**Лекція 12 СТРАТЕГІЇ РОЗРОБКИ ПЗ**

Існує 3 основних стратегії розробки ПЗ:

Послідовна стратегія. Інший термін – стратегія водопаду, тобто класична лінійна послідовність етапів програмування;

Інкрементна стратегія. На початку процесу визначаються всі призначені для користувача системні вимоги, а частина програмування, що залишилася, виконується у вигляді послідовності версій. Перша версія реалізує частину запланованих можливостей, наступна версія реалізує додаткові можливості і т.д., поки не буде отримана повна система;

Еволюційна стратегія. Система також будується у вигляді послідовності версій, але на початку процесу визначені не всі вимоги. Вимоги уточнюються в результаті розробки версій.

Характеристики стратегій конструювання ПО відповідно до вимог стандарту IEEE/EIA 12207.2 приведені в табл. 1.

Таблиця 1. Характеристики стратегій програмування

Стратегія На початку процесу визначені всі вимоги? Множина циклів програмування? Проміжний ПЗ розповсюджується?

Водопадна Так Немає Немає

Інкрементна Так Так Можливо

Еволюційна Немає Так Так

ІІ. Модульне програмування

Технологія модульного програмування – що оформилася на початку 70-хроків XX століття ідея розробки великих програмних систем. Це фундаментальна концепція, що є основою всіх сучасних підходів до проектування і реалізації. В той же час суть її проста і відображає широко відомі наукові і технічні методи, що полягають в пошуку і реалізації деякого базового набору елементів, комбінації яких дають вирішення всіх завдань з певного круга.

Модульне програмування полягає в розробці під конкретне завдання або круг завдань (наочну область) власного базису у вигляді набору модулів, що дозволяє найефективніше по цілому ряду критеріїв побудувати програмний комплекс. Модулі, що входять в базис, це цілі програми (на відміну від примітивів структурного програмування), вирішальні деякі підзадачі основних завдань.

Із застосуванням модульного програмування з'являються можливості колективної розробки програм як набору «незалежних» частин, послідовного зменшення складності методом розбиття складного завдання на простіші підзадачі, нарешті, можливості повторного використання створеної раніше коди.

ІІІ. Об'єктно-орієнтоване програмування

Розвиток апаратної бази привів до можливості рішення при її допомозі все більш складніших завдань. Програми стали великими, а розробка - колективною. Об'єм роботи збільшився, колективи розрослися, код «розбухнув», і з'явилися нові проблеми, яких не було раніше. Несподівано з'ясувалося, що можливості структурного і модульного програмування обмежені і часто вже не дозволяють добиватися бажаного результату (або нічого не працює, або проект не укладається в терміни, або до бюджету, або через рік після написання програми з'ясовується, що її неможливо модифікувати і так далі).

Об'єктно-орієнтована технологія в деякій мірі вирішила більшість описаних проблем. На відміну від розглянутих раніше технологій, об'єктно-

орієнтована технологія працює на стадіях аналізу, проектування і програмування. В основі об'єктно-орієнтованої технології лежать об'єктна модель і об'єктна декомпозиція.

До основних принципів об'єктної моделі часто відносять наступні:

· абстракція;

· інкапсуляція;

· спадкоємність;

· поліморфізм.

Суть об'єктної декомпозиції полягає у виділенні в наочній області класів і об'єктів, а також зв'язків між ними, і лише потім даних і алгоритмів, якими характеризується кожен клас. Таким чином, саме класи стають основним «будівельним блоком» в ООП, тоді як раніше такими блоками були алгоритми.

IV. Компонентне програмування

Компонентне програмування – є розвитком об’єктно-орієнтованої технології. На відміну від ООП введений наступний рівень абстракції – класи об'єднуються в компоненти.

Компонент це:

· програмний код у вигляді самостійного модуля;

· може бути використаний в незмінному вигляді;

· може допускати настройку;

· володіє поведінкою (функціональністю).

Основний принцип компонентного програмування: збірка застосування з готових компонент, в загальному випадку написаних на різних мовах.

Компонент ізольований від зовнішнього світу своїм інтерфейсом – набором методів (їх сигнатурами). Компонентна програма – набір незалежних компонент, зв'язаних один з одним за допомогою інтерфейсів.