Лабораторная работа 8. Создание маплетов с помощью пакета Maplets.

Цель: создание простейших Maplet-приложений(Маплетов), используя пакет Maplets.

## Создание с помощью пакета Maplets

Пакет Maplets содержит 4 подпакета и функцию Display.

**with(Maplets);**



Чтобы создать Maplet-приложение, необходимо определить структуру и поведение маплета с помощью команд подпакетов Elements, Tools, и Utilities.

Подпакет Elements содержит отдельные компоненты, используемые для создания Маплетов, например, окна, кнопки и флажки, включая опции, которые определяют их положение и внешний вид.

Подпакеты Tools и Utilities содержат дополнительные команды для создания маплетов.

Для начала работы необходимо подключить используемые пакеты:

**with(Maplets):**

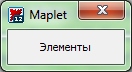
**with(Elements):**

Далее задаем имя Маплета и присваиваем ему функцию **Maplet**. В ней располагаются все элементы маплета. Элементы вводятся в виде списка.

**mpt1:=Maplet([Элементы]):**

Для запуска Маплета используем команду **Display**

**Display(mpt1);**



#### Пример 1. Создание маплета, расположение элементов:

**mpt2:=Maplet(**

**[**

**["Hello There"],**

**["1","2","3"],**

**["ABCDEFG","HIJKLMNOPQR"],**

**[Button("OK",Shutdown())]**

**]**

**):**

**Display(mpt2);** - выполнение маплета.

|  |  |
| --- | --- |
|  | Элементы внутри функции **Maplet** группируются с помощью списков. Здесь присутствуют надписи и кнопка. Кнопка выполняет функцию закрытия маплета. |

Каждому элементу можно задать имя-ссылку, для дальнейшего обращения к этому элементу. Имя задается в квадратных скобках после команды, задающей элемент:

**Button[b1]("OK",Shutdown())**  Имя кнопки b1

#### Пример 2. Создание маплета, расположение элементов:

**mpt3:=Maplet(Window("Title",[**

**"ABC",**

**["DEF","GHI",["J","K","L"]],**

**["M",["OP","QR",["S","T","U"]]]**

**])**

**):**

**Display(mpt3);**

|  |  |
| --- | --- |
|  | В данном маплете использован элемент Window-окно, который содержит все другие элементы. Изменен параметр заголовок окна.  Т.к. почти каждый маплет представляет собой окно, то лучше указывать этот элемент всегда. Ему также можно задать имя в квадратных скобках и обращаться в дальнейшем. |

#### Пример 3. Задание свойств элементам:

**mpt4:= Maplet(**

**Window( "A Colourful Button",**

**[Button("OK", background=blue, enabled=false,**

**tooltip="Can't click me", Shutdown())],**

**width=200,**

**height=200)**

**):**

**Display( mpt4 );**

|  |  |
| --- | --- |
|  | Свойства элементов задаются в виде: опция=значение и перечисляются последним параметром.  Т.е. в кнопке **Button** сначала идет надпись на кнопке, потом перечисляются свойства.  В элементе **Window** сначала заголовок, потом элементы, в конце – свойства: ширина, высота. |

#### Пример 4. Использование ссылок на элементы:

|  |  |
| --- | --- |
| В этом маплете введеное в текстовое поле значение не возвращается: | |
| **mpt5:=Maplet(Window("Text Maplet",[**  **[TextField()],**  **[Button("OK",Shutdown())]**  **])**  **):**  **Display( mpt5);** |  |
| Доработаем этот маплет.  Текстовому полю дадим имя **input**. В функции Shutdown выводим значение переменной **input**. В результате после нажатия на кнопку в рабочую область после работы маплета будет выведено значение из текстового поля. Если окно просто закрыть, то ничего не будет выведено. | |
| **mpt5:=Maplet(Window("Text Maplet",**  **[**  **[TextField[input]()],**  **[Button("OK",Shutdown([input]))]**  **])**  **):**  **Display( mpt5 );** |  |
| В следующем маплете текстовым полям даны имена **input1** и **input2**. Далее после работы функции **Shutdown** возвращаются значения переменных **input1**, **input2** и значения свойств первого текстового поля **visible**-видимость и **background**-цвет фона. | |
| **mpt5:=Maplet(Window("Text Maplet”,**  **[**  **[TextField[input1]()],**  **[TextField[input2]()],**  **[Button("OK",Shutdown([input1,**  **input2,input1(visible),**  **input1(background)]))]**  **])**  **):**  **Display( mpt5);** |  |

Любую опцию элементов можно изменить или вывести, используя синтаксис:

**имя\_элемента(название\_опции)**

#### Пример 5. Действия и команды

Действия и команды:

CloseWindow - закрыть окно

Evaluate – вычислить

RunDialog – запуск диалога

RunWindow – запуск окна

SetOption – изменить опцию элемента

Shutdown – закрытие маплета.

