

ГОУ ВПО «МАТИ» - Российский государственный  
технологический университет имени К. Э. Циолковского

Кафедра «Моделирование систем и информационные технологии»

# ДИПЛОМНАЯ РАБОТА

Разработка языка программирования  
математических расчетов FracCalc на  
основе дробей большой длины

Дипломник: Сеницин М.В.

Руководитель: Лидовский В.В.

# Задачи дипломного проектирования

В процессе дипломного проектирования ставятся задачи следующего содержания:

- Создание класса для хранения неправильной дроби.
- Разработка синтаксиса языка.
- Построение интерпретатора.
- Создание среды разработки.

# Представления чисел

При записи дроби числитель разделяется со знаменателем символом «\_», а целая часть дроби отделяется знаком точки от числителя и знаменателя.

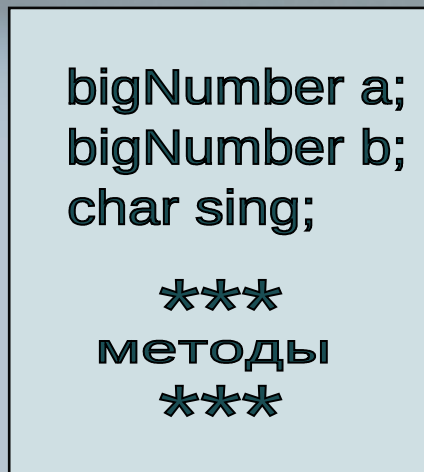
*<целая часть>.<числитель>\_<знаменатель>*

- Числа с десятичной точкой      **1.5**
- Правильные дроби                      **1.1\_2**
- Неправильные дроби                      **3\_2**

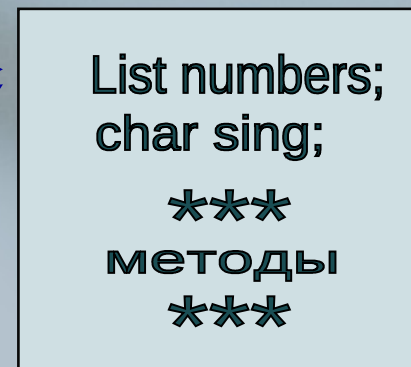
# Класс *Fraction*

Для хранения неправильной дроби в памяти компьютера был разработан класс fraction, структура которого изображена ниже:

