Белый Антон:

1. **Что представляет собой стек-ориентированное программирование?**

Стек-ориентированная парадигма программирования использует для передачи параметров модель стека.

Стек-ориентированный язык программирования оперирует одним или несколькими стеками и обычно использует префиксную или постфиксную нотацию вместо инфиксной, обычной для других языков. Две основные операции, которые выполняются над данными в стеке — pop (удалить верхний элемент и вернуть его) и push (добавить элемент в верх стека). Иногда стек-ориентированные языки предоставляют и более сложные операции, например, dup (скопировать верхний элемент стека и добавить его в верх стека), swap (поменять местами два верхних элемента стека), roll (циклически переставить элементы в заданной части стека) и drop (удалить верхний элемент стека, не возвращая его).

1. **Какая из парадигм, которую вы не знали, вам больше всего понравилась/заинтересовала?**

Нестрогая, обобщенная

1. **Какие парадигмы являются основными для определения языка программирования?**

Не стоит упоминания «на уровне значений» например

Григорьев Антон:

1. **Возможно ли существование языка программирования, который поддерживает лишь одну парадигму?**

Действительно, используемого языка воплощающего одну парадигму не существует. Но если взять в пример Brainfuck - то это эзотерический язык программирования, императивный и на уровне значений. Что тянет за собой причины.

1. **Есть ли парадигма, которая очевидно превосходит другие парадигмы? Или у всех есть как плюсы, так и минусы?**

У всех есть как плюсы так и минусы, но у некоторых больше того или иного

1. **Почему ООП стал так популярен?**

Потому что программирование — это искусство оперирования абстрактной информацией. Это очень сложное искусство — поскольку человеческий мозг вообще не приспособлен для оперирования абстракциями.

1. **Почему в нашем университете нас знакомили только с ООП?**

Потому что программирование — это искусство оперирования абстрактной информацией. Это очень сложное искусство — поскольку человеческий мозг вообще не приспособлен для оперирования абстракциями.

1. **Есть ли язык, который поддерживает все парадигмы, и делает ли это его лучшим языком?**

На данный момент все языки мультипарадигмальные, и тырыпыры

Гетьман Святослав

1. **Почему нет языка, чётко воплощающего одну и только одну парадигму программирования?**
2. **Табличное программирование способно выступить хорошим ORM?**

ХЗ

1. **А возможно написать язык без использования парадигмы на уровне значений?**

Да, парадигма на уровне функций

1. **Были ли попытки обобщить все эти парадигмы в единое целое для создания самого универсального языка программирования? Чем они закончились и почему?**

Борисевич Павел

1. **Какие основные отличия нестрого программирования от строгого?**

В строгости. Строгая модель вычислений означает, что аргументы всегда вычисляются полностью до применения функции к ним. Нестрогая модель вычислений означает, что аргументы не вычисляются до тех пор, пока их значение не используется в теле функции. В ряде языков булевы выражения имеют нестрогий порядок вычисления, называемый в русской литературе «вычислениями по короткой схеме», где вычисления прекращаются, как только результат становится однозначно предсказуем — например, значение «истина» в операции дизъюнкции, «ложь» в операции конъюнкции, и так далее. Операторы ветвления зачастую также имеют ленивую семантику вычислений, то есть возвращают результат всего оператора, как только однозначная ветвь его породит.

1. **Где используется или использовалось табличное программирование?**
2. **Какие есть примеры языков, поддерживающих матричное программирование? Сейчас такие языки программирования где-нибудь используются?**

Матричное программирование — парадигма, обобщающая некоторые операции так, что они могут применяться сразу к массивам значений, а не к отдельными значениям. MatLab

Ярошевич Яна

1. **Где используется процедурная парадигма?**

Процедурное программирование — парадигма, основанная на использовании процедур. Процедура (иногда также называемая подпрограммой или методом) — это последовательность команд, которые следует выполнить. Любая процедура может быть вызвана из любой точки программы, включая другие процедуры или ее же саму (рекурсивный вызов).

Процедурное программирование является частным случаем императивной парадигмы.

1. **Когда и при каких обстоятельствах появился термин "парадигма программирования»?**

Своим современным значением в научно-технической области термин «парадигма» обязан, по-видимому, Томасу Куну и его книге «[Структура научных революций](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%BD%D0%B0%D1%83%D1%87%D0%BD%D1%8B%D1%85_%D1%80%D0%B5%D0%B2%D0%BE%D0%BB%D1%8E%D1%86%D0%B8%D0%B9)» (см. парадигма). Кун называл парадигмами устоявшиеся системы научных взглядов, в рамках которых ведутся исследования. Согласно Куну, в процессе развития научной дисциплины может произойти замена одной парадигмы на другую (как, например, геоцентрическая небесная механика Птолемея сменилась гелиоцентрической системой Коперника), при этом старая парадигма ещё продолжает некоторое время существовать и даже развиваться благодаря тому, что многие её сторонники оказываются по тем или иным причинам неспособны перестроиться для работы в другой парадигме.

