

# Парадигмы программирования

---

Для вопросов по курсу:

Шевская Наталья Владимировна, [natalya.shevskaya@moevm.info](mailto:natalya.shevskaya@moevm.info)

Иванов Дмитрий Владимирович, [dmitry.ivanov@moevm.info](mailto:dmitry.ivanov@moevm.info)

Префикс в теме письма [CS\_23XX]



# План семестра

- Прохождение основного курса и защита лабораторных по темам:
  - Парадигмы программирования и ООП
  - Введение в алгоритмы и структуры данных
  - Введение в анализ данных
- Контрольная работа
- Подведение итогов



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



# 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

- Класс -- описание некоторого типа данных.
- Объект -- экземпляр класса.
- Поля -- свойства/характеристики объектов, описанные в классе.
- Методы -- действия, которые можно делать над объектами или с объектами, определенные в классе.
- Атрибуты -- поля и методы.



# Объектно-ориентированное программирование

- Программа -- совокупность объектов.
- Классы образуют иерархию наследования.

*Программа в Python (модуль) -- тоже “заворачивается” в ООП и становится классом, к атрибутам которого (функции и переменные, которые определены в модуле) можно обращаться через точку*





# 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».



# 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](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](https://ru.wikipedia.org/wiki/Функциональное_программирование)

# Вопросы по курсу можно задавать:

---

Шевская Наталья Владимировна  
[natalya.shevskaya@moevm.info](mailto:natalya.shevskaya@moevm.info)

Иванов Дмитрий Владимирович  
[dmitry.ivanov@moevm.info](mailto:dmitry.ivanov@moevm.info)