



XIV конференция

Свободное программное обеспечение

в высшей школе

Переяславль, 25-27 января 2019

Обучающая среда по программированию на базе СПО

Валерий Лаптев

Константин Обухов

Игорь Мурзаев



Астраханский

государственный технический университет

Проблемы подготовки ИТ - кадров

- ЕГЭ - 2009 год
- на ИТ-специальности стали поступать студенты, которые не изучали программирование в школе
- Возникла задача - обучение программированию с нуля.
- В результате анализа средств обучения принято решение
 - разработать учебный язык программирования
 - реализовать среду для начального обучения программированию на этом языке.

Обучающая система



Обучающая система

IDE - инструментальная подсистема

- Предоставляет информацию
- Обеспечивает решение задач

Управляющая подсистема

- Предоставляет информацию
- Оценивает результат выполнения задания
- Оценивает общий уровень обученности
- Управляет процессом обучения

Платформа преподавателя

- Обеспечивает подготовку учебного курса
- Обеспечивает настройку процесса обучения

Учебный язык программирования

Семантические концепции:

- **понятия** учебного языка должны соответствовать понятиям промышленных императивных ЯП;
- **множество** понятий должно быть минимально;
- **конструкции** учебного языка не должны зависеть ни от аппаратной платформы, ни от ОС;
- **конструкции** языка должны удовлетворять принципам структурного программирования;
- **язык должен поддерживать** модульность, процедурное программирование и объектно-ориентированное программирование;
- **намерения** программиста всегда должны указываться явно (запрет умолчаний).

Учебный язык программирования

Синтаксические концепции:

- базовая лексика языка должна быть русскоязычной;
- ключевые слова должны иметь английский эквивалент;
- каждый оператор языка начинается ключевым словом;
- каждый оператор завершается символом «;» — точка с запятой;
- блочные операторы языка завершаются ключевым словом «конец».

Учебный язык программирования

Прагматические принципы:

- **эффективность** выполнения не слишком важна;
- **загрузка и связывание** модулей должны выполняться динамически;
- **управление памятью** осуществляет система;
- **стандартная библиотека** должна быть реализована на учебном языке.

