МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

Кафедра веб-технологий и компьютерного моделирования

Курсовая работа

Методические указания и рекомендации по применению классов в Free Pascal

Выполнил:  
Студент 2-ого курса 3-ей группы  
Трифонов Вадим

Минск, 2019

Оглавление

[Оглавление 2](#_Toc7596047)

[Введение 3](#_Toc7596048)

[Понятие объектно-ориентированного программирования 4](#_Toc7596049)

[Абстрактные типы данных 5](#_Toc7596050)

[Инкапсуляция 6](#_Toc7596051)

[Наследование 7](#_Toc7596052)

[Конструкторы и деструкторы 8](#_Toc7596053)

[Виртуальные и статические методы 9](#_Toc7596054)

[Полиморфизм 10](#_Toc7596055)

[Литература 11](#_Toc7596056)

Введение

Данное учебно-методическое пособие предназначено для ознакомления и введению в тему “Объектно-Ориентированного Программирования на языке Pascal”.  
Основная цель – максимально понятно и доступно показать каким образом реализуется идеология ООП(Объектно-Ориентированное Программирование) на языке Pascal, сформулировать основные принципы использования. В данном методическом пособии будет описываться основной теоретический материал для каждой из тем и подкрепляться различными задачами с их дальнейшим решением. Это поможет лучше освоить материал и попробовать его самому для лучшего понимания.   
В методичке не изложен материал для изучения азов языка Pascal, предполагается, что вы уже знакомы с основами языка.  
  
Все примеры в учебном пособии были написаны в среде: Free Pascal версии 1.0.12

Понятие объектно-ориентированного программирования

Объектно-Ориентированное Программирование (ООП) – идеология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса.

**Классы и объекты.**В ООП существует два ключевых понятия: **Класс** и **Объект.**

* **Класс** – это абстрактный тип данных (будет рассмотрено позже). Позволяющий описать некоторые характеристики и возможные действия с его экземпляром.
* **Объект** – конкретный представитель класса.

Объекты в ООП представляется в виде некоторых блоков кода, которые относятся к определённому заранее классу. Данные, которые содержит объект, принято называть *атрибутами объекта*. Они позволяют описать состояние объекта и его характеристики.  
Если же нам необходимо взаимодействовать с атрибутами некоторого объекта или же описать его поведение, то необходимо задать процедуры и функции для этого объекта. В ООП функции и процедуры объектов называют *методами*. Но задавать мы их будем не для каждого объекта по отдельности, а для класса, которому принадлежит этот объект. Вызов метода происходит путём обращения к объекту и через точку задается имя вызываемого метода (ИмяОбъекта.ИмяМетода), это происходит в связи с тем, что вызванный метод работает с данными того объекта, относительно которого она был вызван (позже будет приведён пример). Каждый метод имеет доступ к атрибутам объекта.  
Класс является некоторой базой, обладающей всей информацией необходимой для создания объекта: общими данными, характерными для каждого объекта данного класса; описанием поведения объектов (методов)

Перейдём к реализации в FreePascal. Для начала необходимо описать наш первый класс в секции type, делается это путём использования ключевого слова Class (Рисунок 1):

