# Теоретичні запитання з екзаменаційних білетів з ФП 2022 5 балів за теоретичне запитання в кзаменаційному білеті

Поняття, характеристика, особливості парадигм декларативного та імперативного програмування, різниця між цими парадигмами. Приклади програм, що написані мовою імперативного та функціонального програмування

Концепція функціонального програмування: відмінності від імперативного програмування, особливості функціонального стилю програмування, приклад коду функціональною та алгоритмічною мовами.

Функції у функціональному програмуванні: математичне визначення функції та її еквівалентне означення у функціональному програмуванню, застосування (виклик) функції, поняття параметрів та значень функції, поняття прозорості посилань, функція як величина функція як величина даних першого класу, композиція функцій, поняття функцій вищих порядків.

Механізми програмування в мовах функціонального програмування: елементарні вирази, засоби комбінування, засоби абстракції, типи об’єктів

Основні типи даних в мовах функціонального програмування на прикладі мов Scheme (Lisp, Haskell, Scala тощо), навести приклади

Процедури і процеси, які вони породжують в мовах функціонального програмування лінійна рекурсія та ітерація, деревоподібна рекурсія, різниця між рекурсивною процедурою та рекурсивним процесом, хвостова рекурсія (мінімізація неефективності рекурсії), глибина стеку, хвостові виклики та хвостовий контекст. Навести приклади рекурсивних процедур, хвостової рекурсії та переповнення стеку.

Інтерпретація поняття змінної в імперативних та функціональних мовах програмування. Альтернатива присвоєнню в строго функціональних мовах. Навести приклади змінних як фактичних значень, змінних як областей пам’яті, прив’язки значень до змінних.

Вирази в мові Scheme (Lisp, Haskell, Scala, ….): типи виразів, стандартне правило обчислення, префіксна форма, приклади застосування стандартного правила обчислення виразів до атомів, символів, виразів в дужках, підвиразів.

λ-числення як основа визначення функцій (процедур) в функціональному програмуванні. Функції (процедури) вищих порядків: процедури в якості аргументів, побудова процедур за допомогою форми lambda, процедури як значення, що повертаються. Навести приклади коду для застосування процедур вищих порядків.

Основні правила і приклади побудови рекурсивних функцій: поняття хвостової рекурсії та її оптимізація в мовах функціонального програмування. Приклад хвостової рекурсії на Scheme (Lisp, Haskell, Scala та інш.)

Вектори: поняття, визначення, порівняння вектора та списку, поняття довжини та індексів вектора, стандартні процедури обробки векторів Навести приклади коду роботи з векторами.

Класифікація типів виразів (примітивні та похідні). Примітивні типи виразів: посилання на змінні, літерні вирази, процедурі виклики, управління обчисленням (моделювання розгалуження (if), моделювання циклів). Навести приклади перерахованих типів виразів

Комбінаційні форми в мовах функціонального програмування: поняття, префіксна нотація, загальне правило обчислення комбінацій, рекурсивне обчислення комбінацій.

Поняття послідовності в мові Scheme (Lisp, Haskell, Scala, інш.) як впорядкованої сукупності об’єктів даних: подання послідовностей, стрілочна діаграма як спосіб графічного зображення послідовності, засоби створення послідовності та доступу до елементів послідовності

Точкові пари в мові Scheme (Lisp): області застосування, відмінність від спискового запису, операції створення пари, предикати, селектори, приклади коду

Ієрархічні структури: дерева, бінарні дерева, дерева бінарного пошуку, процедури створення, пошуку, фільтрації даних, перетворення елементів дерева, накопичення перетворених даних. Навести приклад коду.

Множини як неупорядковані списки: створення множини, додавання елементів, перевірки співпадіння елементів, реалізація множинних операцій (перетин, об’єднання, різниця, приналежність елемента множині). Дати приклади коду.

Присвоєння у функціональних мовах: роль оператора присвоєння в мовах програмування, форма set! для побудови об’єктів, що мають внутрішній стан, моделювання стану програми за допомогою локальних змінних, моделювання зміну стану за допомогою оператора присвоєння. Навести приклад коду застосування оператора присвоєння та зміни стану програми.

Подання дерев в мові Scheme (Lisp): поняття дерева, абстракція для роботи з деревами (конструктор та селектори дерева).

Відмінності між лінійно-рекурсивним процесом обчислень та лінійно-ітеративним процесом обчислень, навести приклади. Поняття простої та хвостової рекрсії в мовах функціонального програмування

Ієрархічні дані та властивість замикання. Поняття та подання ієрархічних структур типу дерев. Навести приклади коду

Черги: поняття черги, подання черги як структури даних, набір операцій, що визначають чергу (конструктор, селектори, мутатори), неефективність подання черги як звичайного списку, додавання та вилучення елементів з черги. Навести приклади коду для роботи з чергами

Символьні обчислення в Scheme (Lisp): символьне диференціювання алгебричних виразів

Математичні функції як базіс функціонального програмування. Основні принципи функціонального програмування

Функції у функціональному програмуванні: математичне визначення функції та її еквівалентне означення у функціональному програмуванню, застосування (виклик) функції, поняття параметрів та значень функції, поняття прозорості посилань, функція як величина даних першого класу, композиція функцій, поняття функцій вищих порядків.

Поняття послідовності в мові Scheme (Lisp) як впорядкованої сукупності об’єктів даних: подання послідовностей, стрілочна діаграма як спосіб графічного зображення послідовності, засоби створення послідовності та доступу до елементів послідовності

Принципи функціонального програмування. Порівняння функціонального стилю програмування з імперативним. Переваги функціонального стилю перед імперативним.

Присвоєння у функціональних мовах: роль оператора присвоєння в мовах програмування, форма set! для побудови об’єктів, що мають внутрішній стан, моделювання стану програми за допомогою локальних змінних, моделювання зміну стану за допомогою оператора присвоєння. Навести приклад коду застосування оператора присвоєння та зміни стану програми.

λ-числення як теоретичний базис функціонального програмування. Опис функцій як λ-виразів. Вільні та зв’язані ідентифікатори. Застосування (аплікація) значень до λ-виразів. Редукція. Теза Черча-Тьюринга.