Функция Shutdown завершает работу маплета, а функция CloseWindow закрывает указанное окно. Если окно единственное (или последнее), то происходит выход из маплета аналогичный Shutdown.

|  |  |
| --- | --- |
| Выход по двум кнопкам Shutdown: | |
| **mpt6:=Maplet(Window("2 buttons",[**  **["Hello World"],**  **[Button("OK 1",**  **Shutdown("Button 1")),**  **Button("OK 2",**  **Shutdown("Button 2"))]**  **])):**  **Display(mpt6);** | "Button 1" |
| Кнопки выполняют одну из внешних процедур: | |
| Задаем две внешние процедуры:  **procOk:=proc()print("OK was pressed");end proc:**  **procCancel:=proc()print("Cancel was pressed");end proc:**  Далее маплет, который их использует  **mpt7:=Maplet(Window("Title",[**  **[Button("OK",**  **Evaluate(function='procOk()')),**  **Button("Cancel",**  **Evaluate(function='procCancel()'))**  **],**  **[Button("Exit",Shutdown())]**  **])):**  **Display( mpt7 );** | При нажатии на кнопки в рабочую область выводится соответствующий текст. При нажатии на выход – закрытие маплета. |
| **Изменение параметров элементов SetOption.**  Кнопка Exit имеет имя btExit. При нажатии на кнопки меняется ее свойство enabled. | |
| **mpt8:=Maplet(Window("Set Option",[**  **[Button("Enable",**  **SetOption(btExit(enabled)=true)),**  **Button("Disable",**  **SetOption(btExit(enabled)=false)),**  **Button[btExit]("Exit",Shutdown())]**  **])):**  **Display( mpt8);** |  |
| **mpt9:=Maplet(Window("Set Option",[**  **[TextField[TF](),**  **Button("Clear",SetOption(TF="")),**  **Button[exit]("Exit",Shutdown())]**  **])**  **):**  **Display( mpt9 );** | Меняется значение текстового поля. Его имя TF, при нажатии на кнопку оно очищается. |

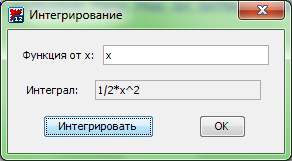
#### Пример 6. Взаимодействие между маплетом и Maple

Для взаимодействия между маплетом и Maple используются методы Get и Set. Они находятся в подпакете Tools пакета Maplets.

**with(Tools);**



Маплет



Содержит:

текстовое поле для введение функции – имеет имя **F**.

текстовое поле для значения интеграла – имеет имя **intF**.

кнопка выполняющая интегрирование. Она запускает процедуру **getInt**, написанную отдельно.

кнопка выход со значением из текстового поля **intF.**

**mpt10:= Maplet(Window("Интегрирование",[**

**["Функция от х: ",TextField[F]()],**

**["Интеграл: ",TextField[intF](editable=false)],**

**[Button("Интегрировать",Evaluate(function='getInt()')),**

**Button("OK",Shutdown([intF]))]**

**])):**

Процедура:

**getInt:=proc()**

**local expr;**

**expr:=Get(F::algebraic);** берется значение из поля **F** – тип алгебраическое, т.к. с текстовым не будет работать интегрирование.

**expr:=int(expr,x);** находим интеграл

**Set(intF=expr);** возвращаем найденное значение в маплет в поле intF. Здесь тоже можно указать или не указывать тип для **intF**.

**end proc:**

#### Пример 7. Работа с несколькими окнами.

Для задания нескольких окно необходимо перечислить их в теле маплета. Следующий пример вызовет ошибку, т.к. не указано какое окно загружается в начале.

**maplet:=Maplet(**

**Window("Hi",["Hi"]),**

**Window("Bye",["Bye"])**

**);**

**Display(maplet);**

Error, (in Maplets:-Tools:-CleanUp:-Maplet) there is no 'onstartup' attribute and no unique window or dialog which may run on startup

Поэтому необходимо обязательно указывать это в опции окна onstartup:

|  |  |
| --- | --- |
| **maplet:=Maplet(onstartup=RunWindow(W1),**  **Window[W1]("Hi",["Hi"]),**  **Window[W2]("Bye",["Bye"]));**  **Display(maplet);** |  |

В предыдущем примере маплет содержит два окна, но используется только одно. Запустим второе окно при нажатии на кнопку. Для этого используем функцию Action, в ней закрываем первое окно и запускаем второе.

**maplet:=Maplet(onstartup=RunWindow(W1),**

**Window[W1]("Hi",[**

**Button("OK",Action(CloseWindow(W1),RunWindow(W2)))]**

**),**

**Window[W2]("Bye",[Button("Okay",Shutdown())]**

**)**

**);**

**Display( maplet );**

|  |  |
| --- | --- |
|  |  |

#### Пример 7. Использование диалогов.

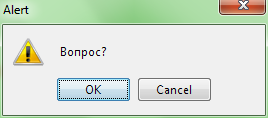
Диалоги имеют готовое стандартное окно.

