор»***

1. Среда программирования Gambas разработана для **операционной системы Linux**, поэтому при включении компьютера выбираем эту систему.

2. Запустить среду программирования GAMBAS: **К→Образование→Разработка→Gambas IDE**

3. Запустится **мастер создания приложений**, необходимо последовательно выбирать:

* на I этапе - «Новый проект...»:



* на II этапе ― тип проекта «Графическое приложение»:



Затем нажать кнопку «**Next**»

* на III этапе определить папку для размещения будущего проекта:



Если необходимо создать новый каталог, можно воспользоваться кнопкой со звездочкой (на рисунке обведена красной линией).

* на IV этапе нужно задать: имя проекта ― только английскими буквами (в указанной на III этапе папке будет создана папка для размещения данного проекта); и заголовок проекта (расшифровка, определяющая решаемую задачу – у нас это будет «Светофор *Фамилия*»).



4. Если все этапы пройдены верно, мы попадаем в окно редактора:



5. Выберем **главный объект – форму FMain** и настроим цвет:

* кликнуть на названии формы в окне навигатора
* настроить цвет формы в окне свойств: **Background** (выбрать на свой вкус, но не красный, желтый или зеленый)

6. Установим на форму 3 надписи – объект **Label** , одну под другой, и установим каждой цвет (моделируя светофор):

* кликнуть на обозначении объекта в палитре компонентов 
* двигаясь по форме с нажатой кнопкой мышки, указать положение и размер объекта
* в окне свойств настроить цвет фона каждой надписи, изменяя значение свойства **Background**: у верхней надписи – красный, у средней – желтый, у нижней – зеленый

**Важно! В окне свойств отображаются свойства *выделенного* объекта! Для выделения – клик на объекте.**

* убрать имя объекта, которое по умолчанию выводится на объект: свойство **Text**, значение удалить

7. Проверить работу проекта. Для этого запустить его на выполнение с помощью кнопки в панели инструментов.

Должно получиться примерно следующее:

х

Светофор Кошкиной

**Задание:** Записать в тетрадь название использованных объектов, их свойств и назначение этих свойств:

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Назначение свойства |
| Форма **FMain** – главный объект-контейнер для *размещения* остальных | Text | Заголовок формы |
| BackGround | Цвет фона формы |
| Надпись **Label** – прямоугольник на форме для *вывода* | Text | То, что выводится |
| BackGround | Цвет фона надписи |

**Обратить внимание, что они и те же свойства используются для разных объектов!**

8. Наш проект должен сообщать пользователю, что ему делать при каждом цвете светофора. Для этого нужно настроить процедуру **обработки события**: если пользователь кликнет на красный прямоугольник, в нем должно появиться слово «Стой!», если на желтый – слово «Приготовься…», а при клике на зеленый прямоугольник – слово «Иди.»

Чтобы это сделать, нужно:

* кликнуть 2 раза на надпись, которая имеет красный цвет фона
* откроется окно программного кода, а курсор будет находиться внутри процедуры обработки события «Клик по надписи Label1»: PUBLIC SUB Label1\_Click()

|

END

* внимательно посмотреть на заголовок процедуры и уточнить имя надписи ( в примере – Label1, а у вас может получиться и по-другому!)
* в процедуре (там, где находится курсор) набрать команду: Label1.Text = “Стой!”

(если в заголовке процедуры имя надписи Label**2**, то и команда Label**2**.Text = “Стой!” и т.д.)

* аналогично обработать две оставшиеся надписи, меняя только имена объекта и значение свойства Text.

9. Снова запустить проект на выполнение. Обязательно проверить событие «клик по надписи»! Правильно ли появляются сообщения?

Д/з № 1: 1. выучить структуру окна редактора, название объектов, их свойств

2. Какие объекты нужны для вычисления площади прямоугольника?

**Урок № 2: Еще элементы управления. Пишем на форме.**

**Проект «Площадь прямоугольника»**

Какие объекты (элементы управления) нужны для вычисления площади прямоугольника? Нужна возможность ввести значения длин сторон! Для этого используются объекты TextBox (текстовое поле). А для того, чтобы запустить процесс вычисления можно использовать кнопку Button.

***Ход практической работы «Площадь прямоугольника»***

1. Создать новый проект.

2. Настроить цвет формы по желанию, заголовок – «*Фамилия* вычисляет площадь».

3. На форме разместить 2 элемента управления Label со следующим значением свойства Text:

* Введите длину прямоугольника
* Введите ширину прямоугольника

4. Возле надписей разместите текстовые поля для ввода значений. В окне свойств очистите значение свойства Text.

