**Перелік теоретичних запитань до іспиту з функціонального програмування**

1. Концепція функціонального програмування: відмінності від імперативного програмування, особливості функціонального стилю програмування, приклад коду функціональною та алгоритмічною мовами.
2. Математичні функції як базіс функціонального програмування
3. Основні принципи функціонального програмування
4. Порівняльний аналіз деяких мов функціонального програмування, зокрема сім’я LISP
5. Особливості мови *Scheme* : конструктивні особливості, області використання мови
6. Механізми програмування *Scheme*: елементарні вирази, засоби комбінування, засоби абстракції, типи об’єктів
7. Основні типи в мові Scheme (Lisp, Haskell), навести приклади
8. Комбінаційні форми: префіксна нотація, загальне правило обчислення комбінацій, рекурсивне обчислення комбінацій
9. Аплікативний і нормальний порядки обчислення процедур в Scheme (Lisp)
10. Основні правила і приклади побудови рекурсивних функцій, навести приклади.
11. Лямбда-обчислення як основа визначення функцій в мовах функціонального програмування, навести приклади.
12. Відмінності між лінійно-рекурсивним процесом обчислень та лінійно-ітеративний процес обчислення, навести приклади
13. Поняття хвостової рекурсії та її оптимізація в мовах функціонального програмування. Приклад хвостової рекурсії на Scheme (Lisp, Haskell)
14. Процедури вищого порядку в Scheme (Lisp, Haskell): процедури як аргументи, процедури як значення, що повераються. Приклади коду.
15. Поняття абстракції даних в мові Scheme (Lisp): ідея абстракції даних, інтерфейс між абстрактними даними і конкретним уявлення складових даних,
16. Арифметика раціональних чисел в мові Scheme (Lisp) як приклад створення складених даних
17. Складова структура даних – пара, конструктори та селектори пари
18. Ієрархічні дані та властивість замикання. Навести приклади коду
19. Поняття послідовності в мові Scheme (Lisp) як впорядкованої сукупності об’єктів даних: подання послідовностей, стрілочна діаграма як спосіб графічного зображення послідовності
20. Точкові пари в мові Scheme (Lisp): області застосування, відмінність від спискового запису, приклади коду
21. Типові операції обробки списків чк послідовностей об’єктів, переваги використання списків.
22. Вектори, рядки, дерева як різновиди спискових структур в мові Scheme (Lisp): відмінності цих структур одна від одної.
23. Подання дерев в мові Scheme (Lisp): поняття дерева, абстракція для роботи з деревами (конструктор та селектори дерева).
24. Таблиці в мові Scheme як список записів: подання таблиць із застосуванням структур типу пари
25. Обробка файлів в Scheme (Lisp, Haskell): призначення та поняття портів та іх зв’язок із файлами, технологія обробки файлів.
26. Символьні обчислення в Scheme (Lisp): символьне диференціювання алгебричних виразів.