Существует несколько видов диалогов:

AlertDialog ColorDialog ConfirmDialog FileDialog InputDialog

MessageDialog QuestionDialog

Рассмотрим AlertDialog. Пользователь может нажать только да/нет.



**maplet:=Maplet(AlertDialog("Вопрос?",**

**'onapprove'=Shutdown("true"),**

**'oncancel'=Shutdown("false")**

**)):**

**Display(maplet);**

"true"

Использование ColorDialog. Запускается диалог выбора цвета, в рабочую область возвращается выбранное значение.

|  |  |
| --- | --- |
| **maplet:=Maplet(ColorDialog[D1](**  **'onapprove' = Shutdown( [D1] ),**  **'oncancel' = Shutdown()**  **) ):**  **Display( maplet );** |  |

Диалоги можно запускать по нажатии кнопки.

#### Пример 8. Создание меню.

Маплет содержит меню File. В нем два пункта – Hello и Exit. При нажатии на Hello запускается ConfirmDialog, при нажатии на выход – выход.

|  |  |  |
| --- | --- | --- |
|  |  |  |

**maplet := Maplet( onstartup = RunWindow( W1 ),**

**Window[W1]( "With Menu",**

**MenuBar( Menu( "File",**

**MenuItem( "Hello", RunDialog( D1 ) ),**

**MenuItem( "Exit", Shutdown() )**

**) ),**

**["Some Text"]**

**),**

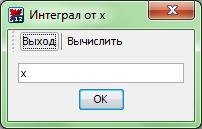
**ConfirmDialog[D1]( "Are you fine?" )**

**):**

**Display( maplet );**

#### Пример 9. Создание панелей инструментов.

Для создания панелей инструментов задается опция окна **toolbar,** ее значение **ToolBar(**перечисление элементов панели**)**



**maplet := Maplet(**

**Window(title = "Интеграл от x",**

**toolbar = ToolBar(**

**ToolBarButton("Выход", Shutdown()),**

**ToolBarSeparator(),** вертикальная черта в меню

**ToolBarButton("Вычислить",**

**onclick=Evaluate(TF1='int(TF1,x)'))**

**),**

**[TextField[TF1](value='x'),**

**Button("OK", Shutdown("OK"))]**

**)**

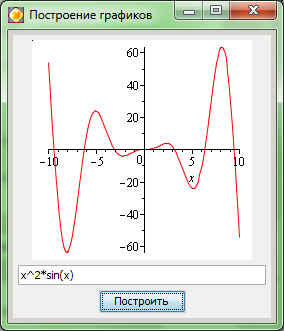
**):**

**Display(maplet):**

## Задание 1.

Создать маплет, который строит график заданной функции:

3 элемента: Plotter, TextField, Button. В текстовое поле вводится функция от х, при нажатии кнопки строится график.



## Задание 2.

Создать маплет, решающий квадратное уравнение. Уравнение вводится пользователем, после решения выводятся найденные действительные корни или сообщение «нет корней».

## Задание 3.

Создать маплет, который ищет длину вектора, заданного в виде:

1) координатами начала и конца вектора.

2) координатами вектора (в форме a\*i+b\*j)

Т.е. необходимо сделать 2 вида ввода. Можно использовать:

1) Два разных окна, вызываемых по двум кнопкам.

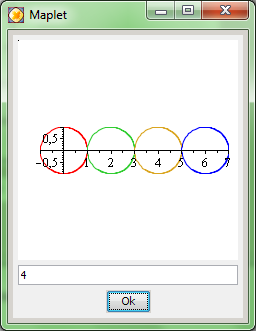
2) Два разных окна по одной кнопке и одному из переключателей: DropDownBox, ComboBox, RadioButton и т.п., которыми выбирается способ ввода координат.

3) одно окно и переключатели из п.2.

## Задание 4.

Создать маплет, рисующий заданное количество окружностей.

Ограничьте количество вводимых окружностей, например от 1 до 5, если введено неверное значение – выводится сообщение об ошибке.



## Задание 5.

Доработать предыдущий маплет. Сделать выбор (поставить переключатель) рисовать окружности или квадраты. Ограничьте количество вводимых объектов, например от 1 до 5, если введено неверное значение – выводится сообщение об ошибке.

## Задание 6.

Сделать маплет рисующий квадрат или окружность заданной площади. Площадь вводится числом. Выбор объекта осуществляется переключателем. Ограничьте вводимую площадь разумным значением, чтобы объект помещался на маплете. Если введено неверное значение – выводится сообщение об ошибке.