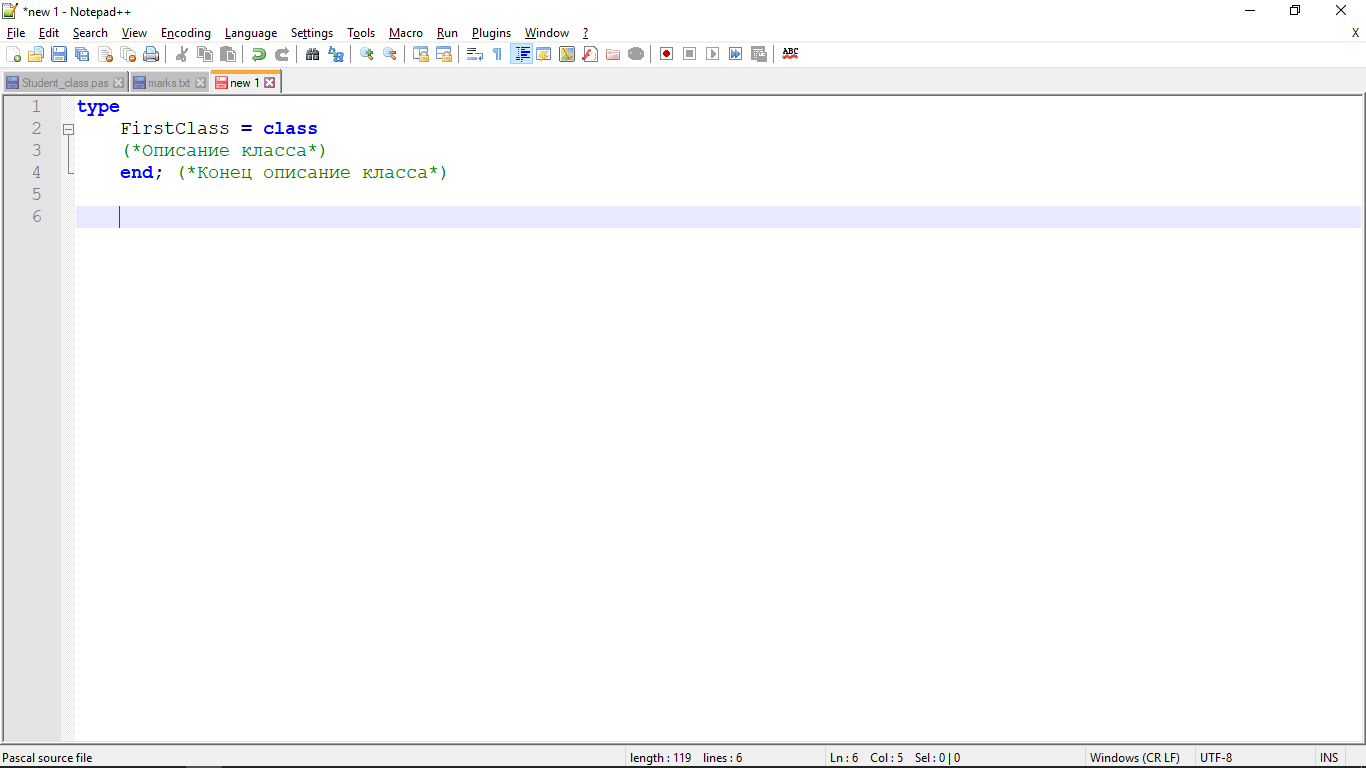


Рисунок 1

Атрибуты описываются как обычные переменные (Рисунок 2):

