

# Інженерія програмного забезпечення тести

## 1-рівень

### 1. Що таке програмна інженерія?

Це застосування системного, вимірюваного підходу до розробки, використання та супроводу програмного забезпечення

### 2. Що таке вимоги до програмної інженерії?

Це властивості, якими має володіти програмне забезпечення для адекватного визначення функцій, умов та обмежень, а також обсягів даних

### 3. Що таке архітектура програмного забезпечення?

Це представлення системи програмного забезпечення, що дає інформацію про компоненти та взаємозв'язки

### 4. Що таке компонент в програмній інженерії?

Абстрактна одиниця інструкцій програмного забезпечення і внутрішніх станів, яка забезпечує трансформацію даних через свої інтерфейси

### 5. Що таке супроводження програмного забезпечення?

Процес покращення, оптимізації та виправлення дефектів у програмному забезпеченні після його вводу до експлуатації

### 6. Що таке реінженерія?

Це удосконалення застарілого програмного забезпечення шляхом його реорганізації або реструктуризації

### 7. Що таке рефакторинг?

Це реорганізація коду для поліпшення характеристик показників якості об'єктно-орієнтовний і компонентних програм без зміни їх поведінки

### 8. Що таке Аутсорсинг?

Передача розробки програмних робіт...

### 9. Що таке життєвий цикл програмного забезпечення?

Сукупність окремих етапів робіт...

### 10. Яка основна суть каскадної моделі?

Етапи залежать один від одного...

### 11. Що таке ітераційна модель?

Передбачає розбиття проекту...

### 12. Що таке спіральна модель?

Усі етапи життєвого циклу...

**13. Що таке інкрементна модель?**

Основний принцип моделі...

**14. Що таке Agile?**

Набір принципів гнучкої...

**15. Яка основна ідея підходу Lean?**

Ощадливо (бережно) ставитись..

**16. Що таке Scrum?**

Методологія ґрунтується....

**17. Що таке конструювання програмного забезпечення?**

Описує детальне створення робочої програмної системи за допомогою комбінації кодування, верифікації, модульного тестування, інтеграційного тестування та відлагодженн

**18. Що таке Kanban?**

Проект ділиться на етапи...

**19. Що таке RUP (Rational Unified Process)?**

Розробка продукту за даним...

**20. Ефектні способи вирішення задач проєктування програмного забезпечення - це :**

Шаблони проєктування програмного...

**21. Скільки існує патернів програмування?**

**23**

**22. Що таке DSDM (Dynamic Systems Development Model)?**

Методологія що демонструє...

**23. Що таке XP (Extreme Programming)?**

Методологія орієнтована...

**24. Що таке об'єктно-орієнтоване програмування?**

Парадигма програмування...

**25. Що таке абстрагування?**

Це спосіб виділити...

**26. Що таке об'єкт?**

Це сутність, екземпляр класу...

27. Що таке інкапсуляція?

Це властивість системи...

28. Що таке наслідування?

Це властивість системи, що дозволяє описати новий клас на основі вже існуючого з частковою або повністю запозиченою функціональністю

29. Що таке поліморфізм?

Це властивість системи використовувати

30. Скільки існує модифікаторів доступу?

4

31. Що таке інтерфейс?

Сукупність засобів...

32. Що таке функціональне програмування?

Це стиль написання...

33. Сукупність документів, що містять відомості необхідні для розробки, виготовлення, супроводу та експлуатації програм це?

Програмна документація

34. Що таке UML?

Уніфікована мова програмування...

35. Що таке діаграма класів?

Основний спосіб опису....

36. Скільки існує типів бінарних зв'язків?

3

37. Що таке верифікація?

Це підтвердження відповідності...

38. Що таке валідація?

Підтверджує що вимоги замовника...

39. Що таке тестування програмного забезпечення?

Перевірка відповідності...

40. Які три парадигми ООП?

Інкапсуляція, поліморфізм...

41. Що таке дефект, баг (defect, bug)?

Недолік компонента...

42. Що таке якість програмного забезпечення?

Це сукупність характеристик...

43. Від чого залежить якість програмного забезпечення?

Продуманої архітектури...

44. Що таке тест-кейс?

Документ, що містить набір вхідних....

45. Яка з команд створить копію (клонує) існуючий каталог (репозиторій) на особистий комп'ютер?

`git clone`

46. Яка з команд визначить стан файлів в Git?

`git status`

47. Що таке компоненти повторного використання (КПВ)?

Це готові компоненти...

48. Яка з команд добавить файл test.txt під контроль git?

`git add test.txt`

49. Яка з команд відправить проєкт у веб репозиторій (наприклад на github):

`git push -u...`

50. Що таке Git?

