

## Практические задания

### Практическое задание 1. Разработка и использование классов для графических объектов.

#### Постановка задачи.

Требуется создать классы, описывающие графические объекты - окружности и прямоугольники. Классы должны содержать необходимые параметры-свойства, конструктор с параметрами, метод отображения Show, метод для перемещения объекта MoveTo. Классы оформляются в виде отдельных модулей, которые подключаются к основной программе для демонстрации возможностей созданных классов. Демонстрационный модуль создается как оконная программа с минимально необходимым набором команд меню.

**Порядок выполнения работы** (для инструментальных пакетов Delphi/Lazarus):

1. Создать новый проект и сохранить его в собственном каталоге.
2. Разместить на главной форме основное меню для последующей реализации команд демонстрации возможностей создаваемых классов. На начальном этапе достаточно в меню поместить одну команду «Окружность» с подкомандами «Создать» и «Переместить».
3. Разместить на форме компонент Image для отображения объектов и выровнять его на всю клиентскую область.
4. Создать обработчик события формы OnCreate для инициализации датчика псевдослучайных чисел и установки режима рисования на поверхности компонента Image:  

```
Image1.Canvas.Pen.Mode := pmNotXOR;
```

этот режим позволяет стирать существующее изображение за счет повторного вывода на то же самое место с автоматическим восстановлением фона.
5. Добавить в проект новый программный модуль, НЕ СВЯЗАННЫЙ со своей формой: команда меню File / New / Unit и обязательно сохранить его

вместе с проектом. Поскольку модуль является чисто программным, то его заготовка содержит лишь заголовок, начало интерфейсной секции (директива `interface`) и начало секции реализации (директива `implementation`).

6. В интерфейсном разделе нового модуля ввести описание (ТОЛЬКО ОПИСАНИЕ!) класса окружностей (в разделе описания типов). В классе объявить поля данных закрытыми, а методы – открытыми. Пример описания класса находится в разделе 2 пособия.

7. Оформить раздел реализации модуля, включив в него программную реализацию каждого объявленного метода. Например:

```
procedure TCircle.MoveTo(dx, dy : integer);
begin
    Show; // повторный вывод для стирания окружности
    x := x + dx; y := y + dy; // изменение координат центра
    Show; // рисование в новом месте
end;
```

В методе `MoveTo` для удобства дальнейшего использования задаются не новые координаты центра, а СМЕЩЕНИЕ относительно него, которое может быть и отрицательным.

8. Программная реализация метода показа окружности может использовать стандартные графические примитивы системы Windows. Например, вывод окружности выполняется следующим образом:

```
Form1.Image1.Canvas.Ellipse(x-r, y-r, x+r, y+r);
```

Поскольку этот код использует компонент `Image` основной формы, необходимо подключить модуль главной формы к модулю с описанием класса окружностей, записав сразу после директивы `implementation` строку

```
uses ИмяГлавногоМодуля;
```

9. После полного описания всех методов сохранить модуль и подключить его к основному модулю.

10. В разделе реализации основного модуля объявить одну-две объектные переменные классового типа `TCircle` (см. раздел 3 пособия).

11. Написать обработчик команды создания окружности, выполняющий следующие действия (см. раздел 3 пособия):

- создание нового объекта за счет вызова конструктора со случайными параметрами;
- отображение объекта на экране за счет вызова метода Show.

12. Написать обработчик команды перемещения окружности в новую точку со случайными координатами, используя генерацию значений смещения координат в пределах [-100, 100].

13. Сохранить проект и проверить работу программы.

14. Добавить в программу возможность использования массива ссылок на окружности (см. раздел 3 пособия), для чего:

- добавить в меню новые подкоманды «Создать Массив» и «Переместить Массив»
- в раздел реализации демо-модуля добавить объявление массива объектных переменных
- написать обработчики команд меню, выполняющие показ и перемещение сразу всего набора окружностей.

15. Сохранить проект и проверить его работу.

16. Добавить в проект новый модуль для класса прямоугольников

17. В интерфейсном разделе нового модуля ввести описание класса прямоугольников.

18. Оформить раздел реализации модуля, включив в него программную реализацию каждого объявленного метода.

Программная реализация метода показа прямоугольника выполняется с помощью стандартного вызова Rectangle:

`Form1.Image1.Canvas. Rectangle (x, y, x+ширина, y+высота);`

(обязательно подключить модуль главной формы к модулю с описанием класса прямоугольников)

19. После полного описания всех методов сохранить модуль и подключить его к основному модулю.

20. В раздел реализации основного модуля добавить переменную класса прямоугольников, а также – массив объектных переменных для выполнения групповой обработки объектов-прямоугольников.

21. Добавить в основное меню новую команду «Прямоугольник», содержащую подкоманды для создания и перемещения как одиночного прямоугольника, так и набора прямоугольников

22. Написать обработчики объявленных подкоманд

23. Сохранить проект и проверить его работу.