

# ПРИМЕНЕНИЕ МОДЕЛИРОВАНИЯ НА UML ПРИ РАЗРАБОТКЕ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ ПРОЕКТОВ РЕШЕНИЯ МАТЕМАТИЧЕСКИХ ЗАДАЧ

Широкова Ольга Александровна,  
Казанский федеральный университет  
[shirokova2602@mail.ru](mailto:shirokova2602@mail.ru)

На современном этапе развития информационных технологий совершенствуются методология и технология разработки программного обеспечения, которые, в основном, базируются на объектно-ориентированном подходе, что находит отражение в государственном образовательном стандарте высшего профессионального образования для подготовки бакалавров в области информатики и программирования.

Теоретические основы объектно-ориентированного программирования (ООП) и объектно-ориентированного проектирования представлены в работах Буча Г., Грэхема И. и др. Исходя из основных положений объектной модели, разработанной Бучем Г., под термином «объектно-ориентированное проектирование» будем понимать метод, сочетающий процесс объектно-ориентированной декомпозиции и систему обозначений для представления логической и физической, статической и динамической модели проектирования системы.

Важнейшей целью при изучении ООП является обучение студентов объектно-ориентированной декомпозиции при осуществлении поиска классов объектов, на основе которых строятся программные системы.

Обучаемые испытывают затруднения при создании объектно-ориентированного программного кода, так как объектно-ориентированные языки программирования не позволяют наглядно представить классы объектов и отношения между ними, не способствуют формированию общего представления о создаваемом программном коде в отличие от унифицированного языка моделирования (UML), являющегося средством объектно-ориентированного проектирования. Язык UML дает возможность создавать и изменять модели программной системы с помощью визуальных элементов, а также использовать специальные инструменты (CASE-средства) для автоматизированного генерирования программного кода на основе диаграмм UML, наглядно реализующих взаимосвязь ООП и объектно-ориентированного проектирования.

Таким образом, возникает необходимость совершенствования существующих методических подходов к обучению будущих учителей информатики ООП за счет использования унифицированного языка моделирования и средств автоматического генерирования объектно-ориентированного программного кода, являющихся инструментами объектно-ориентированного проектирования.

Основываясь на выявленных условиях совершенствования методики обучения ООП на основе объектно-ориентированного проектирования, а также на исследованиях Бадда Г., Барнса Д., Буча Г., Грэхема И., Коллинга М., обуславливается необходимость совместного использования унифицированного языка моделирования, являющегося средством объектно-ориентированного проектирования, и автоматизированных средств генерирования объектно-ориентированного программного кода. В работе установлено, что язык моделирования позволяет наглядно представить классы объектов и отношения между ними, которые могут быть использованы при создании объектно-ориентированного программного кода. Специальные инструменты (CASE-средства) позволяют создавать программный код на основе диаграмм языка моделирования

Возможности ООП можно эффективно использовать при реализации алгоритмов вычислительного типа. Базовыми понятиями линейной алгебры и аналитической геометрии являются вектор и матрица. Их моделью в алгоритмических языках являются массивы. Часто при решении задач аналитической геометрии и линейной алгебры необходимо использовать

массивы, размерность которых не фиксирована. В алгоритмических языках для этого удобно пользоваться динамическими массивами, их размер может определяться на этапе вычислений, а не в момент трансляции. Эти массивы относятся к ссылочным типам, требующими распределения в динамической памяти. Принцип инкапсуляции ООП позволяет соединить в описании класса воедино и элементы динамического массива, и операции над ними. Эти возможности реализует описанный ниже класс *TMas*, созданный для решения серии задач, использующих массивы различной длины. Таким же образом можно создать класс *TMatr* для динамического двумерного массива. Полное описание этих классов предусматривает достаточное количество методов, реализующих основные операции линейной алгебры и аналитической геометрии.

В работе предлагается создать проект для решения задач аналитической геометрии в пространстве и на плоскости: Отметим, что в этих задачах длина массива произвольна и указывается пользователем на этапе вычислений.

