

Объектно-ориентированное программирование

Лабораторная работа №1

Пример

Создать класс:

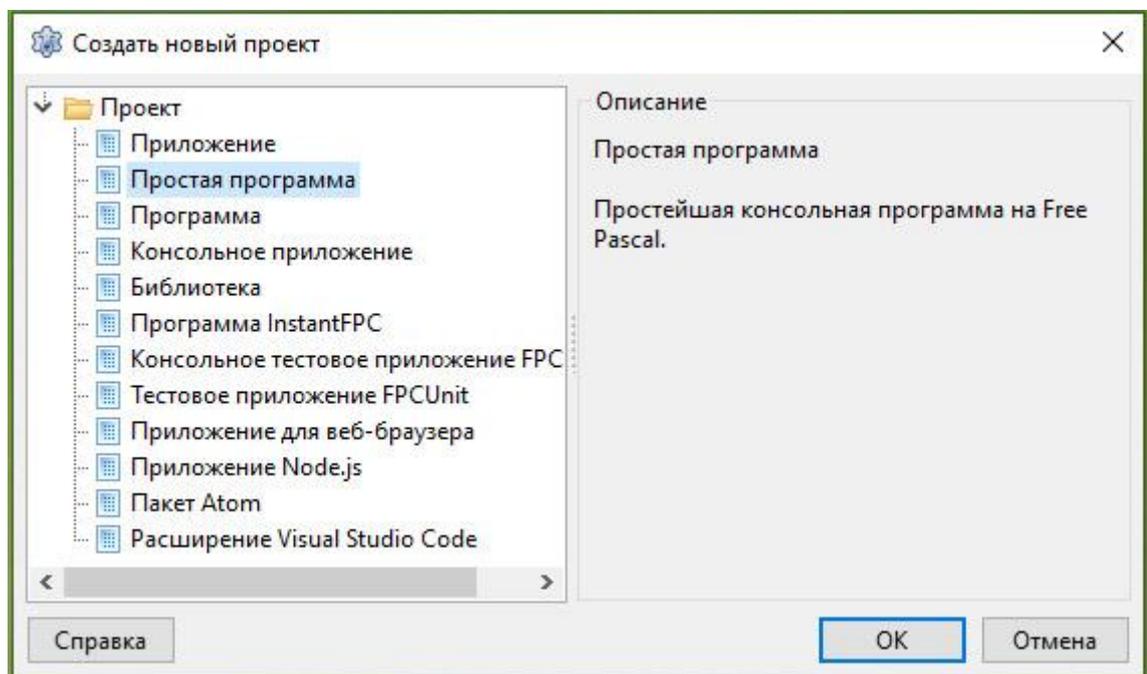
Комплексное число в алгебраической форме $C = x + iy$.

Члены класса: Действительная x и мнимая y части числа.

Методы класса: метод вычисления модуля комплексного числа и метод вывода комплексного числа.

Продемонстрировать работу класса на своём примере.

Выбираем в Lazarus при создании проекта вариант «Простая программа» (см рис.)



Получим заготовку – текст из трёх строк:

```
program Project1;
```

```
begin
```

```
end.
```

Мы потом этот текст заменим своим.

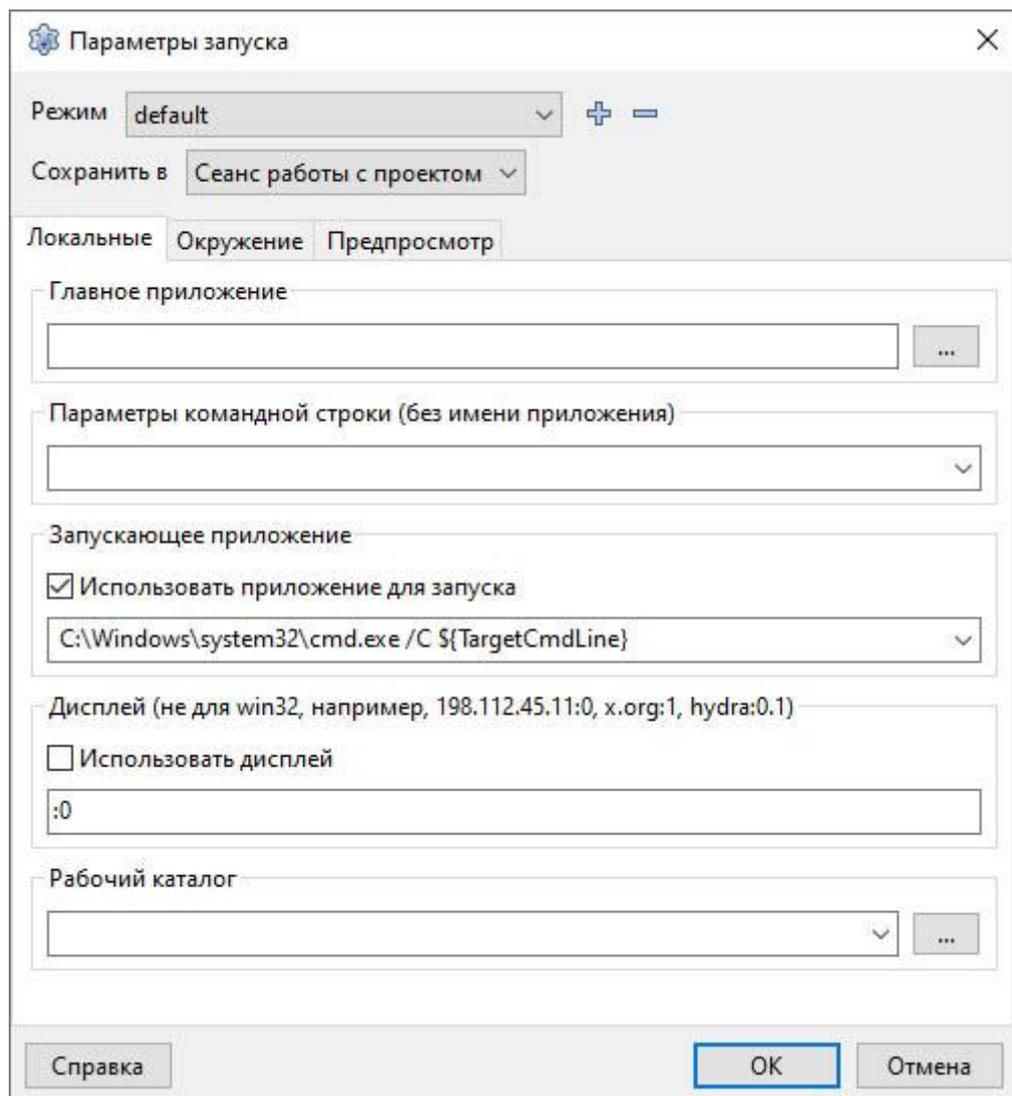
Далее в меню Lazarus выбираем «Файл» – «Создать модуль».