Semantic Language = SLang

Среда программирования

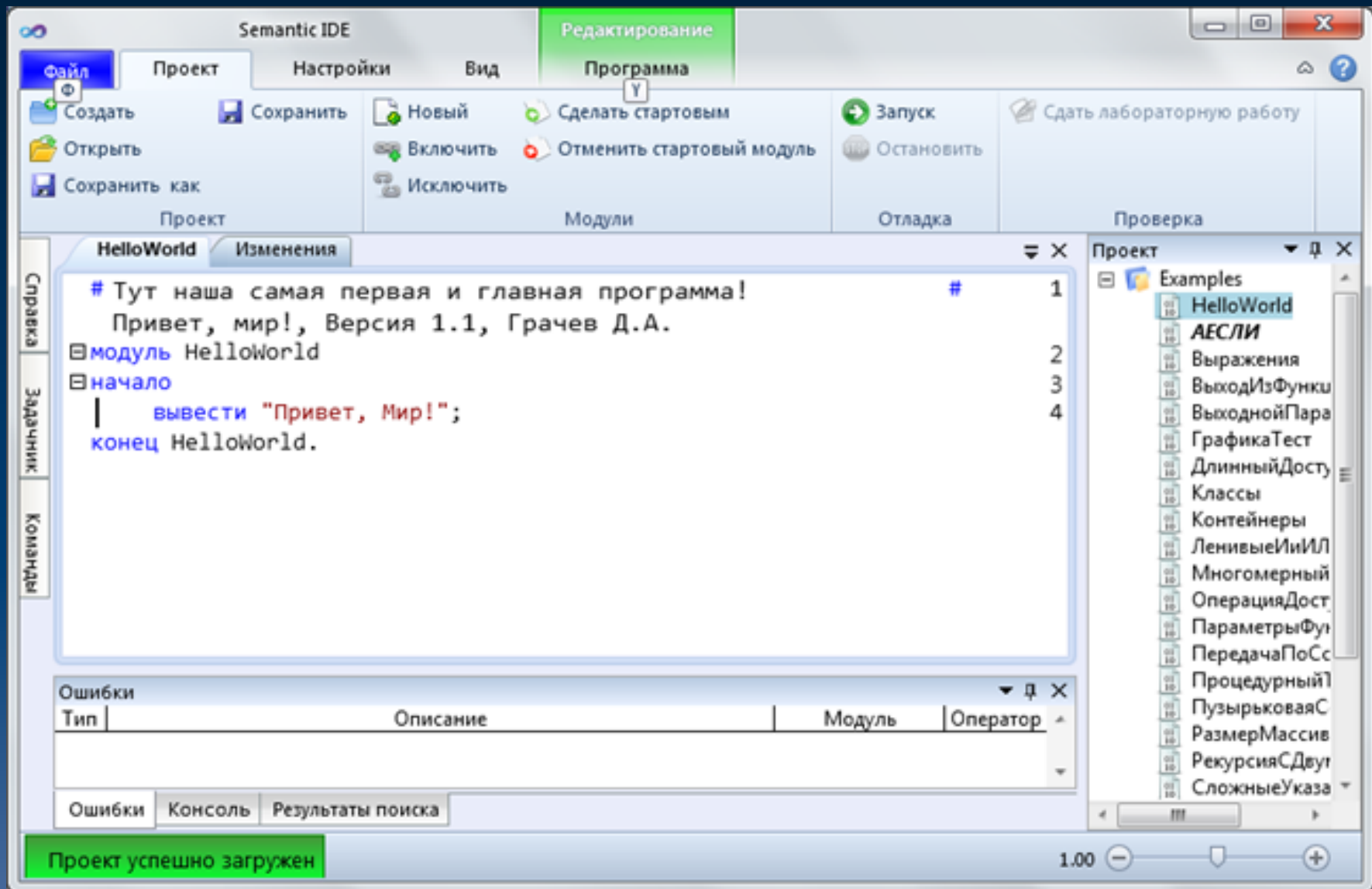
Основные концепции

- должна **поддерживать работу** с одно-, и с многомодульными программами;
- должна **обеспечивать независимый** от платформы механизм накопления программных компонент;
- **ввод/вывод** данных должен осуществляться в рамках среды без выхода в ОС;
- должна **обеспечивать разработку системы помощи** в редакторе без привлечения сторонних средств.

IDE - семантический редактор

- оперирует конструкциями учебного языка и объектами программы, а не символами текста;
- изменение ключевых слов в коде невозможно;
- ошибки определяются в момент набора программы;
- набор кода возможен в русской и в английской лексике;
- переключение лексики с русской в английскую и обратно (без повторного анализа кода);
- реализует переключение синтаксиса языка без повторного анализа кода;
- реализован механизм добавления нового синтаксиса;

IDE - семантический редактор



Пример кода

модуль Факториал

начало

переменная-целое $i := 1;$

переменная-вещ $current := 1;$

константа целое $N = 15;$

пока $i < N$ **повторять**

присвоить $current := current * i;$

присвоить $i := i + 1;$

вывести $'\n';$

вывести $current;$

конец цикла;

конец Факториал.