**fraction**



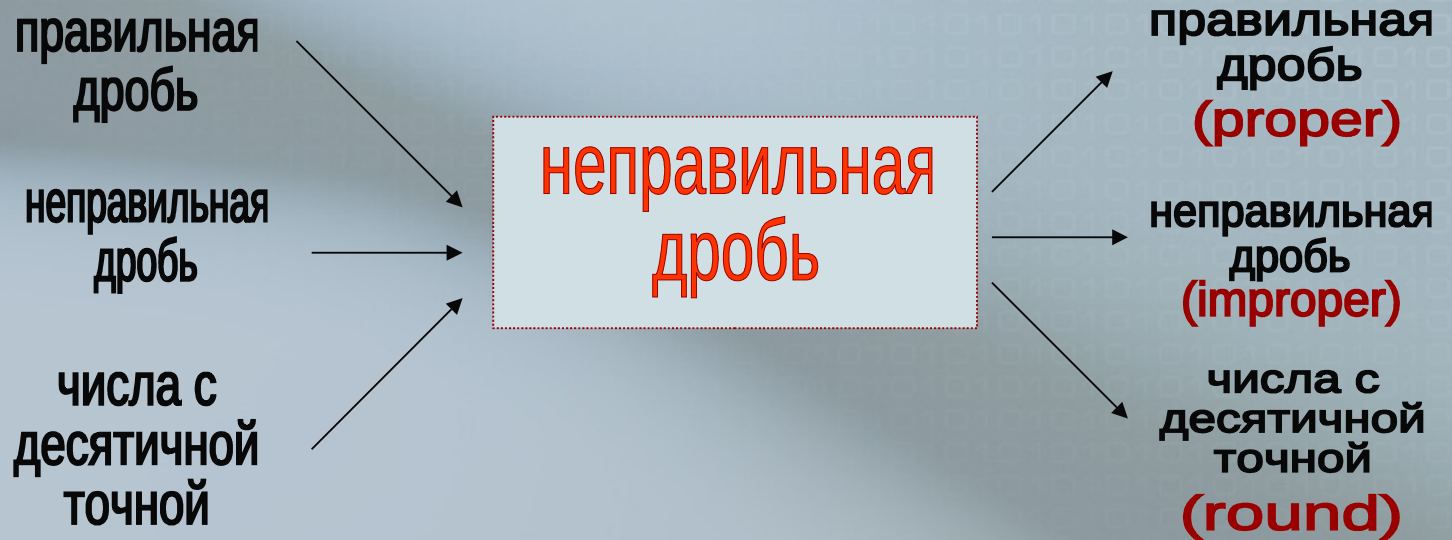
**bigNumber**



**разряды**

# Формат хранения чисел

Числа могут вводиться в любом удобном для пользователя формате, из которых число всегда преобразуется в неправильную дробь.



Формат вывода значений на экран переключается с помощью соответствующих команд.

# Грамматика языка программирования

## FracCalc

### Алфавит

Константы    Идентификаторы    Разделители    Операторы

имена  
переменных

имена  
функций

служебные  
слова

### Инструкции

Управляющие инструкции

Декларативные конструкции

команды

вызов  
функции

описание  
функции

# Операции

- **Арифметические:** сложение(+),  
вычитание(-)  
и деление(/), умножение(\*)  
и возведение в степень (^);
- **Операции сравнения:** <, <=, ==, !=, >=, >;
- **Логические операции:** && (and) и || (or);
- **Оператор запятая (следования):** ','  
A = (5.1\_2, B = 3\_5, 7\_6)  
> 7\_6

# Управляющие инструкции

- Инструкции выбора
  - ❖ Условный оператор *if*
- Инструкции итерации
  - ❖ цикл с условием *while*
  - ❖ цикл с параметрами *for*
- Инструкции перехода
  - ❖ *break*
  - ❖ *continue*



# Пользовательские функции

Формат описания функции следующий:

```
Имя_функции (список аргументов) {  
    Инструкция  
}
```

После имени функции в скобках указывается список параметров в виде списка имен переменных через запятую, если они необходимы.

Переопределение функции автоматически удаляет старое. Все переменные, используемые в функции и ее параметры, являются локальными. Для возвращения результата в главную программу, используется оператор:

```
return выражение;
```

# Управляющие команды

- Команды переключения в режим вывода значение:
  - ❖ **proper** – правильная дробь
  - ❖ **improper** – неправильная дробь
  - ❖ **round n** – вещественный вид (по умолчанию 5 знаков после запятой)
- Команды загрузки файла
  - include <имя файла>**
- Команда выхода из программы
  - exit**

# Разработка интерпретатора

В процессе дипломного проектирования было разработано две реализации проекта:

- На языке C++ с использованием программы YACC для построения синтаксического анализатора контекстно-свободного языка;
- На языке Java с использованием библиотеки графических компонент Swing

# Реализация интерпретатора на языке C++

При данной реализации проекта использовалась программа YACC (Yet Another Compiler Compiler), предназначенная для построения синтаксического анализатора контекстно-свободного языка. Анализируемый язык описывается с помощью грамматики в виде, близком форме Бэкуса-Наура (НФБН).

Программа представляет собой консольное приложение интерактивного режима работы. Интерпретатор работает под управлением ОС Linux и Microsoft Windows.

# Реализация интерпретатора на языке Java

Синтаксический разбор текста программы осуществляется в три этапа:

- разбиение текста программы на законченные инструкции;
- инструкция проходит через массив шаблонов всевозможных инструкций в языке в итоге чего определяется ее тип;
- все выражения в инструкциях преобразуются в польскую обратную запись и вычисляются.

