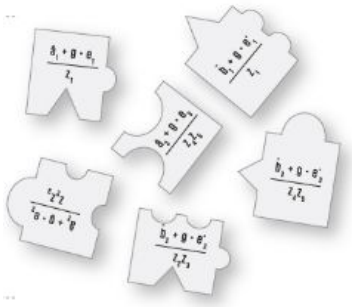
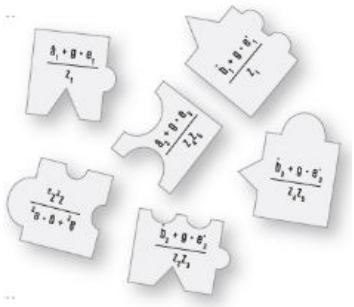


ДИПЛОМНАЯ РАБОТА МАГИСТРА

ОБОСНОВАНИЕ МЕТОДА ЗАЩИТЫ ПРОГРАММНОГО КОДА НА ОСНОВЕ ПРАВИЛЬНЫХ СКОБОЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Выполнил Богуцкий Д.А.

Руководитель работы проф. Алексеев М.А.



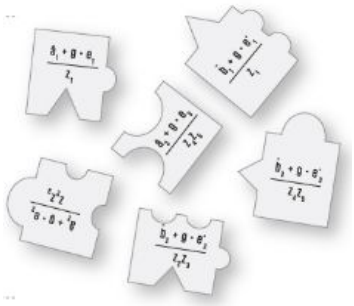
ЦЕЛЬ РАБОТЫ

Объект исследования: методы маскирования информации о используемых в программах расчетных выражений .

Цель магистерской работы: исследование и разработка метода маскирования программного кода расчетных зависимостей с использованием правильных скобочных последовательностей

В работе рассматриваются основные принципы маскирования программного кода и предлагается новая методика выбора маскирующего преобразования выражений ,описывающих определенные расчетные зависимости

Для достижения поставленной цели в работе использовались элементы теории компиляторов, методы теории информации и комбинаторики.



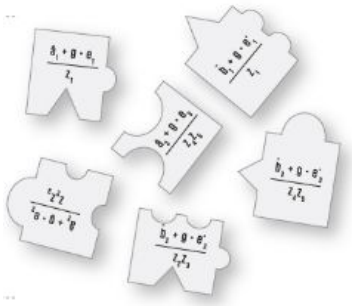
АКТУАЛЬНОСТЬ РАБОТЫ

Разработка наиболее эффективного метода защиты программного продукта становиться одной из важных задач программистов, занимающихся разработкой специализированного платного программного обеспечения

Надёжная защита кода нужна в тех случаях, когда код содержит важные торговые секреты фирмы, уникальные алгоритмы или расчетные зависимости.