5. Внизу или сбоку разместите кнопку Button , задайте значение свойства Text – «Вычислить»

6. Проверить работу проекта.

Должно получиться примерно следующее:

Кошкина вычисляет площадь

х

Введите длину прямоугольника

Введите ширину прямоугольника

Вычислить

**Задание:** Записать в тетрадь название использованных объектов, их свойств и назначение этих свойств:

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Назначение свойства |
| Текстовое поле **TextBox** – для *ввода значений* во время выполнения проекта | Text | Содержимое поля |
| Кнопка **Button** – для *запуска процессов* | Text | Надпись на кнопке |

7. При нажатии на кнопку значения текстовых полей должны перемножиться. Для этого нужно настроить процедуру обработки события «Клик по кнопке»:

PRINT Val(TextBox1.Text) \* Val(TextBox2.Text)

**Команда Print выводит сообщения в окно вывода – в нижней части окна.**

**Функция Val превратит текстовые данные в числовые.**

8. Проверить работу проекта, посчитать с его помощью площадь нескольких прямоугольников.

Д/з № 2: 1. выучить назначения и свойства новых объектов, команду Print, функцию Val

2. Продумать план проекта «Калькулятор»

**Урок № 3: Переменные и выражения. Новые свойства.**

**Проект «Калькулятор»**

Переменная – величина, значение которой изменяется в процессе выполнения проекта.

Каждая переменная занимает именованное место в оперативной памяти компьютера. Поэтому, чтобы определить переменную, необходимо дать ей имя и указать ее тип. Для этого в начале процедуры нужно дать команду:

Dim *Имя* As *Тип*

Имена переменных состоят из латинских букв и цифр.

|  |  |
| --- | --- |
| Тип переменной | Диапазон значений |
| Byte | Целые 0 .. 255 |
| Integer | Целые -32768 .. +32767 |
| Single | Вещественные |

Например, команда «DIM X As Byte, Y As Byte, Z As Integer» определяет 2 целые беззнаковые величины X и Y (они будут принимать небольшие положительные значения) и одну целую величину Z, которая может быть и отрицательной.

Для изменения значения переменной используется **оператор присваивания**. В языке программирования Gambas он обозначается знаком равенства. Например, команду:

x = Val(TextBox1.Text)

следует понимать так: переменная x получает числовое значение из текстового окна Edit1.

А команду «F = 2 + 4 /3»: посчитать значение выражения справа от знака присваивания и результат положить в переменную F.

Пусть дан фрагмент программного кода: X = 2

Y = 5

X = 2 \* Y + X

Y = X ^ 2 - Y

Переменные последовательно получают значения:

|  |  |
| --- | --- |
| X | Y |
| 2 |  |
|  | 5 |
| 12 |  |
|  | 139 |

При вычислении выражений необходимо учитывать приоритет выполнения арифметических операций:

1. Возведение в степень: ^
2. Умножение и деление: \*, /
3. Сложение и вычитание: +, -

Операции одинакового приоритета выполняются последовательно. Изменение порядка выполнения операций происходит с помощью скобок.

Например, программируя расчет выражения , можно в отдельные переменные присвоить части выражения, а можно «вытянуть» выражение в одну команду:

y = x ^ k / (k + 1 / (x + 2 \* k))

***Ход практической работы «Калькулятор»***

1. Создать новый проект.

2. Настроить цвет формы по желанию, заголовок – «Калькулятор *Фамилия*».

3. На форме разместить 2 элемента управления TextBox, настроить их свойства:

|  |  |  |
| --- | --- | --- |
| Alignment | Выравнивание | Center |
| Font | Размер и начертание шрифта | Размер +3 |
| ForeGround | Цвет символов | Выбрать из набора свободных цветов |

4. Внизу или между текстовыми окнами разместите кнопки с заголовками: +, -, \*, /, x^y

5. Поставьте элемент управления Надпись для вывода результатов вычислений, настройте свойства Alignment, Font, ForeGround.

6. Проверить работу проекта.

7. На каждую кнопку назначить процедуру обработки события Click. Например, для операции сложения это будет следующий программный код:

Dim x As Single, y As Single

x = TextBox1.Text

y = TextBox2.Text

Label1.Text = x + y

8. Проверить работу проекта.