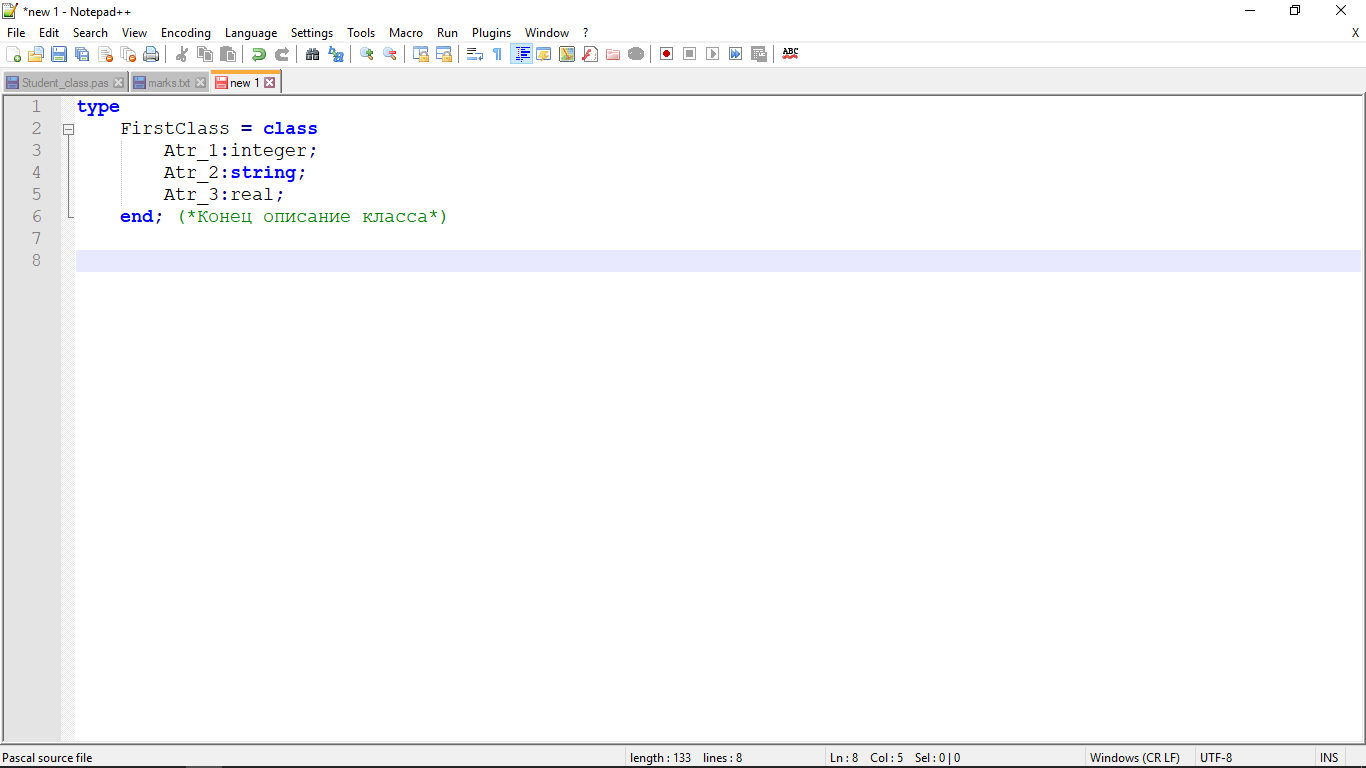


Рисунок 2

Методы описываются в секции описания класса, а его реализация задаётся после в виде “procedure\functionИмяКласса.ИмяМетода(ПередаваемыеДанные)”