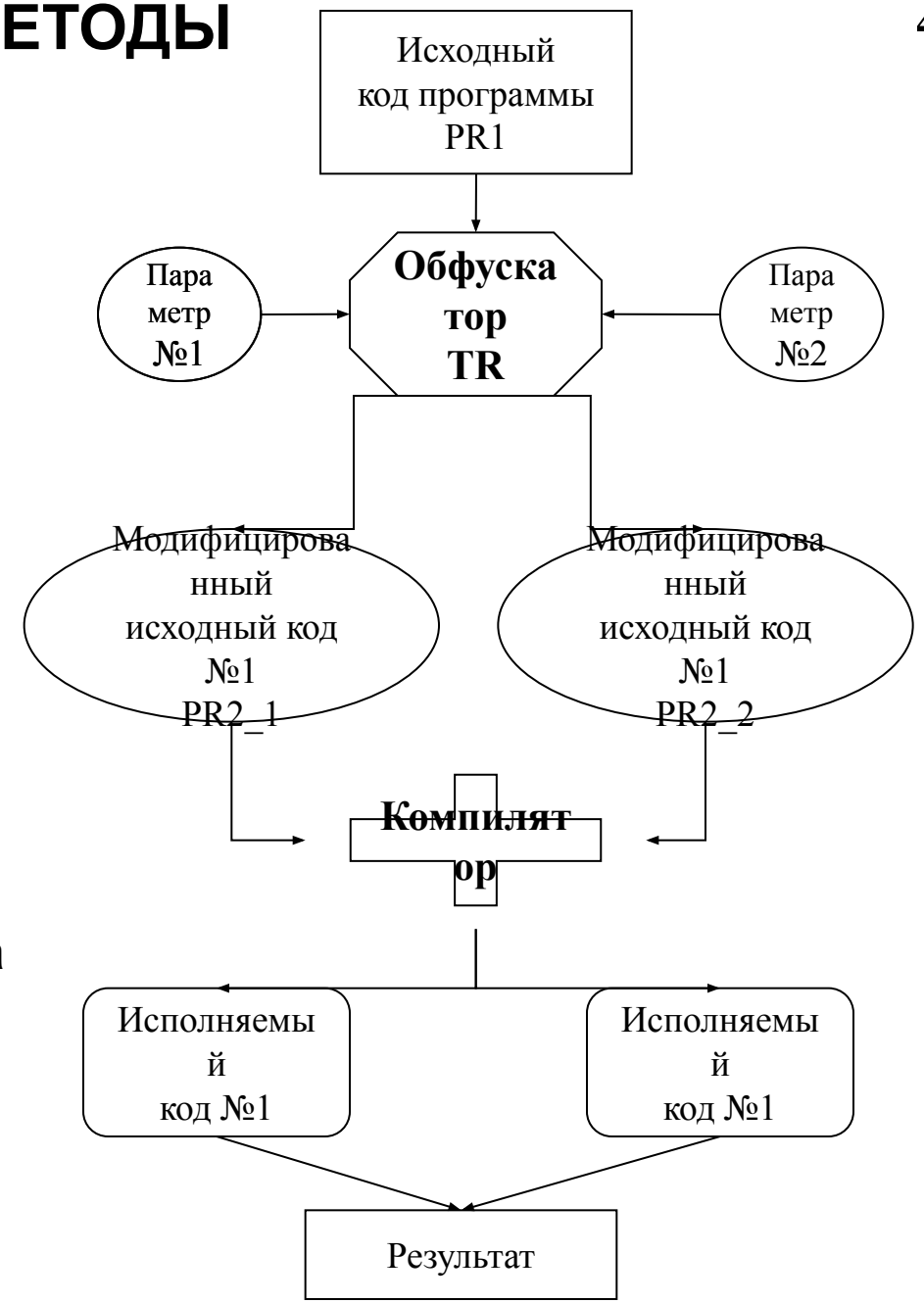
Обфускация (запутывание)- один из методов защиты программного кода, который позволяет усложнить процесс реверсивной инженерии кода защищаемого программного продукта .

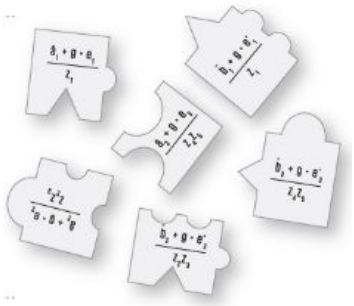
Суть процесса обфускации заключается в том, чтобы запутать программный код и устранить большинство логических связей в нем, то есть трансформировать его так, чтобы он был очень труден для изучения и модификации посторонними лицами



ПРОЦЕСС И МЕТОДЫ ОБФУСКАЦИИ

- форматирования, которые изменяют только внешний вид программы;
- преобразования структур данных, изменяющие структуры данных, с которыми работает программа;
- преобразования потока управления программы, которые изменяют структуру её графа потока управления.





ОСНОВНАЯ ИДЕЯ ПРЕДЛАГАЕМОГО МЕТОДА

Суть метода заключается в усложнении представления математических выражений в коде программы, за счет “вставки” в них дополнительного набора парных скобок, не изменяющих нужный порядок вычислений.