Д/з № 3: 1. выучить новые свойства

2. выполнить задания по карточке (один вариант)

**6 Инф Д/з «Переменные»**

1. Определить значение переменных после команд присваивания:

A := 15; Z := 10;

A := (A + 5) / Z;

Z := Z \* A;

A := A \* A

2. Составить блок-схему и программу решения задач, построить протокол:

* Найти объем куба.
* Найти значение выражения 

**6 Инф Д/з «Переменные»**

1. Определить значение переменных после команд присваивания:

X := 0; K := 1;

K := 5 \* K;

X := X + K;

K := X

2. Составить блок-схему и программу решения задач, построить протокол:

* Найти площадь поверхности куба.
* Найти значение выражения 

**6 Инф Д/з «Переменные»**

1. Определить значение переменных после команд присваивания:

Y := 5; L := 10;

A := (Y + L) / 2;

Y := Y \* Y;

L := L \* A

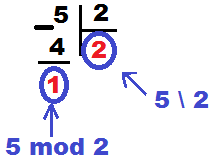
2. Составить блок-схему и программу решения задач, построить протокол:

* Найти площадь квадрата, длина стороны которого в 2 раза больше заданной длины А.
* Найти значение выражения 

**Урок №4: Целочисленная арифметика**

Вы уже знаете, какие арифметические операции можно осуществлять в языках программирования: сложение, вычитание, умножение, деление, возведение в степень. Но бывают случаи, когда при делении не нужно знать результата целиком, а нужно получить отдельно целую часть и остаток отделения. Вот примеры таких задач:

* Найти сумму цифр двухзначного числа.
* В вагон помещается ровно 5 тонн груза. Сколько вагонов нужно для перевозки 42389 кг?
* Сколько плиток размером 10х10 см нужно для оклейки стены размером АхВ м?

Операторы целочисленной арифметики – это:

целая часть от деления \

остаток от деления mod

Найти значения переменных:

Х = 15: Y = 4

Z = X \ 4

X = X mod Y

Д/з № 4: 1. выучить новые операторы

2. выполнить задания по карточке (один вариант)

**6 Инф Д/з «Целочисленная арифметика»**

1. Найти сумму первой и последней цифры 4-хзначного числа.

2. Дана масса в центнерах. Найти число полных тонн в нем.

3.\* Дано 3-хзначное число. Найти число, образованное перестановкой его последней цифры вперед.

**6 Инф Д/з «Целочисленная арифметика»**

1. Найти сумму первой и последней цифры 3-хзначного числа.

2. Дано расстояние в см. Найти число полных метров в нем.

3.\* Дано 3-хзначное число. Найти число, образованное перестановкой его первой цифры назад.

**6 Инф Д/з «Целочисленная арифметика»**

1. Найти сумму первой и последней цифры 5-тизначного числа.

2. Дана масса в кг. Найти число полных центнеров в нем.

3.\* Дан прямоугольник с размерами 543х130 мм. Сколько квадратов со стороной 130 мм можно отрезать от него.

**Урок №5: Рисуем на форме!**

Для рисования на форме используются графические методы: Объект.Метод(аргументы)

I этап. Определить область рисования: Draw.Begin(Fmain)

Форма превращается в холст для рисования с системой координат:

х

у

Единица измерения – пиксель.

Каждая точка определяется парой целых чисел – координатами (х, у).

II этап. Графические методы:

|  |  |  |
| --- | --- | --- |
| Метод | Действие | Пример |
| Draw.Line(xl, yl, х2, y2) | вычерчивает отрезок прямой между точками с координатами (x1, y1) и (х2, у2) | Draw.Line(0, 0, 100, 100) |
| Draw.Point(x, y) | вычерчивает точку с координатами (x,y) | Draw.Point(150, 200) |
| Draw.Rect(x, у, w, h) | чертит прямоугольник, задаваемый координатами верхнего левого угла (х, у), шириной w пикселей и высотой h пикселей | Draw.Rect(100,500,20,10) |
| Draw.Polygon(x1, y1, ..., xn, yn) | чертит замкнутый многоугольник, последняя точка соединяется с первой | Draw.Polygon(100,1, 100,50, 120, 1) |
| Draw.Circle(x, у, r) | чертит окружность с центром в точке (x,y) и радиусом r | Draw.Circle(100, 30, 20) |
| Draw.Ellipse(x, у, w, h) | чертит овал, вписанный в прямоугольник, заданный аргументами метода | Draw.Ellipse(50,40,10,20) |

Перед рисованием можно задать цвет линий:

* Draw.ForeColor = Color.Black ← установка системного цвета
* Draw.ForeColor = Color.RGB(К, З, С) ← установка пользовательского цвета смешиванием красной, зеленой и синей составляющих

или сразу определить цвет закраски фигуры: Draw.FillColor = Color.Цвет

III этап. Закрыть область рисования: Draw.End

Д/з № 5: 1. выучить операторы рисования

2. продумать план рисования своего рисунка