Розподілена система... (версій)

## 2-рівень

1) (дві відповіді з п'яти) Тестування - це одна з технік контролю якості, що включає в себе активності з планування робіт, виконання тестування і також:

збільшення складності

проектування тестів

стандарти програмування

конструювання

аналіз отриманих результатів

2) Які є найпоширеніші види патернів??

Породжуючі

Абстрактні

Фабрика

Структурні

Поведінкові

Прості

Складні

3)

4)

5) Відповідно до структурного програмування будь яку програму можна створити використовуючи три конструкції?

Розгалуження

Об'єкти

Фабрика

Послідовність виконання

Цикл

Динамічний поліморфізм

Складні структури об'єктів

6) Поліморфізм можна грубо розділити на?

Абстрактний поліморфізм

Статичний поліморфізм

Динамічний поліморфізм

Не динамічний поліморфізм

Поведінковий поліморфізм

7) Існує 4 види модифікаторів доступу: public, privat а також:

Init

Protected

Default

None

Note privat

8) Для класів можна використувати тільки два модифікатори доступу:

Public

Privat

Default

PIP

UDALYTY

9) Зазвичай до етапів життєвого циклу відносять такі етапи: аналіз вимог, проектування, програмування, експлуатація:

Тестування і налагодження

Продаж

Маркетинг

Супровід і підтримку

Model-View-View

10) Найвідоміші моделі життєвого циклу: каскадна модель, ітераційна а також:

Спиральна

Класи сервісів

Інкриментна

Показова

Модель розгляду

11) Назвіть дві основні гнучкі методології розробки програмного забезпечення:

Scrum

Status kod

Service

Kanban

MVS

12) Ілюстративними засобами вираження характеристик ПЗ в архітектурному проєкті використовуються ER-діаграми, DFD-діаграми, макети:

UML-діаграми

Скорочення затрат

Блок-схеми

Важка і складна інтеграція

Картинки

13) Вимоги до ПЗ поділяють на:

Системні

Інтерфейсні

Функціональні

Не функціональні

Складні

Структуризовані

Прості

14) Для Git системи файли в робочому каталозі діляться на:

Складний файл

Відслідковуючі

Файл конструктора

Не відслідковуючі

Файл технології та бізнес файл

15) Виберіть три найважливіші парадигми ООП:

Поліморфізм

Інтерфейс

Наслідування

Інкапсуляція

Індикація

Структуризація

Простота

16) **Вимоги** - це властивості якими має володіти ПЗ для адекватного визначення функції, умови і обмежень виконань ПЗ.

17) Зразок проектування- це **шаблон** рішення часто зустрічається і який можна використовувати всякий раз, коли ця задача виникає.

18) Компоненти конструюються самостійно, як деяка абстракція, що містить у собі **інформаційну** частину й артефакти.

19) **Верифікація**- це підтвердження відповідності кінцевого продукту визначним еталонним вимогам.

20) **Валідація** підтверджує, що вимоги замовника, послуги або системні задоволення.

21) Супроводження **програмного** забезпечення - це процес покращення, оптимізації та виправлення дефектів і програмному забезпеченні після його вводу в експлуатацію.

22) Реферсна **інженерія** полягає у відновленні специфікацій (графів викликів, потоків даних) за отриманим кодом системи для її аналізу на більш високому рівні.

23) Реорганізація коду для поліпшення їх характеристик і показників якості об'єктно-орієнтованих і компонентних програм без зміни їх поведінки називається **рефакторингом**.

24)

25) TCP - інтеракційна **модель** передбачає розбиття проекту на частини (етапи, інтеракції) і проходження етапів життєвого циклу на кожному з них.

26) Ідея підходу LEAN полягає в тому, що ми ощадливо ставимось до **ресурсів**.

27) Scrum методологія ґрунтується на понятті **спринту** протягом якого виконується робота над проектом.

28) За Kanban методології проект ділиться на етапи, що візуалізуються у вигляді канбан-**дошки**.

29) **Наслідування**- це процес за допомогою якого один об'єкт може набувати властивості іншого.



30) Сутність, екземпляр класу, які можна посилати повідомлення і яка може на них реагувати, використовуючи свої дані називається **об'єкт**.