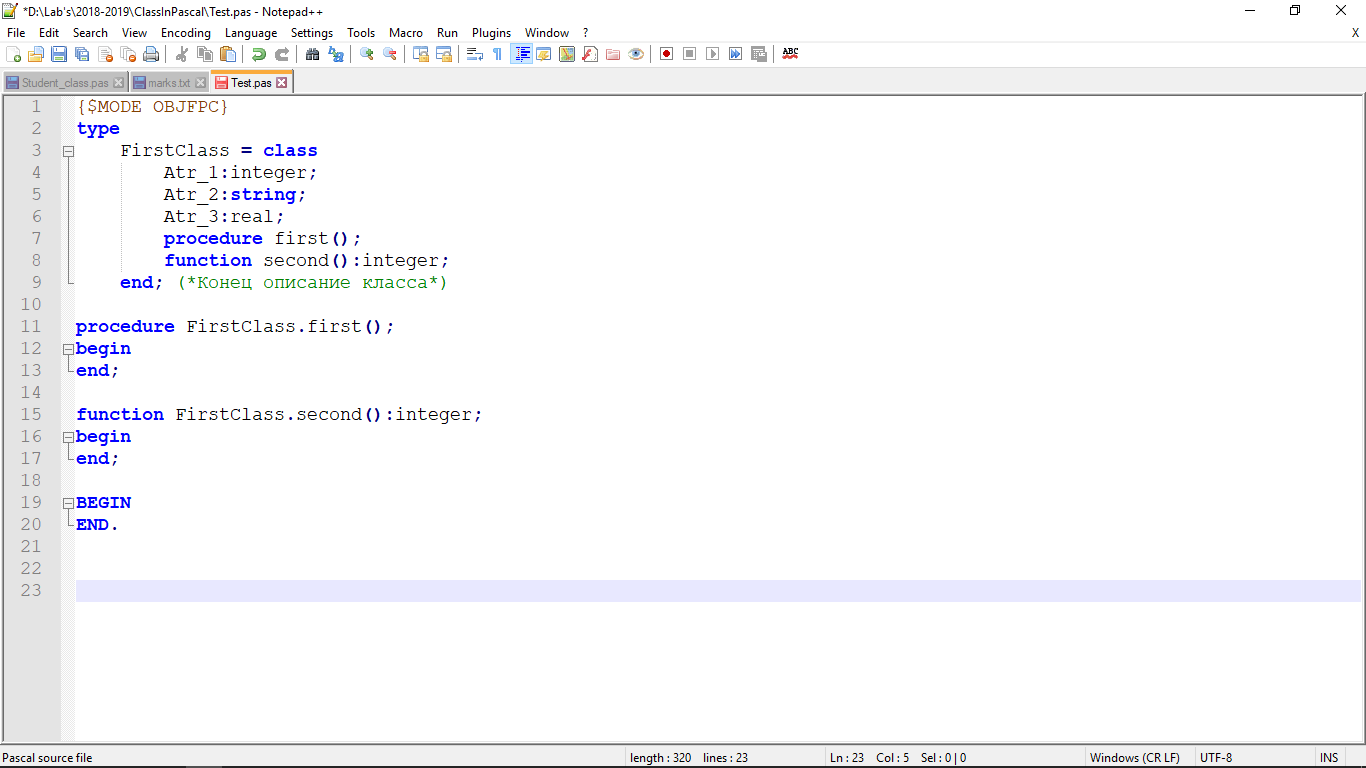


Рисунок 3

Для создания конкретного объекта в FreePascal используются специальный конструктор Create (про конструкторы и деструкторы будет сказано позже), его можно переопределить используя конструкцию constructor Create(), переопределение используется для определения атрибутов во время создания объекта.