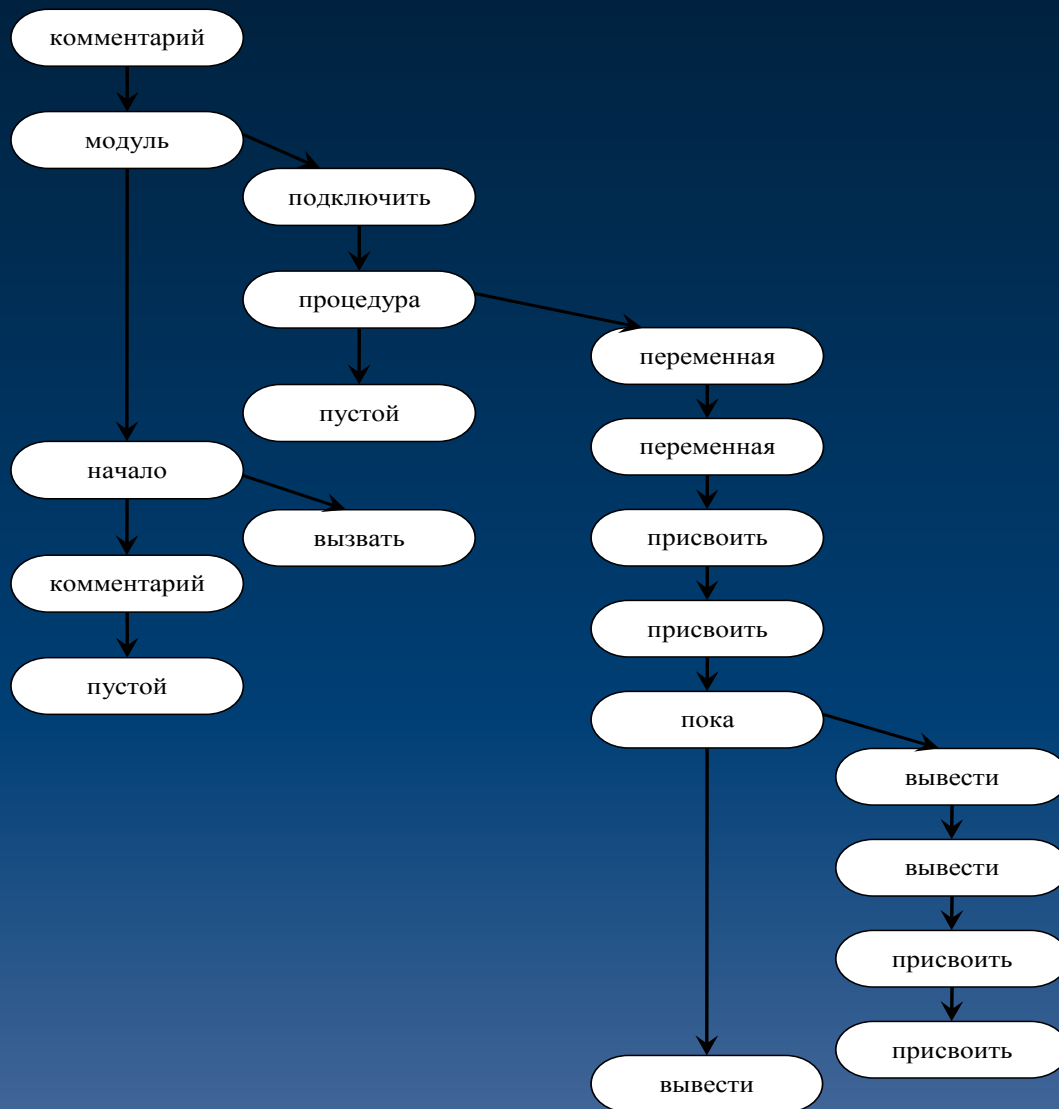
Пример кода

```
# Вычисление факториалов #  
module Факториал  
start  
    variable-integer i := 1;  
    variable-real current := 1;  
    constant integer N = 15;  
  
    while i < N repeat  
        let current := current * i;  
        let i := i + 1;  
        output '\n';  
        output current;  
    end while;  
end Факториал.
```

Пример кода

```
MODULE Факториал;  
  
BEGIN  
  VAR i: INTEGER := 1;  
  VAR current: REAL := 1;  
  CONST N: INTEGER = 15;  
  
  WHILE i < N REPEAT  
    LET current := current * i;  
    LET i := i + 1;  
    write('\n');  
    write(current);  
  END  
END Факториал;
```

