

СОЗДАНИЕ ПРИЛОЖЕНИЯ, РЕАЛИЗУЮЩЕГО ПРИНЦИП ПОЛИМОРФИЗМА

Цель работы: Изучить правила создания дочерних классов

1. Краткие сведения

Одним из основных принципов ООП является наследование. Этот принцип означает, что если необходимо создать новый класс, лишь немного отличающийся от старого, то совершенно нет необходимости в переписывании заново уже существующих полей и методов. Для этого требуется объявить новый класс, например, TNewObject

```
TNewObject = class(TOldObject).
```

Он является потомком или дочерним классом старого класса TOldObject, называемого предком или родительским классом. Далее следует добавить к нему новые поля, методы и свойства — иными словами, то, что нужно при переходе от общего к частному.

Унаследованные от класса-предка поля и методы доступны в дочернем классе. Если имеет место совпадение имен методов, то говорят, что они перекрываются.

Поведение методов при наследовании является краеугольным камнем объектно-ориентированного программирования. В зависимости от того, какие действия происходят при вызове, методы делятся на три группы. К первой группе относятся статические методы, ко второй — виртуальные (virtual) и динамические (dynamic) и, наконец, к третьей — появившиеся только в Delphi 4 — перегружаемые (overload) методы.

Методы первой группы полностью перекрываются в классах-потомках при их переопределении. При этом можно полностью изменить объявление метода. Методы второй группы при наследовании должны сохранять наименование и тип. Перегружаемые методы дополняют механизм наследования возможностью использовать нужный вариант метода (собственный или родительский) в зависимости от условий применения.

2. Постановка задачи

Разработать дочерний класс вычисления определенного интеграла с различными подинтегральными функциями. Реализовать возможность арифметических операций над интегралами с выбранными функциями. Примерный вид приложения показан на рис.1.

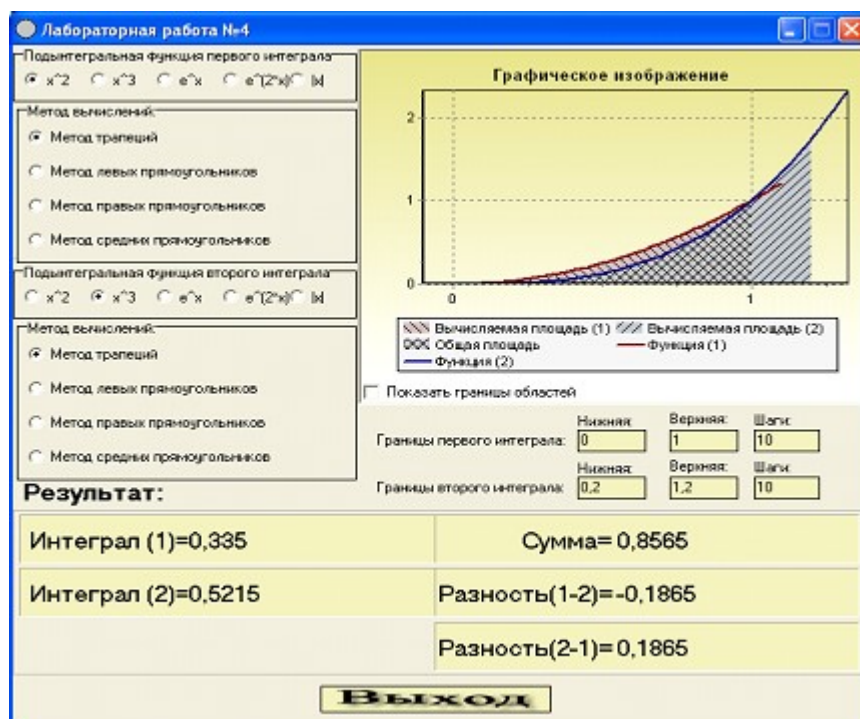


Рис.1. Вычисление определенного интеграла

3. Задания

Разработать приложение в соответствии с л.р. №1 и п.2.