

# Доработка языка программирования Freepascal: реализация замыканий.

Выполнил студент группы с8503а

Кевролетин Василий Владимирович

Руководитель: старший преподаватель кафедры  
информатики, математического и компьютерного  
моделирования Кленин Александр Сергеевич

## Технические особенности проекта

- Поддержка большого числа процессоров.
- Поддержка большого числа операционных систем.
- Поддержка нескольких диалектов Pascal.

## Организационные особенности проекта

- Открытый исходный код.
- Разрабатывается постоянной командой добровольцев.
- Принимает доработки от сторонних разработчиков.

# Этапы компиляции: часть 1

## Лексический анализ

поток символов → поток лексем

## Синтаксический анализ

поток символов → АСТ, таблицы символов

## Семантический анализ

проверка соответствия типов

## Высокоуровневая оптимизация

## Генерация кода

АСТ → машинный код\*

## Низкоуровневая оптимизация

машинный код → машинный код

## Компиляция модуля

SomeUnit.pas

- SomeUnit.ppu - таблицы символов
- SomeUnit.o - объектный код

# Анонимной функции

## Пример

```
1 procedure Factory: TProc;  
2 begin  
3     Result := procedure  
4         begin  
5             Writeln;  
6         end;  
7 end;
```

# Вложенной функции

## Пример

```
1 procedure outer;  
2   var i: Integer;  
3  
4     procedure inner; begin  
5       i := 10;  
6     end;  
7  
8   begin  
9     ...  
10  end;
```

# Замыкание

## Пример. Продление жизни локальных переменных.

```
1      procedure Factory(data: Integer): TProc;  
2      begin  
3          Result := procedure  
4              begin  
5                  Writeln( data );  
6              end;  
7      end;  
8  
9      var f1: TProc;  
10     begin  
11         f1 := Factory(10);  
12         f2 := Factory(20);  
13         f1();           { 10 }  
14         f2();           { 20 }  
15     end;
```

## Пример

```
1  var i: Integer;  
2      f: TProc;  
3  begin  
4      i := 0;  
5      f := procedure  
6          begin  
7              Writeln(i);  
8          end;  
9      i := 10;  
10     f();           { 10 }  
11 end;
```



## Пример

```
1 int main()
2 {
3     int i = 0;
4     auto f = [=] { std::cout << i; };
5     i = 10;
6     f();           /* 0 */
7 }
```

# Анонимные функции без замыканий

## Пример

```
1  std::function<void(void)> factory(int data) {
2      return [&data] { std::cout << data << "\n"; };
3  }
4
5  int main()
6  {
7      auto f1 = factory(10);
8      auto f2 = factory(20);
9      f1();                               /* 20 */
10     f2();                               /* 20 */
11 }
```

g++ version 4.7.2,

command-lin g++ -std=gnu++0x main.cpp

# Реализация замыканий в современных ЯП

ЯП	анонимные функции	вложенные функции	захват по значению	захват по ссылке	замыкания
Perl	+	+/-		+	+
Python	+	+		+	+
Ruby	+	+		+	+
Scheme	+	+		+	+
Elisp	+	+		+	
C					
C++	+		+	+	
Delphi	+	+		+	+
fpc		+			

# Анонимные методы Delphi

## Пример

```
1  type TProc = reference to procedure;  
2  
3  var  
4      p : TProc;  
5      i : Integer;  
6  
7  begin  
8      i := 10;  
9      p := procedure begin  
10         Writeln(i)  
11     end;  
12     p();  
13 end.
```

## Объявление нового класса

```
1  type
2      TFrameObject = class (TInterfacedObject)
3          i : Integer;
4          procedure Proc;
5      end;
6
7  procedure TFrameObject.Proc;
8  begin
9      Writeln(Self.i);
10 end;
```

## Создание объекта

```
1  var
2    p : procedure of object;
3    frameObj: TFrameObject;
4
5  begin
6    frameObj := TFrameObject.Create;
7
8    frameObj.i := 10;
9    p := @frameObj.Proc;
10   p();
11  end.
```

## Проделанная работа

- Изучена предметная область.
- Изучено внутреннее устройство компилятора fpc.
- Получено согласие разработчиков на совместимую с Delphi реализацию.

## План реализации

- Совместимая с Delphi реализация.
- Возможность захвата переменных по значению.
- Согласование и реализация упрощенного синтаксиса.