**Опитування №2**

1. Чим на Вашу думку викликана потреба в стандартизації ПЗ?
2. Які базові стандарти на розробку ПЗ Ви знаєте?
3. Як розподілені процеси ЖЦ ПЗ в стандарті ISO 12207?
4. В чому особливості стандарту ISO 12207?
5. В чому цінність стандарту ISO 12207**?**
6. Як визначається відповідно до стандарту IS0 12207 вимога кваліфікації?
7. Що поєднує стандарти ISO/IEC 12207 та ISO/IEC 15288?
8. Яке призначення стандартів серії ISO 9000?
9. Як на Вашу думку впливає стандарт СММ на якість програмного забезпечення?
10. Які процеси, які притаманні будь-якому проекту створення ПЗ? (*Етапи проектування, кодування, тестування і налагодження*)
11. Які етапи ЖЦ складають власне розробку ПЗ? *(Традиційний погляд: Постановка задачі (стадія «Технічне завдання»); Аналіз вимог і розробка специфікацій (стадія «Ескізний проект»); Проектування (стадія «Технічний проект»); Реалізація (стадія «Робочий проект»).*
12. В чому полягає логічне проектування ?(*включає ті проектні операції, які безпосередньо не залежать від наявних технічних і програмних засобів, що становлять середовище функціонування майбутнього програмного продукту*)
13. В чому полягає фізичне проектування ? (*прив'язка до конкретних технічних і програмних засобів середовища функціонування, тобто облік обмежень, визначених у специфікаціях*)
14. В чому полягає формалізація задачі при розробленні ПЗ?
15. Як специфікуються програми? (*визначаються функціональне призначення програми, вхідні дані, вихідні дані; здійснюється формальний однозначний опис задачі, ліквідуються (уточнюються) недоліки, які могли бути допущені в постановці задачі*)
16. Назвіть етапи реалізації ПЗ*.( Реалізація являє собою процес поетапного написання кодів програми на обраною мовою програмування (кодування), їх тестування і налагодження*)
17. Як пов‘язане тестування та налагодження програм? (*тестування це перевірка відповідності між реальною поведінкою програми і її очікуваною поведінкою в спеціально заданих, штучних умовах*)
18. Наведіть етапи розробки програмного забезпечення.
19. Що включає в себе постановка задачі та передпроектні дослідження?
20. Перерахуйте функціональні та експлуатаційні вимоги до програмному продукту.
21. Перерахуйте правила розробки технічного завдання.( *Технічне завдання являє собою документ, в якому сформульовані основні цілі розробки, вимоги до програмного продукту, визначено терміни та етапи розробки та регламентований процес приймально-здавальних випробувань*.)
22. Назвіть основні розділи технічного завдання.? («*Найменування та область застосування», «Підстава для розробки», «Призначення розробки», «Технічні вимоги до програми або програмного виробу», «Техніко-економічні показники», «Стадії та етапи розробки», «Порядок контролю і приймання»*)
23. Надайте характеристику областей знань SWEBOK.
24. Які національні стандарти розробки програмного забезпечення Ви знаєте?
25. Які міжнародні стандарти розробки програмного забезпечення Ви знаєте?
26. За якими стандартами виконується оцінювання якості програмного забезпечення?
27. Що таке архітектура ПЗ? (*Архітектура ― це фундаментальна організація системи, втілена в компонентах, їх взаємозв'язках, середовищі, і принципах, що управляють їх дизайном і еволюцією. Це множина основних проектних рішень про ПЗ.)*
28. Що таке описова та розпорядча архітектура ПЗ? (*Розпорядча системна архітектура складається з проектних рішень, що були прийняті перед конструюванням ПЗ. Також її можна назвати продуманою чи призначеною архітектурою. Описова архітектура ПЗ визначає те, як система була побудована. Це «реалізована» чи «введена у використання» архітектура*.)
29. Що таке архітектурний дрейф та ерозія? (*Архітектурний дрейф – це введення основних проектних рішень до описової архітектури ПЗ, які не включені в розпорядчу архітектуру, але які не конфліктують з її рішеннями. Архітектурна ерозія – це введення архітектурних рішень до описової архітектури ПЗ, при цьому рішення не повинні порушувати його розпорядчої архітектури.)*
30. Що таке відновлення архітектури? (*Відновлення архітектури – це процес відтворення архітектури на основі компонентів ПЗ, що були отримані на етапі її конструювання.)*
31. Що таке компонент ПЗ? (*це архітектурна одиниця, яка відповідає за функціональність ПЗ та/або зберігання даних, надає доступ до своїх сервісів за допомогою зовнішнього інтерфейсу і має чітко визначену залежність від контексту у якому цей компонент застосовують*)
32. Що таке з’єднувач ПЗ? (*це архітектурна одиниця, що відповідає за здійснення взаємодії між компонентами*)
33. Що таке архітектурна конфігурація або топологія? (*це специфікація з’єднань між з’єднувачами та компонентами*.)