# Разработка графической среды

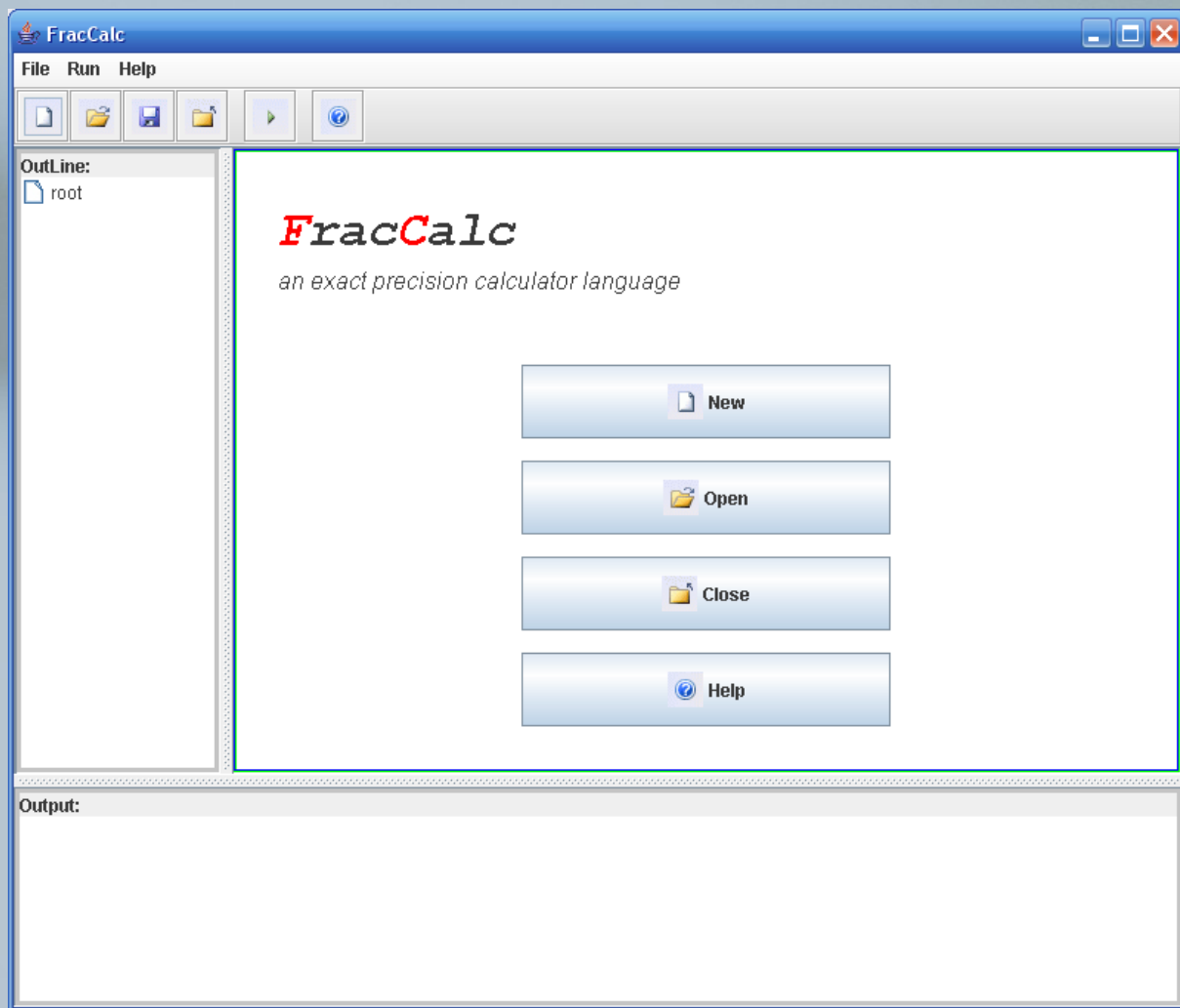
Для построения графического интерфейса была выбрана библиотека графических компонентов Swing, входящая в стандартную библиотеку классов Java.

Swing был разработан компанией [Sun Microsystems](#).

Библиотека Swing имеет ряд преимуществ:

- богатый набор интерфейсных примитивов
- настраиваемый внешний вид на различных платформах (look and feel)
- раздельная архитектура модель-вид (model-view)
- встроенная поддержка HTML

# Графическая среда разработки



**СПАСИБ**

**О**

**ЗА**

**ВНИМАНИЕ**