**Практична робота №5.**

**Реалізація програмного забезпечення. САSЕ-технології**

**Мета**: *ознайомитись з особливостями та перевагами застосування САSЕ-технологій*

**Теоретичні відомості**

*Процес програмування* (написання програмного коду, кодування) зазвичай випливає безпосередньо за процесом проектування. Але для деяких класів програм, наприклад критичних по надійності систем, остання стадія проектування (детальне проектування) і початок кодування можуть перекриватися. У процесі проектування можуть використовуватися САSЕ-засоби, які дозволяють одержати кістякову програму. Така програма містить код для визначення і реалізації інтерфейсів, і в багатьох випадках програмісту залишається тільки додати код, що реалізує деякі деталі функціонування програмного компонента.

**Програмування** ‒ індивідуальний процес, тут не існує загальних правил, яким необхідно випливати при написанні програмного коду. Деякі програмісти починають кодування з компонентів, які вони добре розуміють, залишаючи наостанку кодування компонентів, які є для них “темними”. Інші застосовують протилежний підхід, залишаючи прості для них компоненти на потім.

Зазвичай програмісти самі тестують написаний ними програмний код для виявлення можливих помилок і програмних дефектів. Цей процес називається налагодженням програми. У принципі тестування і налагодження є різними процесами. При тестуванні встановлюється наявність програмних помилок. У ході налагодження встановлюється місце розташування помилок, потім вони усуваються.

Програміст, який проводить налагодження повинен згенерувати такі режими роботи системи, які допоможуть виявити програмні помилки по аномальній поведінці системи. Локалізація помилок може вимагати проведення ручного трасування коду програми. У процесі тестування і налагодження можуть допомогти налагоджувальні засоби, що показують значення програмних змінних і виконуючі трасування операторів, що виконуються.

**Абревіатура САSЕ** (Computer-Aided Software Engineering - автоматизована розробка ПЗ) позначає спеціальний тип програмного забезпечення, призначеного для підтримки таких процесів створення ПЗ, як розробка вимог, проектування, кодування і тестування програм. Тому до САSЕ-засобів відносяться редактори проектів, словники даних, компілятори, відладчики, засоби побудови систем і т.п.

САSЕ-технології пропонують підтримку процесу створення ПЗ шляхом автоматизації деяких етапів розробки, а також створення і надання інформації, необхідної для розробки.

Приведемо приклади тих процесів, які можна автоматизувати за допомогою САSЕ-засобів.

1. Розробка графічних моделей системи на етапах створення специфікації і проектування.
2. Проектування структури ПЗ з використанням словників даних, що зберігають інформацію про об'єкти структури і зв'язкаи між ними.
3. Генерування користувацьких інтерфейсів на основі графічного опису інтерфейсу, створюваного в діалоговому режимі.
4. Налагодження програм на основі інформації, одержуваної в ході виконання програми.
5. Автоматична трансляція програм, написаних на застарілих мовах програмування (наприклад, COBOL), у програми, написані на сучасних мовах.

Розширення застосування САSЕ-технологій обмежують два фактори.

Створення ПЗ, особливо етап проектування, багато в чому є творчим процесом. Існуючі САSЕ-засоби автоматизують рутинні процеси, спроби залучити їх до розв'язку інтелектуальних і творчих завдань проектування особливим успіхом не увінчалися.

У багатьох організаціях-розробниках створення ПЗ ‒ результат роботи команди фахівців із програмного забезпечення. При цьому багато часу витрачається на “порожнє” спілкування між членами команди розробників. У цій ситуації САSЕ-технології не можуть запропонувати нічого такого, що здатне підвищити продуктивність праці розробників.

**Класифікація САSЕ-засобів**

Класифікація САSЕ-засобів допомагає зрозуміти їхні основні типи і роль, яку вони відіграють у підтримці процесів створення програмного забезпечення. Існує кілька різних класифікацій САSЕ-засобів, і кожна пропонує свій погляд на ці програмні продукти. Розглянемо наступні класифікації.

* Класифікація по виконуваних функціях.
* Класифікація по типах процесів розробки, які вони підтримують.
* Класифікація по категоріях, де САSЕ-засоби класифікуються по ступеню інтеграції програмних модулів, що підтримують різні процеси розробки.

У табл. 1.1 представлена класифікація по виконуваних функціях із прикладами відповідних САSЕ-засобів. Це неповний список типів САSЕ-засобів, зокрема тут не представлені засоби підтримки повторного використання програмних компонентів.

Таблиця 1.1. Класифікація САSЕ-засобів за функціями, що виконуються



У табл. 1.2 представлена інша класифікація САSЕ-засобів. Класифікація по типах показує, які процеси створення ПЗ підтримуються тими або іншими САSЕ-засобами. Засоби планування і оцінювання, редагування текстів, підготовки документації і керування конфігурацією можна використовувати на всіх етапах розробки ПЗ.

Таблиця 1.2. Класифікація САSЕ-засобів за типами підтримуваних ними процесів розробки



Інша класифікація САSЕ-засобів будується на основі широти охвату процесів розробки ПЗ, підтримуваних даним засобом. Запропонована класифікація, що містить наступні три категорії:

* Допоміжні програми (tools) підтримують окремі процеси розробки ПЗ, такі як перевірка несуперечності архітектури системи, компіляція програм, порівняння результатів тестів і т.п. Допоміжні програми можуть бути універсальними функціонально-закінченими засобами (наприклад, текстової процесор) або можуть входити до складу інструментальних засобів.
* Інструментальні засоби (workbenches) підтримують визначені процеси розробки ПЗ, наприклад створення специфікації, проектування і т.д. Зазвичай інструментальні засоби є набором допоміжних програм, які в більшому або меншому ступені інтегровані.
* Робочі середовища розробника (environments) підтримують усі або більшість процесів розробки ПЗ. Робочі середовища зазвичай включають кілька різних інтегрованих інструментальних засобів.

На практиці границі між САSЕ-засобами різних категорій розмиті. Допоміжну програму можна придбати як окремий продукт, але вона може використовуватися для підтримки різних процесів розробки. Наприклад, більшість текстових процесорів у цей час мають у своєму розпорядженні вбудованих редактори діаграм; або інструментальні САSЕ-засоби для проектування все частіше пропонують підтримку процесам програмування і тестування, тим самим наближаючись до робочих середовищ. Тому не завжди можна легко позиціонувати який-небудь САSЕ-продукт по категоріях відповідно до цієї класифікації. Разом з тим класифікація по категоріях корисна для розуміння того, наскільки широкий діапазон процесів розробки, які можуть бути підтримані тем або іншим Сasе-засобом.

**Завдання**

1. На основі інформації з попередніх робіт виберіть Case-засоби, які будуть корисні у реалізації вашого продукту. Оберіть не менше п’яти варіантів типів case-засобів за функціями, що виконуються. Свій вибір обґрунтуйте, пояснивши їх важливість для вашого конкретного програмного продукту.
2. Результати роботи оформіть у вигляді таблиці та подайте викладачу.
3. Дайте відповіді на контрольні запитання.

**Контрольні запитання**

1. Поясніть поняття case-технології.
2. Для яких процесів можуть бути використані case-технології?
3. Наведіть приклади класифікації САSЕ-засобів за функціями, що виконуються.
4. Як класифікують САSЕ-засоби на основі широти охвату процесів розробки ПЗ.