1. *Пусть в алгоритме программы предусмотрена необходимость выполнения следующего выражения*

X =A+B*(C*sin(D)-E*cos(D)) **(1)**

которое при ранее заданных значениях A,B,C,D и E дает значение **X=X1**

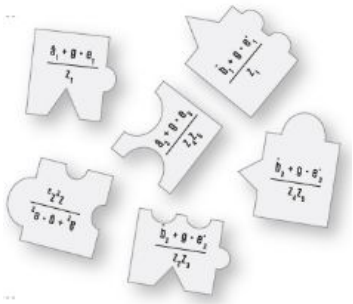
2. *В исходном коде программы записывается выражение*

X =A+B*C*sin(D)-E*cos(D) **(2)** **X=X2 (X1 ≠ X2)**

3. *Пусть в* программе имеется некая функция ,которая по какому то алгоритму вставляет в нужное место выражения (2) правильную скобочную последовательность **(( ))** таким образом, чтобы выражение (2) приобрело вид

X =A+(B*(C*sin(D)-E*cos(D))) **(3)** **X=X3 (X1 = X3)**

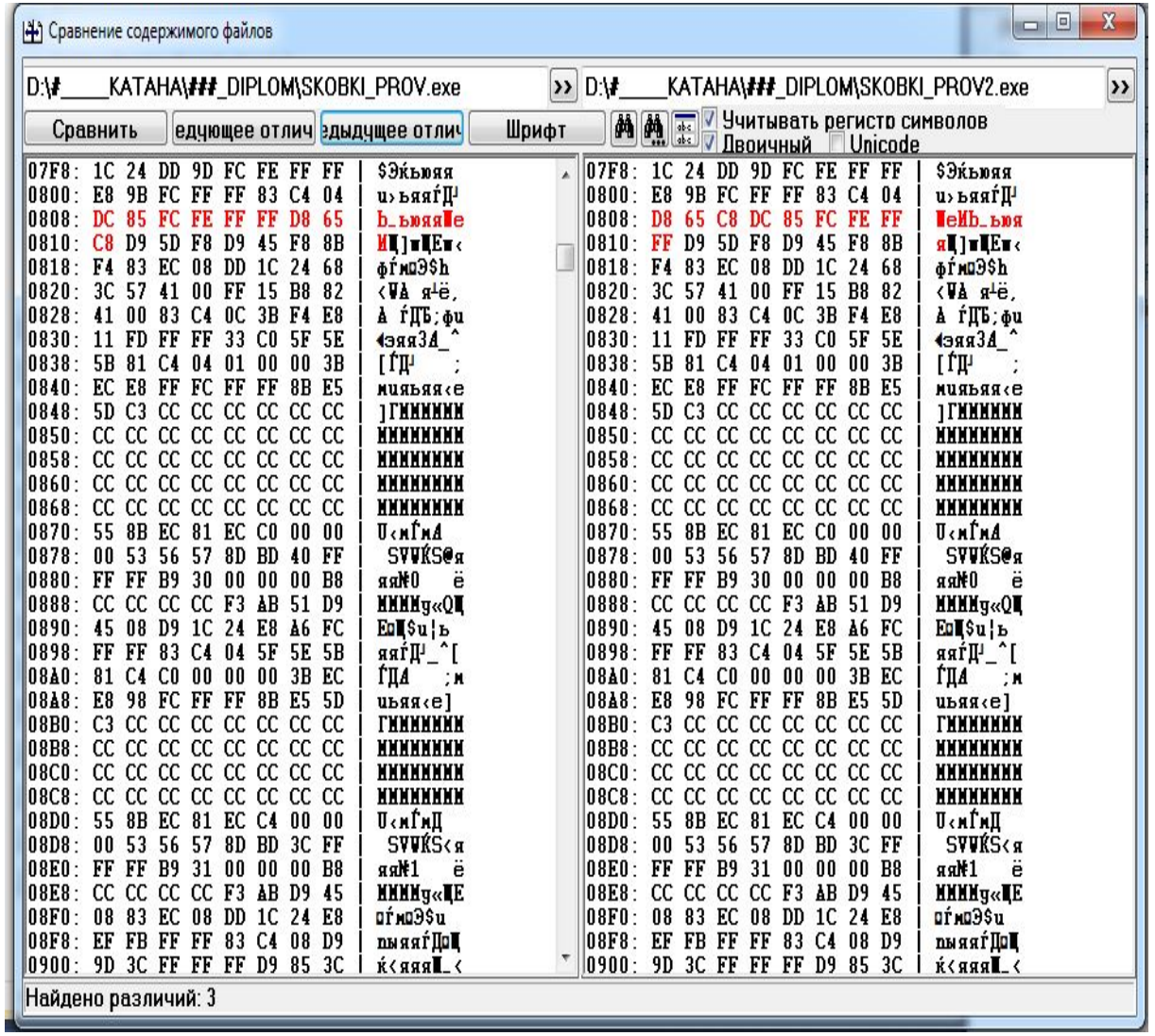
Такое преобразование по не только восстанавливает порядок вычисления, но и добавляет в исходную зависимость дополнительную парную скобку, не изменяющую порядок вычислений.