Семантическое дерево



Обход дерева

метод (Узел этот) **Интерпретировать()**

если

|этот.Дочерний != пусто():

 вызвать этот.Дочерний.**Интерпретировать()**;

|этот.Следующий != пусто():

 вызвать этот.Следующий.**Интерпретировать()**;

конец ветвления;

вызвать память.СобратьМусор();

конец Интерпретировать;

- **Позволяет** автоматизировать построение сценариев выполнения задания
- **Позволяет** реализовать механизм оценивания умений и навыков программирования

Проекты среды

The screenshot displays the Semantic IDE application window. The title bar reads "Semantic IDE". The menu bar includes "Проект", "Работа", "Настройки", and "Вид". The left sidebar contains the following items: "Создать", "Открыть", "Сохранить", "Сохранить как...", "Закреть", "Прототип", "Локализация", "Последние проекты", "Проекты среды" (highlighted), "О программе", and "Выйти". The main area lists environment projects:

- Примеры
C:\Users\W\Documents\Semantic IDE\Examples\Examples.prj
- Стандартная библиотека
C:\Users\W\Documents\Semantic IDE\Framework\Framework.prj
- Документация
C:\Users\W\Documents\Semantic IDE\Help\Справка.prj
- Задачник
C:\Users\W\Documents\Semantic IDE\Taskbook\Задачник.prj

The Windows taskbar at the bottom shows icons for the Start menu, File Explorer, Home, Internet Explorer, Mail, Photos, Task View, and several application windows. The system tray on the right shows the volume icon, network status, and the date/time: 0:36, 27.10.2018.

Проект примеров

The screenshot displays the Semantic IDE interface with the following components:

- Menu Bar:** Проект, Работа, Настройки, Вид.
- Toolbar:** Создать, Сделать стартовым, Выполнить, Запустить, Пройти тест, Включить, Отменить стартовый, Остановить, Закончить, Настройки теста, Исключить, Модули, Отладка, Правка, Контрольный режим, Матем. выражения.
- Code Editor:** Contains Pascal code for a factorial module. The line `write(current);` is highlighted in blue.
- File Explorer (Project):** Lists various example files such as HelloWorld, АЕСЛИ, Выражения, etc.
- Error Console:** A table with columns: Тип, Описание, Имя модуля, Оператор. It is currently empty.
- Command Console:** Shows the output of the program execution: "Началось выполнение модуля 'Факториал'", followed by the numbers 1, 2, and a dash (-).
- Status Bar:** Includes a red notification "Проект уже открыт в Semantic IDE", a zoom level of 1.00, and a page number 17.

Проект уже открыт в Semantic IDE

1.00

Проект Задачник

Semantic IDE

Проект Работа Настройки Вид

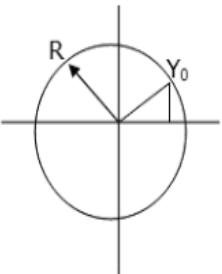
Создать Сделать стартовым Выполнить Запустить Пройти тест
Включить Отменить стартовый Остановить Закончить Настройки теста
Исключить Визуализировать Изменить пароль

Модули Отладка Плавка Контрольный режим Матем. выражения

Лекция_02 Главная

$L \leq R$

$L = \sqrt{X_0^2 + Y_0^2}$



```
graph TD; Start([начало]) --> Input[/X0, Y0, R/]; Input --> Calc["L = sqrt(X0^2 + Y0^2)"]; Calc --> Assign["K := L <= R"]; Assign --> Output[/K/]; Output --> End([конец]);
```