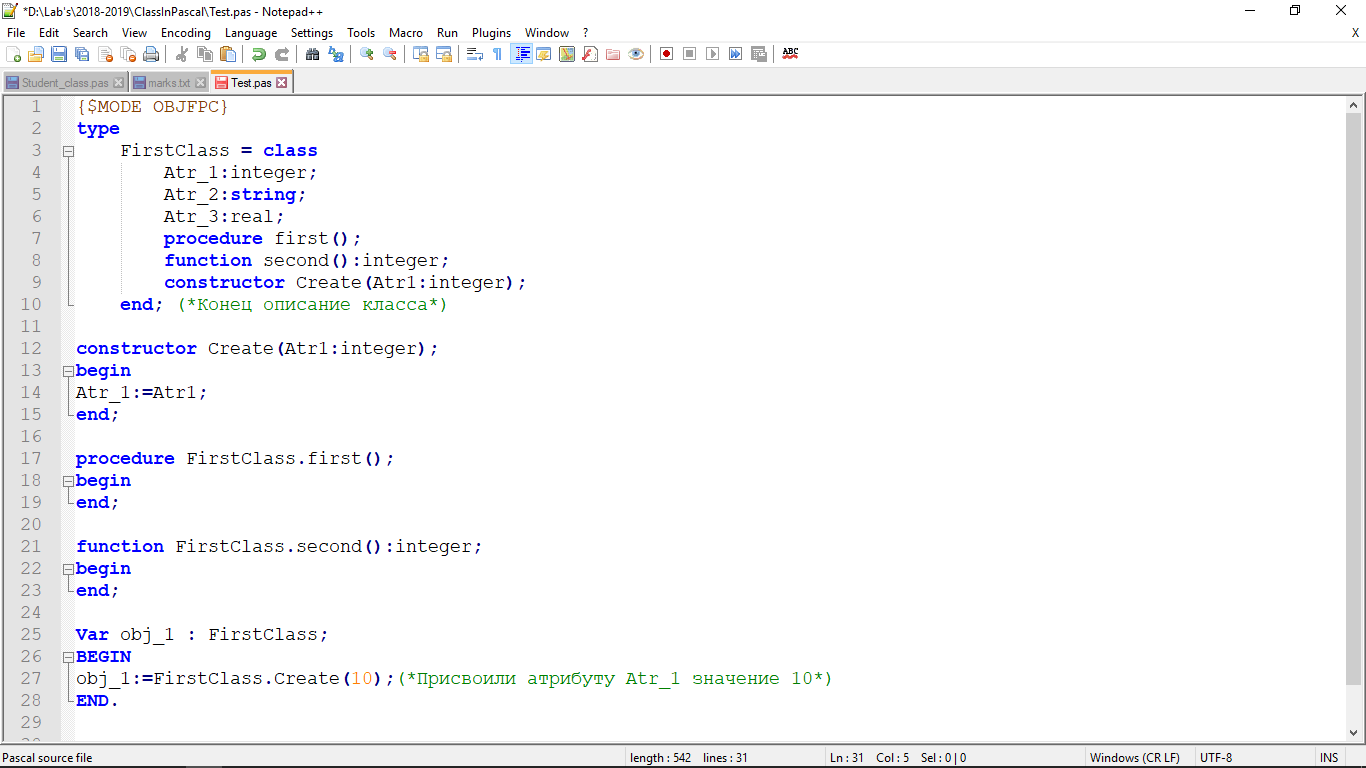


Рисунок 4

**Пример:**Допустим мы создаем программный продукт, позволяющий обрабатывать информацию о каждом студенте некоторого университета. Для этого в начале определим абстрактный класс – студент. Этот класс является общей структурой для каждого студента, в нем мы опишем: ID (уникальный номер) Студента, Имя, Фамилию, Пол, Средний балл. Каждый объект этого класса, будет являться информацией об одном студенте.  
Допустим, что информация такая как ID, Имя, Фамилию, Пол будет передаваться во время создания объекта, а информация об оценках, полученных по результатам сессии, будет храниться в отдельном текстовом файле (в файле запись вида –“ ID студента --- [оценки за сессию]”). Для того что бы вычислить средний балл для каждого студента необходимо прочитать файл, найти необходимую нам строку (содержащую ID необходимого студента), подсчитать средний балл. Именно для этой задачи будем использовать метод класса. Для класса Студент описан метода НайтиСредБалл(), который высчитывает средний балл из текстового файла. Находит оценки в файле исходя из ID студента (объекта, класса Студент). Допустим, Студент\_1 – объект класса Студент, имеющий следующие атрибуты: ID - 1, Имя – Иван, Фамилия – Иванов, Пол – М, Средний балл – “-1”(-1 – ещё не подсчитан). Для вызова метода используем конструкцию “ИмяОбъекта.ИмяМетода()” – Студент\_1.НайтиСредБалл(). Данный метод берёт информацию об ID в объекте Студент\_1, ищет в файле его оценки и находит Средний балл.

