

## **План статьи журнала «Теория и практика языков программирования», 2013**

1. Основополагающие принципы программирования
2. История развития
3. Подходы возможности представления теорий и языков программирования

## **Тезисы статьи журнала «Теория и практика языков программирования», 2013**

**Автор** Орлов А.С.

### **Отражение классического содержания учебной дисциплины по языкам программирования.**

Для использования языка программирования следует описать каждую конструкцию языка в отдельности, а также ее применение в совокупности с другими конструкциями. В языке существует множество различных конструкций, точное определение которых необходимо как программисту, применяющему язык, так и разработчику компилятора для этого языка. Программисту эти знания позволяют прогнозировать вычисления, производимые операторами программы.

Ранние попытки точно задать семантику делались путем добавления расширений к BNF-грамматике языка программирования. Дополнительную информацию о семантике можно было извлечь из дерева грамматического разбора. Идея, предложенная Дональдом Кнудом, заключалась в том, чтобы сопоставить каждому узлу дерева разбора программы некоторую функцию, определяющую семантику этого узла. Подобная информация сохранялась в так называемых атрибутах.

Атрибутные грамматики можно рассматривать как обобщение аппарата синтезируемых атрибутов. Они позволяют осмысливать конструкции в зависимости от окружающего контекста. В этом случае значения атрибутов передаются как вверх, так и вниз по дереву разбора. Дело в том, что смысл узла на дереве разбора может зависеть не только от его поддеревя, но и от информации из другой части дерева.

Операционное определение языка программирования описывает выполнение программы, составленной на данном языке, средствами виртуального компьютера. Виртуальный компьютер определяется как абстрактный автомат. Внутренние состояния этого автомата моделируют состояния вычислительного процесса при выполнении программы.

Аксиоматическая семантика основывается на применении аппарата исчисления предикатов и теории доказательств. Семантику каждой конструкции языка определяют, как некий набор аксиом или правил вывода, используемый для вывода результатов выполнения этой конструкции. Чтобы понять смысл всей программы (то есть разобраться, что и как она делает), эти аксиомы и правила вывода следует применять так же, как при доказательстве обычных математических теорем. Аксиомы и правила вывода используют для оценки значений переменных после выполнения каждого оператора программы.

Денотационная семантика синтезирует определение функции, которую вычисляет каждая программа, написанная на языке программирования. Спецификация языка создается как иерархия определений функций, вычисляемых каждой отдельной программной конструкцией.