Проект

- Задачник
 - Лекции
 - Лекция_01_Знакомство_со_средой
 - Лекция_02_Ветвления
 - Лекция_02
 - Пример_1_Л_02
 - Пример_2_Способ_1_Л_02
 - Пример_2_Способ_2_Л_02
 - Пример_3_Способ_1_Л_02
 - Пример_3_Способ_2_Л_02
 - Пример_3_Способ_3_Л_02
 - Лекция_03_Циклические_алгоритмы
 - Лекция_04_Процедурное_программирование
 - Лекция_05_Рекурсия
 - Лекция_06_Классы
 - Лекция_07_Файлы
 - Лекция_08_Массивы
 - Лекция_09_Матрицы
 - Лекция_10_Динамические_структуры_данных
 - Практические_работы
 - Варианты_заданий
 - Лабораторная_01
 - Лабораторная_02

Проект Задачник Документация

Ошибки

Тип	Описание	Имя модуля	Оператор
-----	----------	------------	----------

Консоль

Началось выполнение модуля "Факториал"

```
1
2
-
```

Ошибки Результаты тестирования Журнал команд Результаты поиска Консоль

1.00

Выводы – 5 лет эксплуатации

- **Участие в реальном проекте** повышало мотивацию студентов.
- **Русская лексика** языка упрощает для начинающих понимание и усвоение основ программирования.
- **Отсутствие синтаксических ошибок** при наборе кода позволяет больше внимания уделять решению задачи,
- **Скорость набора кода** существенно выше, чем в традиционном текстовом редакторе.
- **Отсутствие** операций с буфером обмена не позволяет использовать «копипаст» и стимулирует писать функции.
- **Нетрадиционный язык** и нетекстовый формат файла программы снижает количество попыток нечестного выполнения заданий.

Недостатки проекта

- Проектная документация отсутствует, поэтому проект сложно развивать.
- Работает только на платформе Windows и перенос его на Linux или в Интернет по трудоемкости сравним с новой разработкой.
- Среда является локальной, предназначена для одного пользователя и не содержит никаких средств для коллективной работы.
- Учебный язык интерпретируется, исполняемый файл не создается, и программы работают только в среде, что не всегда удобно.

Студенты испытывают трудности при переходе к работе в профессиональных средах

Новая система

- Исполняющая подсистема (IDE) – клиент
- Управляющая подсистема – сервер
- Сервер базы данных
- Клиент преподавателя

Все разработки выполняются группой студентов под управлением автора в рамках **курсового проектирования** по различным дисциплинам.

Новая система – IDE

Редактор на основе VSCode – дорабатывается до семантического редактора

Компилятор Slang -> C++

- Front-end – парсер
- Back-end – конвертер Slang -> C++

IDE – 2 варианта:

- автономная система
- клиент для управляющей подсистемы.

Новая система – сервер

- подсистема «антиплагиат»
- подсистема проверки задания
- подсистема оценки задания
 - оценка выполнения
 - оценка качества кода
 - оценка действий программиста
- подсистема общей оценки
- подсистема генерации заданий
- подсистема адаптивного управления обучением

Новая система – сервер БД

- пользователи системы
- множество выполненных заданий подсистемы «антиплагиат»
- множество тестовых вопросов разного типа для проверки знаний
- наборы конкретных заданий по разным темам разной сложности
- множество типовых шаблонов заданий для генерации конкретных вариантов
- Все результаты работы каждого пользователя

СУБД PostgreSQL

Новая система – средства реализации

Компилятор Slang -> C++

- ANTLR 4, C#, Net Core
- Code::Blocks, C++, TDM-GCC

Управляющая система и веб-клиент

C#, ASP.NET Core, Entity Framework Core,
HTML5, Bootstrap 3, jQuery.

Rider IDE – JetBrains Product Pack for Students.

Проверялось: ОС Windows 7 и выше

Linux-дистрибутивах (Debian – Ubuntu 14.04).



Лаптев Валерий Викторович

WLaptev@yandex.ru

Laptev@ilabsltd.com

8-905-361-55-24

Астраханский государственный технический
университет, кафедра АСОИУ