34. Дайте порівняльну характеристику різним типам архітектури програмних систем. Наведіть приклади програм, які використовують ці архітектури.
35. Що таке нефункціональна властивість ПЗ? (*обмеження на те, як ПЗ реалізує і доставляє свою функціональність*)
36. Що таке ефективність ПЗ? (*це якість, яка відображає здатність ПЗ до задоволення вимог продуктивності при одночасній мінімізації використання його ресурсів*)
37. Що таке складність ПЗ? (*вказує до якої міри ПЗ або однин з його компонентів, містить проектні рішення чи реалізацію, які важко зрозуміти і перевірити*)
38. Що таке неоднорідність ПЗ? (*це якість ПЗ, що передбачає, що ПЗ складається з декількох різнорідних компонентів або функціонує в декількох різнорідних обчислювальних середовищах одночасно*)
39. Що таке надійність ПЗ? (*це набір властивостей ПЗ, що дозволяє розраховувати, що ПЗ буде функціонувати так, як було заплановано*.)
40. Що таке масштабованість ПЗ? (*це можливість системи бути зміненою з урахуванням нових вимог*)
41. Що таке адаптованість ПЗ? (*або пристосовуваність – є здатністю ПЗ до задоволення нових вимог і пристосовуватися до нових умов роботи під час його життєвого циклу*)
42. Що таке шаблон проектування? (*шаблон проектування або паттерн – повторювана архітектурна конструкція, що представляє собою рішення проблеми проектування в рамках деякого контексту*)
43. Які групи шаблонів вам відомі? *(Твірні шаблони або породжуючі шаблони, Структурні шаблони, Поведінкові шаблони, Шаблони паралельного програмування, Carrier Rider Mapper, Аналітичні шаблони, Комунікаційні шаблони, Організаційні шаблони, Анті-паттерни)*
44. Опишіть основні характеристики шаблону FAÇADE. *(спрощує використання складної системи, окремої частини системи або забезпечує звернення до системи деяким специфічним чином; надає лише колекцію методів, простих у розумінні та використанні. Клієнту надається спрощений інтерфейс)*
45. Які переваги та недоліки, на Вашу думку, має шаблон проектування FACADE? (*Переваги: Приховує реалізацію підсистеми від клієнтів, що полегшує використання підсистеми. Сприяє слабкій взаємодії між підсистемою та її клієнтами, що дозволяє змінити класи, які включають підсистему, не впливаючи на клієнтів. Зменшує компіляційні залежностей у великих програмних системах. Спрощує системи перенесення на інші платформи, оскільки менш імовірно, що для побудови однієї підсистеми потрібно побудувати всі інші. Недоліки: Не заважає сучасним клієнтам отримувати доступ до базових класів. Фасад не додає жодної функції, він просто спрощує інтерфейси*)
46. Опишіть основні характеристики шаблону Adapter.( *призначений для організації використання функцій об'єкту, недоступного для модифікації, через спеціально створений інтерфейс. Адаптує інтерфейс одного класу в інший, очікуваний клієнтом. Адаптер забезпечує роботу класів з несумісними інтерфейсами, та найчастіше застосовується тоді, коли система підтримує необхідні дані і поведінку, але має невідповідний інтерфейс. Шаблон Адаптер дозволяє включати вже існуючі об'єкти в нові об'єктні структури, незалежно від відмінностей в їхніх інтерфейсах.)*
47. Які переваги та недоліки, на Вашу думку, має шаблон проектування Adapter? (*Переваги: Допомагає досягти багаторазового використання та гнучкості. Клас клієнта не ускладнюється необхідністю використовувати інший інтерфейс і може використовувати поліморфізм для обміну між різними реалізаціями адаптерів. Недоліки: Всі запити пересилаються, тому спостерігається незначне збільшення накладних витрат. Іноді багато адаптацій потрібні по ланцюгу адаптера, щоб досягти потрібного типу.*)