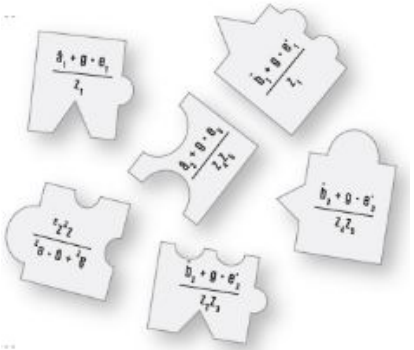


ВЛИЯНИЯ КОМПИЛЯТОРА

```
 {
  float x,a,b,c,d;
   b=5.5;  c=3.3;  d=10;
   a=b+c+d;
   x=a*a+b*c+sin(c)-d;
   printf("%f",x);
        return 0;   }
```

```
{
 float x,a,b,c,d;
  b=5.5;  c=3.3;  d=10;
  a=((((b)+(c)+(d))));
  x=(((a*a)+(b*c)+(sin(c)-d)));
  printf("%f",x);
       return 0;   }
```





ПРИМЕНЕНИЕ МЕТОДА

1
АЛГОРИТМ
Формула
X =A+B*(C*sin(D)-E*cos(D))

2
ИСХОДНЫЙ КОД
Строка **S**
" X =A+B*C*sin(D)-E*cos(D) "
Строка **P**
" . . . . . "

Обфускатор

3
ИСХОДНЫЙ КОД
Строка **P**
"(( )) "
Скобочная последовательность

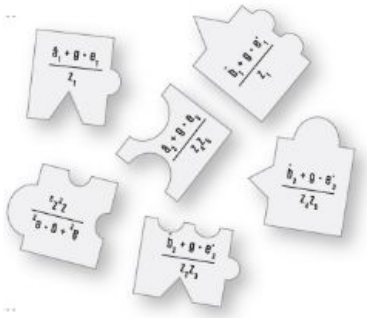
Компилятор

КБ

4
ИСХОДНЫЙ КОД
Строка
Функция наложения
"X =A+(B*(C*sin(D)-E*cos(D)))"

5
ИСХОДНЫЙ КОД
Функция синтаксического анализа строки (парсер)

6
Результат



ПРАВИЛЬНЫЕ СКОБОЧНЫЕ ПОСЛЕДОВАТЕЛЬНОСТИ

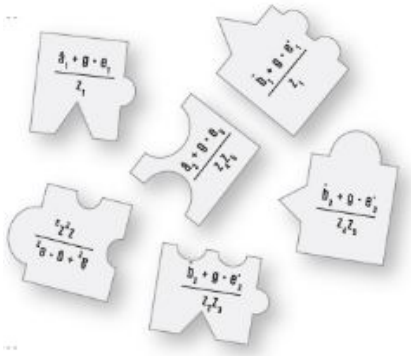
Правильная скобочная последовательность (ПСП) — символьная последовательность, составленная в алфавите, состоящем из символов, сгруппированных в упорядоченные пары (типы скобок «(» и «)», «[» и «]», и т. п.), удовлетворяющая определённым правилам, обеспечивающим последовательную вложенность подпоследовательностей, обрамлённых открытой и закрытой скобкой одного типа.

Количество ПСП для заданного количества пар скобок одного типа определяется числом Каталана

C n = (2n)! / (n! (n+1)!)

