**Федеральное государственное образовательное бюджетное**

**учреждение высшего образования**

**«Ульяновский государственный университет»**

**Факультет математики информационных и авиационных технологий**

**Доклад**

**По теме “Язык программирования Miranda”**

**Выполнил студент группы ПРИ-0-21 05.04.2025 Нуштаев Н.П.**

**Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Перцев А. А.**

Ульяновск 2024

**Введение**

Язык программирования Miranda — это функциональный язык программирования, разработанный в 1985 году британским учёным Дэвидом Тёрнером. Название языка происходит от имени персонажа Шекспира из пьесы «Буря», что символизирует его элегантность и академическую направленность. Miranda стал одним из первых коммерчески доступных чисто функциональных языков программирования и оказал значительное влияние на развитие современных функциональных языков, таких как Haskell.

Основная цель создания Miranda заключалась в предоставлении удобного инструмента для решения задач, связанных с обработкой данных, математическими вычислениями и алгоритмами. Язык отличается своей лаконичностью, строгой типизацией и поддержкой ленивых вычислений (lazy evaluation), что делает его мощным средством для написания компактных и эффективных программ.

**Основные характеристики языка**

1. Функциональная парадигма  
   Miranda является чисто функциональным языком, что означает, что программы в нём представляют собой набор функций, которые преобразуют входные данные в выходные. В отличие от императивных языков, здесь отсутствует понятие состояния или изменяемых переменных. Это позволяет писать более надёжный и предсказуемый код.
2. Ленивые вычисления (Lazy Evaluation)  
   Одной из ключевых особенностей Miranda является использование ленивых вычислений. Это означает, что выражения вычисляются только тогда, когда их результат действительно необходим. Такой подход позволяет работать с бесконечными структурами данных, такими как потоки или списки, что значительно упрощает написание сложных алгоритмов.
3. Строгая типизация  
   Miranda поддерживает строгую статическую типизацию, но при этом типы переменных не нужно явно указывать — они выводятся автоматически компилятором. Это сочетание строгости и удобства делает язык безопасным и простым в использовании.
4. Рекурсия как основной механизм повторения  
   В Miranda отсутствуют традиционные циклы, такие как **for** или **while**. Вместо этого повторение реализуется через рекурсию, что является естественным для функциональных языков.
5. Поддержка полиморфизма  
   Miranda поддерживает параметрический полиморфизм, что позволяет создавать универсальные функции, работающие с данными разных типов. Например, функция для работы со списками может быть применима к спискам целых чисел, строк или других типов данных.
6. Алгебраические типы данных  
   Язык предоставляет возможность определять собственные типы данных, включая рекурсивные структуры, такие как списки, деревья и другие абстракции.

**Примеры кода:**

Факториал –   
fac 0 = 1

fac n = n \* fac (n - 1)

Бесконечный список чисел Фибоначчи –

fib = 1 : 1 : [a + b | a <- fib; b <- tl fib]

Фильтрация чётных чисел –

evens [] = []

evens (x:xs) = if x mod 2 = 0 then x : evens xs else evens xs

**Преимущества и недостатки**

**Преимущества:**

1. Чистота и предсказуемость — отсутствие побочных эффектов делает программы более надёжными.
2. Ленивые вычисления — позволяют работать с бесконечными структурами данных.
3. Автоматический вывод типов — уменьшает количество ошибок и упрощает написание кода.
4. Математическая основа — язык хорошо подходит для задач, связанных с математикой и логикой.

**Недостатки:**

1. Ограниченная производительность — ленивые вычисления могут приводить к неэффективному использованию ресурсов.
2. Сложность отладки — отсутствие состояний и побочных эффектов затрудняет отладку программ.
3. Низкая популярность — Miranda не получил широкого распространения, что ограничивает доступность библиотек и инструментов.

**Влияние на развитие программирования**

Miranda сыграл важную роль в развитии функционального программирования. Его концепции легли в основу таких языков, как Haskell, который стал одним из самых популярных функциональных языков. В частности, многие идеи Miranda, такие как ленивые вычисления, автоматический вывод типов и работа с алгебраическими типами данных, были унаследованы и развиты в Haskell.

Кроме того, Miranda популяризировал функциональный подход в академической среде, что способствовало его дальнейшему изучению и применению в научных исследованиях.

**Заключение**

Язык программирования Miranda — это важный этап в истории развития функционального программирования. Несмотря на то, что он не получил широкого распространения, его идеи оказали огромное влияние на современные языки и подходы к разработке программного обеспечения. Miranda остаётся интересным объектом изучения для тех, кто хочет глубже понять принципы функционального программирования и его преимущества.

**Вывод**: Miranda — это язык, который заложил основу для развития функциональных языков программирования и продолжает вдохновлять разработчиков и исследователей по всему миру.