### **Создание классов *TMas* и *TMatr***

Для разработки визуального проекта решения предложенных задач создаются два модуля *Massiv* и *Matrisa* с описанием классов *TMas*, *TMatr*. В этих классах используются свойства *Elem* и *Elemm*. Описав класс в отдельном модуле, разработчик может использовать его в других программах. В модуле *Massiv* используется свойство *Property Elem*. Аналогичным образом описываются методы и свойства для второго класса *TMatr* в модуле *Matrisa*.

При работе конструктора значения полей *kMin*, *kMax*, *jMax* и *jMin*, также как и в классе *TMas*, в *TMatr* передаются как параметры, определяющие количество элементов массива. Метод *ElemPP* позволяет интерпретировать байты памяти, отведенные под элементы с номерами *k* и *j*, как значение вещественного типа. Он записывается в секции *Protected*, поэтому доступен потомкам класса *TMatr*.

### **Отношение композиции между классами *TMas*, *TMatr* и *TForm***

Разработка визуального проекта начинается с планирования и структурирования, т.е. создания проекта системы. В мире программного обеспечения для этого служат модели. Модель – это абстракция, описывающая суть сложной проблемы или структуры без акцента на несущественных деталях, тем самым делая ее более понятной. Модели помогают организовывать, отображать, понимать и создавать сложные проекты.

Композицией называют такое отношение между классами, когда один является неотъемлемой частью второго. Физическая композиция реализуется включением в класс фиксированного количества полей, являющихся объектами другого класса. Обычно такие поля принято называть объектными.

На диаграмме классов композиция изображается линией с закрашенным ромбом, указывающим на класс большей сложности, в который происходит включение объектных полей. Между классами *TMas* и *TMatr* существует отношение композиции. Представим диаграмму классов приложения решения задач с использованием классов *TMas* и *TMatr* (рис. 1).