где n - количества пар скобок одного типа

n	Cn
2	2
3	5
4	14
5	42
…..	…….
18	477638700
…..	….
29	1,002E+15
30	3,815E+15

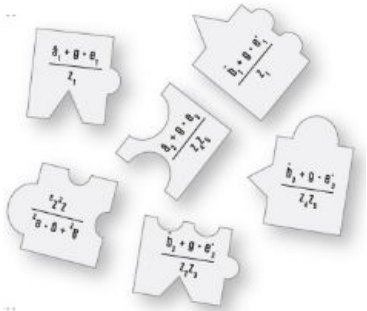


СИНТАКСИЧЕСКИЙ АНАЛИЗ СТРОКИ

Синтаксический анализ (парсинг) — процесс сопоставления линейной последовательности слов(токенов) языка программирования с его формальной грамматикой.

Синтаксический анализ - это основа всех компиляторов и интерпретаторов с языков высокого уровня.

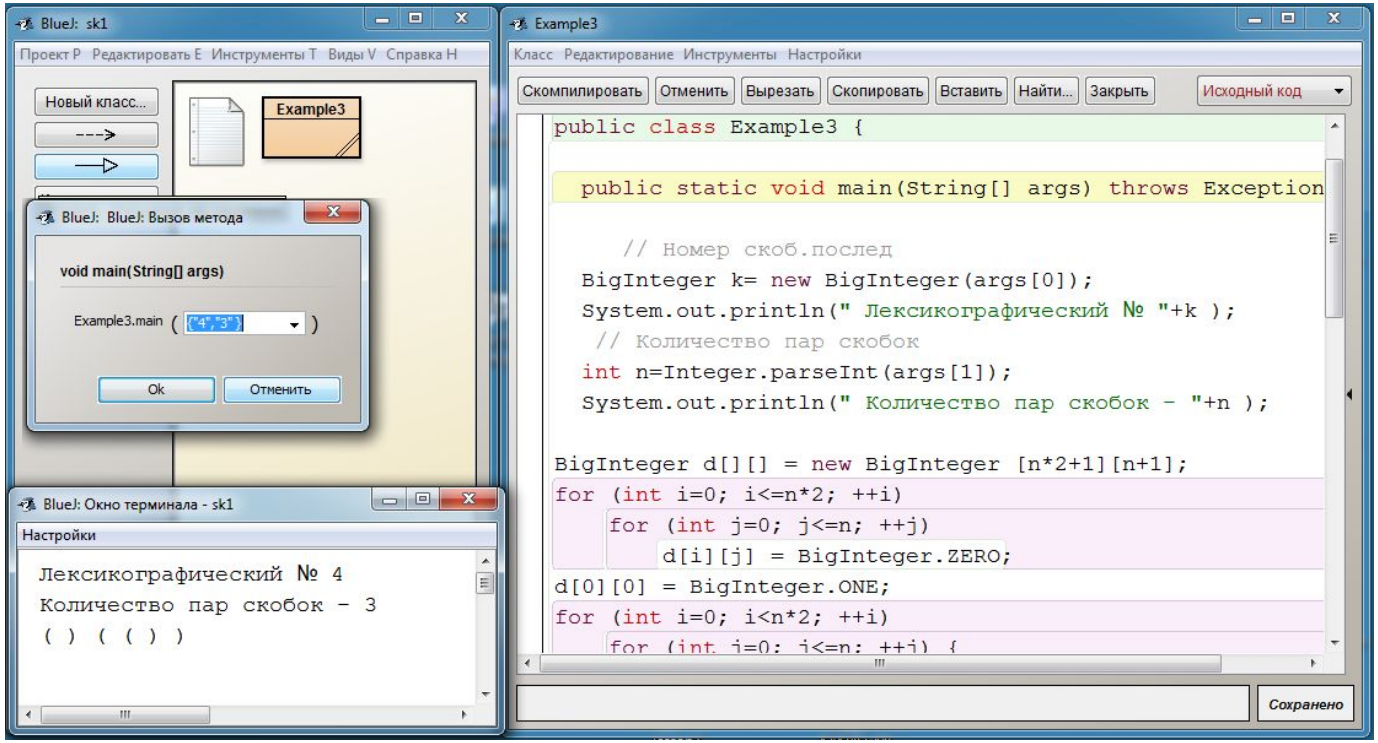
В ходе синтаксического анализа исходный текст преобразуется в структуру данных, обычно — в дерево, которое отражает синтаксическую структуру входной последовательности и хорошо подходит для дальнейшей обработки



