

Парадигмы программирования

Для вопросов по курсу:

natalya.razmochaeva@moevm.info

Префикс в теме письма [CS_03XX]

*Берленко Татьяна Андреевна
Шевская Наталья Владимировна
СПбГЭТУ “ЛЭТИ”, ФКТИ, МОЭВМ*

Что такое парадигма программирования?

- Совокупность идей и понятий, определяющих стиль написания компьютерных программ (подход к программированию).
- Слово «парадигма» используется в программировании для определения семейства обозначений (нотаций), разделяющих общий способ (методику) реализаций программ

Императивная и декларативная парадигмы

<u>Императивный подход</u>	<u>Декларативный подход</u>
<ul style="list-style-type: none">● Последовательное выполнение инструкций	<ul style="list-style-type: none">● Отсутствие последовательного выполнения инструкций
<ul style="list-style-type: none">● Доступны данные после выполнения предыдущих инструкций	<ul style="list-style-type: none">● Есть описание того, как поставлена задача и как должен выглядеть результат
<ul style="list-style-type: none">● Используется оператор присваивания/именованные переменные/подпрограммы	<ul style="list-style-type: none">● Не используется оператор присваивания/подпрограммы/циклы

Императивная и декларативная парадигмы (пример)

➤ Императивный подход:

- Pascal, Python, C, C++, Java

➤ Декларативный подход:

- SQL, Prolog

Один язык может сочетать в себе императивную парадигму и подвиды декларативной.

Например, императивный Python поддерживает функциональную парадигму -- подвид декларативной парадигмы.

Процедурное программирование

- Императивная парадигма
- Последовательное выполнение операторов
- Задача декомпозируется на шаги, которые решаются один за другим

Объектно-ориентированное программирование

- Программа -- совокупность объектов.
- Объект -- экземпляр класса.
- Класс -- описание некоторого типа данных.
- Классы образуют иерархию наследования.

Программа -- или модуль -- тоже заворачивается в ООП и становится классом, к атрибутам которого (функции и переменные, которые определены в модуле) можно обращаться

Логическое программирование: Prolog

- Prolog -- логический (декларативный) язык программирования.
- Используется в области искусственного интеллекта, компьютерной лингвистики и нечислового программирования в целом.
- Логика программы выражается в терминах отношений, представленных в виде фактов и правил.
- Для того чтобы инициировать вычисления, выполняется специальный запрос к **базе знаний**, на которые система логического программирования генерирует ответы «истина» и «ложь».

Логическое программирование. Пример на Prolog

domains

person = symbol

predicates

male(person)

female(person)

parents(person, person, person)

clauses

male("Frank").

male("Sam").

female("Mary").

female("Debbie").

parents("Sam","Frank","Mary").

parents("Debbie","Frank","Mary").

who_is_the_son :- parents(Son, Father, Mother), male(Son),
write("The son is ",Son,"."),nl.

goal

who_is_the_son

Результат работы программы на Prolog

The son is Sam.

Функциональное программирование

- Функциональное программирование — раздел дискретной математики и парадигма программирования, в которой процесс вычисления трактуется как вычисление значений функций в математическом понимании последних (в отличие от функций как подпрограмм в процедурном программировании).
- Относится к декларативной парадигме.
- Не предполагается явное хранение состояния программы.

Функции в функциональном программировании

➤ **Функция высшего порядка**

Функция, которая может принимать на вход и/или возвращать другую функцию.

➤ **Чистая функция**

Функция зависит только от своих параметров и не взаимодействует с внешними данными. Это значит, что для одних и тех же данных гарантировано получится один и тот же результат (также говорят, что функция детерминирована и не имеет *побочных эффектов*).

Чистая функция

- зависит только от своих параметров и не взаимодействует с внешними данными (не меняет глобальные переменные, не имеет побочных (side-effect) эффектов).
- не выводит информацию вовне (будь файл, или поток вывода).
- не меняет аргументы, поданные на вход функции.
- не меняет ход выполнения программы (не использует принудительное завершение программы, например);
- не использует внутри себя “нечистые” функции.

Функциональное программирование. Свойства

- Данные неизменяемые.
- Программа -- совокупность чистых функций.
- Отсутствие циклов.
- Использование функций высшего порядка.
- Функция может быть сохранена в переменную.
- Функция не зависит от имени, по которому мы к ней обращаемся.

Примеры языков функционального программирования

- Лисп
- F#
- Haskell

Источники и полезные ссылки

- Сравнение языков программирования по парадигмам:
https://ru.wikipedia.org/wiki/Сравнение_языков_программирования
- Пример программы на языке Пролог
<http://www.verim.org/project/prolog/listing/prezidenty>
- Википедия о языке Пролог:
[https://ru.wikipedia.org/wiki/Пролог_\(язык_программирования\)](https://ru.wikipedia.org/wiki/Пролог_(язык_программирования))
- Функциональное программирование:
https://ru.wikipedia.org/wiki/Функциональное_программирование

Вопросы по курсу можно задавать:

Шевская Наталья Владимировна
natalya.razmochaeva@moevm.info,

Берленко Татьяна Андреевна
tatyana.berlenko@moevm.info