Термин «парадигма программирования» впервые применил в 1978 году Роберт Флойд в своей лекции[2] лауреата премии Тьюринга.

Флойд отмечает, что в программировании можно наблюдать явление, подобное парадигмам Куна, но, в отличие от них, парадигмы программирования не являются взаимоисключающими:

Если прогресс искусства программирования в целом требует постоянного изобретения и усовершенствования парадигм, то совершенствование искусства отдельного программиста требует, чтобы он расширял свой репертуар парадигм.

Таким образом, по мнению Роберта Флойда, в отличие от парадигм в научном мире, описанных Куном, парадигмы программирования могут сочетаться, обогащая инструментарий программиста.

1. **С какими языками связана обобщенная парадигма?**

Обобщённое программирование состоит в написании алгоритмов в терминах абстрактных типов данных; когда алгоритм используется для конкретных типов данных, создается экземпляр этого алгоритма с типами данных, переданными в качестве параметров. Такой стиль программирования позволяет использовать универсальный код для похожих заданий, имеющих дело с разными типами данных, и таким образом уменьшить дублирование кода.

Обобщённое программирование широко используется для реализации универсальных контейнеров и алгоритмов. Так, стандартная библиотека шаблонов STL в С++ предоставляет набор контейнеров (динамический массив, связный список, очередь, множество и т.д.) и алгоритмов, применимых к этим или пользовательским контейнерам.

1. **Что за парадигма "обмен сообщениями»?**

Обмен сообщениями (message passing) — метод программирования, который основан на рассылке сообщений — адресной или широковещательной. Сообщения могут иметь любую природу — данные, экземпляры классов, фрагменты кода и т.д. (А++)

Ровдо Дарья

1. **Что представляет собой парадигма "обмен сообщениями"?**
2. **В чем фишка языка «Cat»?**

Cat — функциональный стек-ориентированный язык программирования, созданный под влиянием Joy.

Cat был создан Кристофером Диггинсом (Christopher Diggins), который остается его главным разработчиком. У языка есть несколько реализаций, включая официальную, написанную на C#.

В спецификации Cat определены три уровня команд языка. level-0, или ядро Cat, — чисто функциональный язык. Следующие уровни расширяют набор доступных команд.

Все команды языка обмениваются данными через единую структуру данных — стек. Стек может содержать числа, строки, списки, функции и любые другие данные. Это привносит в язык некоторые особенности, отсутствующие в других стековых языках.

1. **Что обозначает термин табличное программирование? Как я понимаю, это тоже самое, что работа с базой данных, в которой тоже данные хранятся в таблицах. То есть табличное программирование это как программа, в которой все данные находятся в бд?**

Михальцова Анна

1. **Что представляют собой "нестрогие функции"?**
2. **Как возникают парадигмы программирования?**

По мере необходимости и по мере извращенности

1. **К какой парадигме можно отнести язык R?**

императивная, матричная, на уровне значений, объектно-ориентированная, процедурная, рефлексивная, функциональная

1. **Чем отличаются табличная и матричная парадигма?**

Грушевский Андрей

1. **В чем преимущества одних парадигм перед другими, когда это критически?**

Когда надо

1. **Что такое матричное программирование (или что-то такое)?**
2. **В каких случаях использование одной или другой парадигмы значительно упрощает код или увеличивает скорость написания программы?**

Когда случается

Трубач Геннадий

1. В чем минусы ООП?
2. **Что такое АОП?**

Аспектно-ориентированное программирование (АОП) — парадигма программирования, основанная на идее разделения функциональности, особенно сквозной функциональности, для улучшения разбиения программы на модули.

1. **Что такое строгие и нестрогие функции?**
2. **Что такое атом в парадигме на уровне функций?**

атомы — единицы данных, с которыми оперируют функции. Данные появляются только на входе и выходе программы, и нигде внутри. Атомы могут быть скалярами или множествами других атомов.

Ипатов Алексей

1. **Почему эзотерическая парадигма не воспринимается всерьёз?**

Потому что потому

1. **Где применяется табличная парадигма?**
2. **Расскажите про скалярную парадигму.**

Скалярное программирование — низкоуровневая парадигма, диктующая отсутствие в языке матричных операций. Каждая операция применяется к отдельным скалярным величинам, но не ко всему массиву. Таким образом, программист должен организовать обработку массива как последовательность скалярных операций.