В результате образуется небольшая заготовка – файл Unit1.pas.

Сохраним пока эти заготовки в свою папку (пусть это будет Primer1).

Выбираем в меню «Проект» – «Сохранить проект как», находим свою папку и называем проект MyPrimer1. Далее будет предложено сохранить модуль unit1, мы его назовём, например, MyUnit1. В названиях папок в Linux не должно быть пробелов.

Поскольку мы будем работать в консольном режиме, то в Linux для этого надо настроить параметры запуска программы – через меню «Запуск» – «Параметры запуска»: отмечаем здесь «Использовать приложение для запуска» (см. рисунок ниже).



Теперь текст в MyUnit1.pas заменяем своим:

```
unit MyUnit1;
```

```
{ $mode ObjFPC } { $H+ }
```

```
interface
```

```
uses
```

```
Classes, SysUtils;
```

```
type
```

```
TKomplex = class  
private  
    x,y:double;  
    procedure SetKomplex(x1,y1:double);  
public  
    constructor Build(x1,y1:double);  
    function GetModule:double;  
    procedure Output;  
end;
```

```
implementation
```

```
constructor TKomplex.Build(x1,y1:double);  
begin  
    SetKomplex(x1,y1);  
end;
```

```
procedure TKomplex.SetKomplex(x1,y1:double);  
begin  
    x:= x1;  
    y:= y1;  
end;
```

```
function TKomplex.GetModule:double;  
begin  
    result:= sqrt(x*x + y*y);  
end;
```

```
procedure TKomplex.Output;  
begin  
    writeln(format('C = %.2f + i%.2f', [x,y]));  
end;
```

```
end.
```

Текст в файле MyPrimer1.lpr заменяем следующим:

```

program MyPrimer1;

{ $mode objfpc } { $H+ }

uses
    { $IFDEF UNIX }
    cthreads,
    { $ENDIF }
    Classes, SysUtils, MyUnit1;

var
    C: TKomplex;
    x,y: double;
    s: string;

begin
    write('x=');
    readln(s);
    x:= StrToFloat(s);
    write('y=');
    readln(s);
    y:= StrToFloat(s);
    C:= TKomplex.Build(x,y);
    writeln(format('Module=%.2f', [C.GetModule]));
    C.Output;
    // далее подобные действия для остановки Linux подключает сам
    { $IFDEF windows }
        writeln('-----');
        writeln('Press <Enter>');
        readln;
    { $ENDIF }
end.

```

Изучите внимательно данный пример. Обратите внимание, что для класса мы создаём отдельный модуль и потом его используем в своём проекте.

Задачи для самостоятельного решения

1. Создать класс:

Вектор.

Члены класса: три прямоугольные декартовы координаты.

Методы класса: метод вывода вектора, метод вычисления длины вектора.

Продемонстрировать работу класса на своём примере.

2. Создать класс:

Матрица.

Члены класса: Размерность матрицы, элементы матрицы.

Методы класса: метод вывода матрицы, метод вычисления определителя матрицы, проверка, является ли матрица диагональной, нулевой, единичной, проверка, является ли матрица симметричной ($A^T=A$).

Продемонстрировать работу класса на своём примере.

3. Создать класс:

Прямая.

Члены класса: Координаты двух точек (x_1, y_1) и (x_2, y_2) .

Методы класса: метод вывода уравнения прямой.

Продемонстрировать работу класса на своём примере.

4. Создать класс:

Парабола $y = ax^2 + bx + c$.

Члены класса: Коэффициенты a , b , c .

Методы класса: методы вывода уравнения параболы, вычисления экстремума функции (min или max).

Продемонстрировать работу класса на своём примере.

5. Создать класс:

Время.

Члены класса: Часы, минуты, секунды.

Методы класса: метод вывода времени и составляющей суток (до 6 – ночь, до 12 – утро, до 18 – день, до 24 – вечер)

Продемонстрировать работу класса на своём примере.