Теперь перейдём к реализации этой же задачи на FreePascal

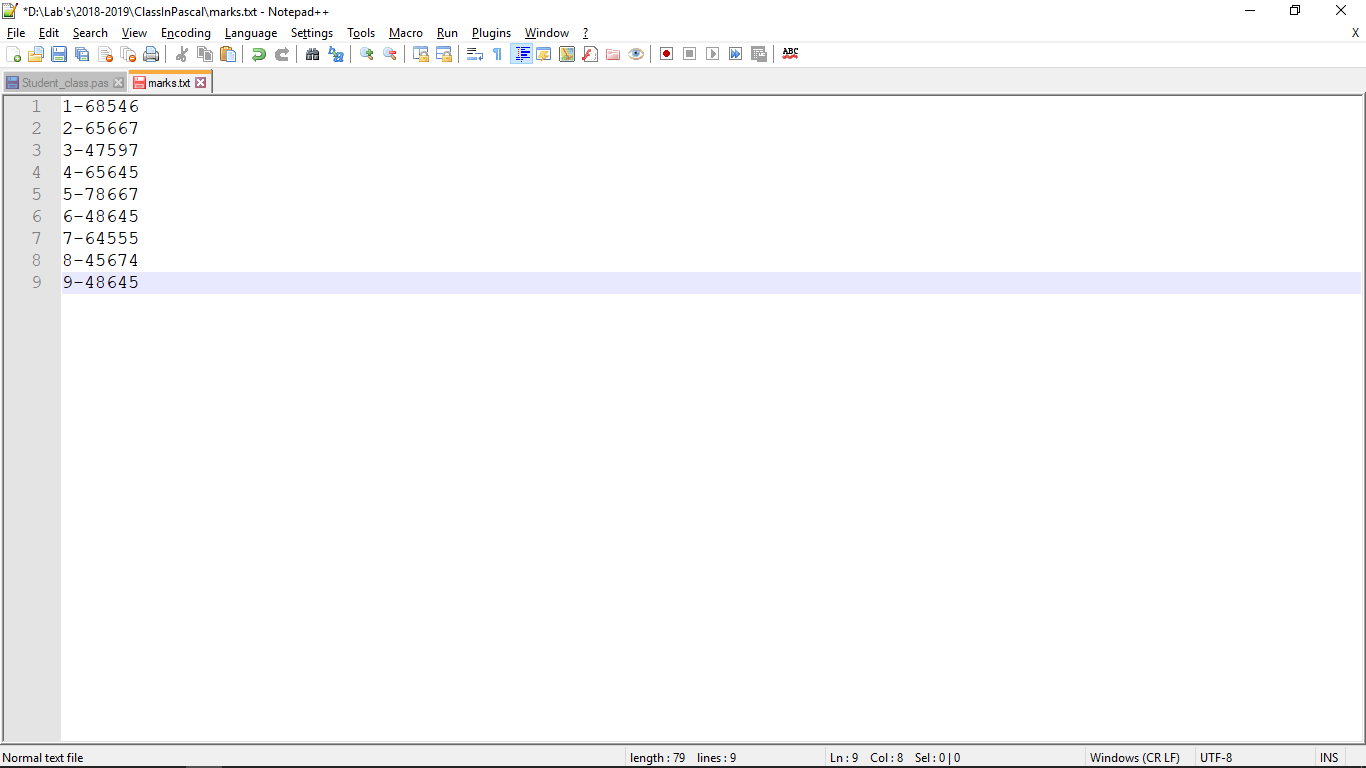


Рисунок 5

На Рис.3 представлен текстовый файл, содержащий ID студента и его оценки

Абстрактные типы данных

Место для ввода текста.

Инкапсуляция

Место для ввода текста.

Наследование

Место для ввода текста.

Полиморфизм

Место для ввода текста.

Конструкторы и деструкторы

Центральной идеей ООП является реализация понятия "абстракция".  
Смысл абстракции заключается в том, что сущность произвольной сложности  
можно рассматривать, а также производить определенные действия над ней, как над единым целым.  
Пример: Задача составления расписания занятий.  
Необходимые абстракции: студент, курс лекций, преподаватель, аудитория.  
Операции:  
− Определить студента в группу  
− Назначить аудиторию для группы  
− . . . . . . . . . .  
Одним из основных способов создания абстракции является использование  
концепции иерархической классификации. Ее суть заключается в том, что  
сложные системы разбиваются на более простые фрагменты.  
Практически все сложные системы иерархичны, и уровни их иерархии отражают различные уровни абстракции. Для каждой конкретной задачи рассматривается соответствующий уровень. Выбор низшего уровня абстракции достаточно произволен. Выбранный уровень в одном случае в качестве низшего уровня  
может оказаться уровнем достаточно высокой абстракции в другом проекте.  
Различают типовую иерархию и структурную иерархию, которые далее  
мы будем называть соответственно структурой классов и структурой объектов.

Виртуальные и статические методы

Место для ввода текста.

Литература

* Расолько, Г. А. Сборнік задач па курсу «Метады праграміравання і информатика»: дапаможнік для студэнтаў мех. -мат. фак. спец. 1-31 03 01-02 «Матэматыка (навукова-педагагічная дзейнасць)». У 2 ч. Ч. ІI. Алгарытмы апрацоўкі даных / Г. А. Расолько, Е. В. Кремень, Ю. А. Кремень, Минск 2013г. – 91с.
* Расолько, Г. А. Метады праграміравання і інфарматыка: канспект лекцый для студэнтаў мех. -мат. фак. спец. 1-31 03 01-02 «Матэматыка (навукова-педагагічная дзейнасць)». У 2 ч. Ч. ІІ. Алгарытмы апрацоўкі даных / Г. А. Расолько, Е. В. Кремень, Ю. А. Кремень, Минск 2013г. – 100с.