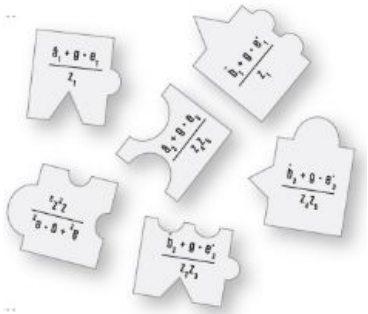
РЕАЛИЗАЦИЯ ФУНКЦИЙ ГЕНЕРАЦИИ ПРАВИЛЬНОЙ СКОБОЧНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ

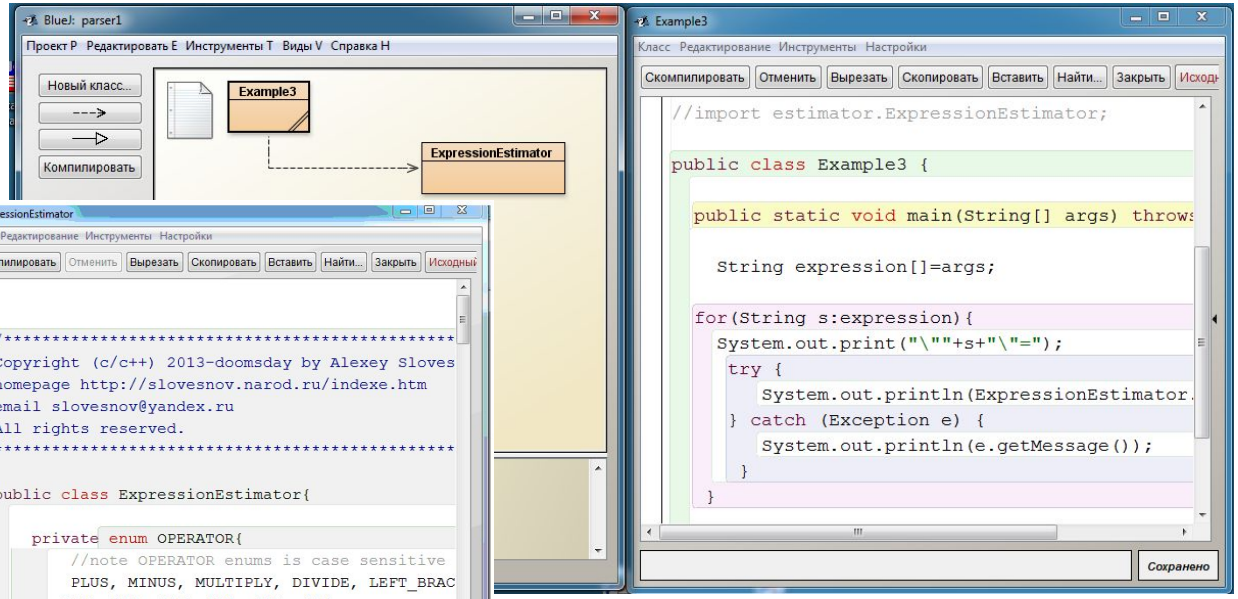
Программная реализация была осуществлена на языке JAVA в среде BlueJ.