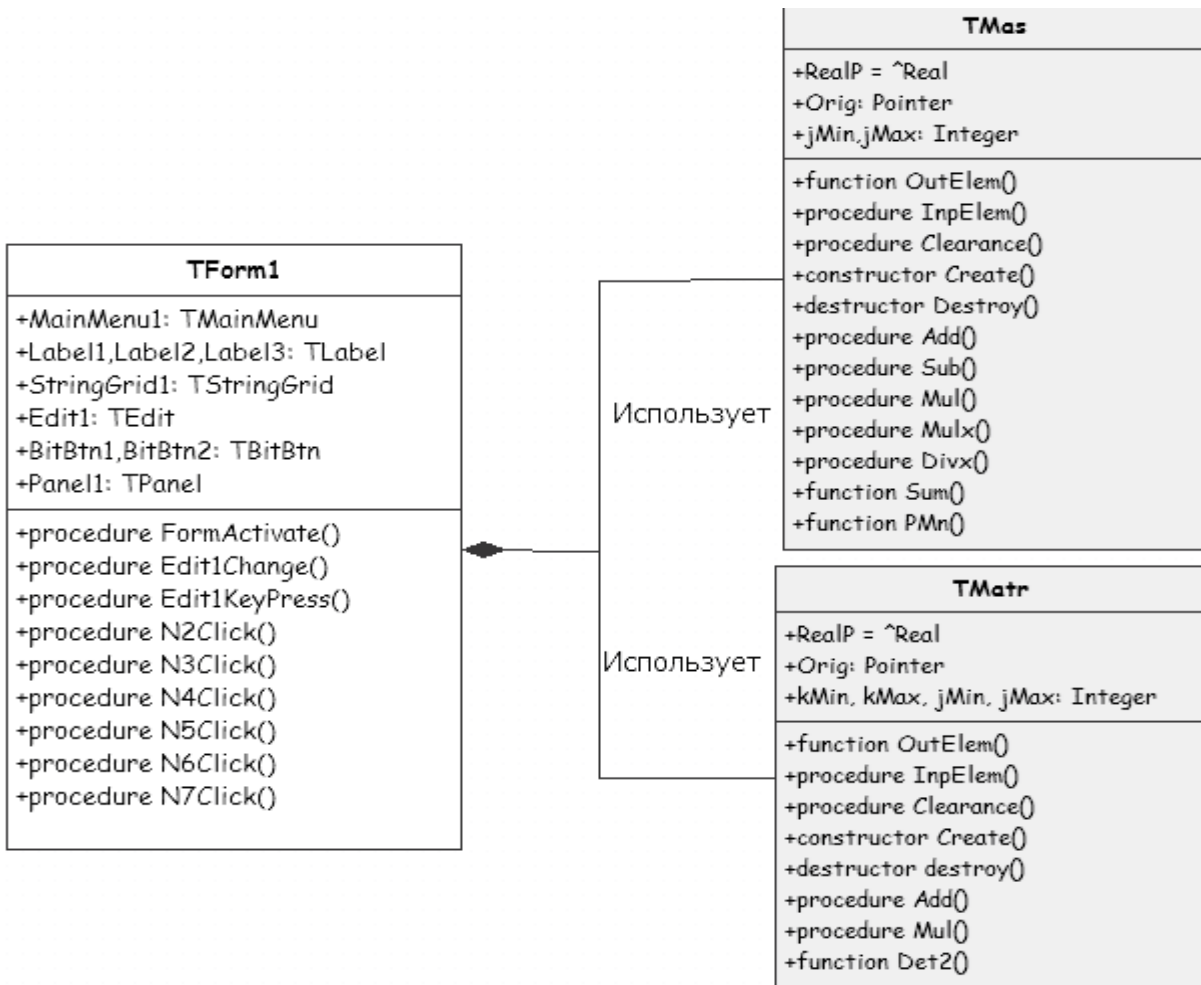


Рис. 1. Диаграмма классов приложения с использованием классов TMatr и TMas

Интерфейс проекта создается с помощью меню, размещенного на форме. Меню диалогового окна позволяет решать задачи аналитической геометрии в пространстве и на плоскости. В приложениях Windows, как правило, реализуется интерфейс со свободной навигацией, который должен обеспечивать пользователю возможность выполнения предусмотренных действий во всех допустимых последовательностях. В процессе визуального проектирования интерфейса мы получаем заготовку модуля с полями-компонентами. Меню диалогового окна позволяет решать поставленные задачи (Рис.2а, б).

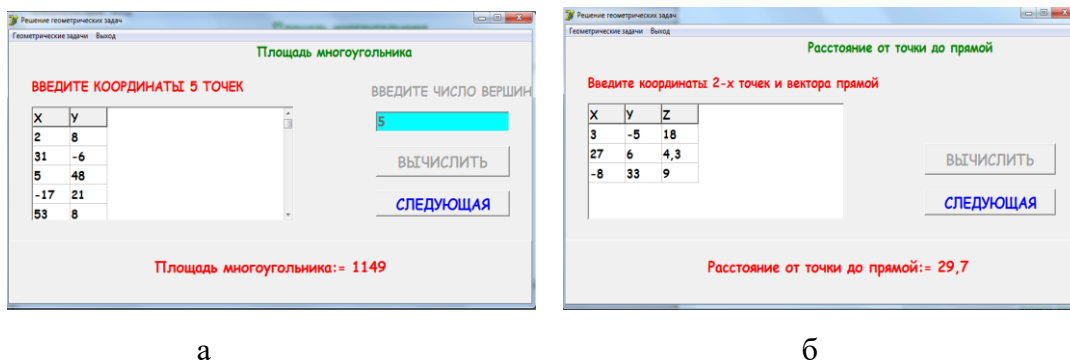


Рис. 2

Таким образом, создание проектов решения математических задач способствует формированию навыков объектно-ориентированного и визуального программирования моделей реальных объектов и структур.