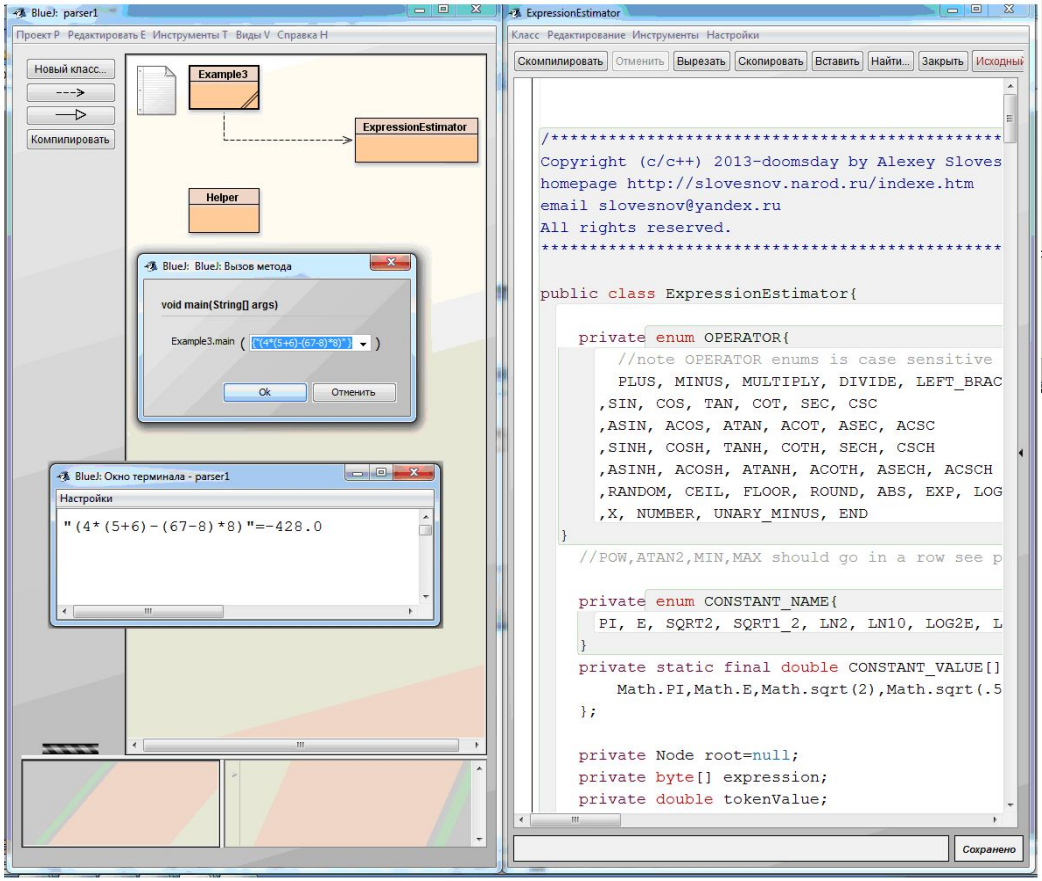
BlueJ — интерактивная среда разработки на языке Java, подходящая для разработки небольших программ.



РЕАЛИЗАЦИЯ СИНТАКСИЧЕСКОГО АНАЛИЗАТОРА СТРОКИ







ЗНАЧЕНИЕ РАБОТЫ

- **Научная новизна** состоит в разработке метода запутывания исходного кода программы, основанном на применении правильных скобочных последовательностей , и позволяющим скрыть порядок действий в расчетных зависимостях программы при обфускации.
- **Практическое значение работы** заключается в разработке алгоритмов и программных модулей генерации правильной скобочной последовательности и синтаксического анализатора строк
- **Область применения**. Разработанная методика запутывания исходного кода может применяться для затруднения декомпиляции/отладки, изучения программ с целью обнаружения функциональности и обхода систем проверки лицензий.

Дальнейшее развитие работы

Дальнейшим развитием исследований в этом направлении целесообразно проводить в направлении теоретического обоснования предложенного метода и разработки эвристических алгоритмов наложения строкового представления расчетных зависимостей и правильных скобочных последовательностей

Основные результаты магистерской работы

1. Проведен анализ возможных атак на программное обеспечение.

2. Проведен анализ существующих методов запутывания программного кода и принципов построения специальных программ-обфускаторов.

3. Предложен метод запутывания вычислительных выражений, применяемых в программе, и основанный на применении правильных скобочных последовательностей.

4. Предложена схема написания программного кода ,предусматривающая возможность применения вышеуказанного метода.

5. Разработаны программные модули ,обеспечивающие реализацию метода защиты программного кода на основе правильных скобочных последовательностей.

6. Сформулирована комбинаторная задача, решение которой позволит провести теоретически оценку предлагаемого метода запутывающего преобразования.