### 3-рівень

1) Ефектність способу вирішення задач проектування програмного забезпечення - **шаблони проектування**

Описує детальне створення робочої програмної системи за допомогою комбінації кодування, верифікації (перевірки), модульного тестування, інтеграційного тестування та підлагодження - **конструювання програмного забезпечення**

Це застосування системного, вимірювального підходу до розробки, використання та супроводу програмного забезпечення - **програмна інженерія**

Одна з парадигм програмування, яка розглядає програму як множину об'єктів, що взаємодіють між собою - **ООП**

#### Інкапсуляція

2) Механізм який поєднує дані та методи, що обробляють ці дані і захищає і те і інше від зовнішнього впливу або не вірного використання - **інкапсуляція**

Процес завдяки якому один об'єкт може придбати властивості, тобто наслідувати властивість іншого об'єкта і додавати риси характерні тільки для нього самого - **наслідування**

Властивість, яка дозволяє одне і те саме ім'я використовувати для вирішення декількох технічно різних задач - **поліморфізм**

#### Абстракція

3) Це удосконалення застарілого програмного забезпечення шляхом його реорганізації або реструктуризації - **реінженерія**

Це реорганізація коду для поліпшення характеристик показників якості об'єктивно-орієнтований і компонентних програм без змінни їх поведінки - **рефакторинг**

Передача розробки програмних робіт інші допоміжні компанії - **аутсорсинг**

Абстрактна одиниця інструкції ПЗ внутрішніх станів, яка забезпечує трансформацію даних через свої інтерфейси - **компонент**

Процес покращення, оптимізації та виправлення дефектів у ПЗ після його вводу до експлуатації - **супровід ПЗ**

### **Поліморфізм**

4) Сукупність окремих етапів робіт, що проводяться у заданому порядку протягом періоду часу, який починається з вирішення питання про розроблення ПЗ і закінчується припиненням використання ПЗ вказаного ресурсу - **життєвий цикл ПЗ**

Це представлення системи ПЗ, що дає інформацію про компоненти та взаємозв'язки - **архітектура ПЗ**

Це властивості, якими має володіти ПЗ для адекватного визначення функцій, умов та обмежень, а також обсягів даних - **вимоги до ПЗ**

Уніфікована мова моделювання, використовується у парадигмі об'єкто-орієнтованому програмуванні, використовують графічні позначення для створення абстрактної моделі системи - **UML**

### **OPTION**

5) Розподілена система керування версіями файлів та спільної роботи, проект створив Лінус Торвальд = **GIT**

Недолік компонента або системи який може привести до відмови певної функціональності = **Bug**

Загальна назва UNIX-подібних операційних систем на основі одноіменного ядра = **Linux**

Комерційне інтегроване середовище розробки для різних мов програмування (Java, Python, Scala, Php, та інші) від компанії JetBrains = **IDEA**

### **GRUD**

6) Підтверджує, що вимоги замовника (користувача), послуги або системи задоволені = **Валідація**

Це підтвердження відповідності кінцевого продукту визначеним еталонним вимогам = **Верифікація**

Перевірка відповідності між реальною та очікуваною поведінкою програми, що здійснюється на кінцевому наборі тестів, обраному певним чином = **Тестування ПЗ**

## **Абстракція**

7) Це сукупність характеристик ПЗ, що відображають його здатність задовільняти встановлені і передбачувані потреби = **Якість ПЗ**

Документ, що містить набір вхідних значень, а також очікуваний результат проведення тесту, розроблених для перевірки відповідності вимогам = **Тест-кейс**

Це готові компоненти, елементи оформлених знань, що використовуються у ході розроблення програм = **Компоненти повторного використання**

Це стиль написання програм через складання набору функцій = **Функціональне програмування**

## **XML**

8) Основний принцип моделі має розширення можливостей, добування модулів і функцій програми, добавляють нові версії = **Інкремента модель**

Усі етапи життєвого циклу моделі йдуть витками на кожному з яких відбувається проектування, кодування, дизайн, тестування і так далі = **Спіральна модель**

Передбачає розбиття проекту на частини (етапи-ітерації) і проходження етапів життєвого циклу на кожному з них = **Ітераційна модель**

Етапи залежать один від одного і наступний починається тоді, коли завершиться попередній = **Каскадна модель**

## **ООП**

9) Набір принципів гнучкої розробки (всього 12) та ідей ПЗ = **Agile**

Ощадливо (бережно) ставитись до ресурсів особливого часу і вирішувати завдання найпростішими способами = **Lean**

Методологія ґрунтується на понятті спринту (Sprint) протягом якого виконується робота над продуктом = **Scrum**

Проект ділиться на етапи, що візуалізуються у вигляді дошки = **Kanban**

## **Server**

10) Розробка продукту за даним методом складається з 4 фаз (початкова стадія, уточнення, побудова, впровадження) = **RUP**

Методологія що демонструє набір принципів, визначених типів ролей і технік = **DSDM**

Методологія орієнтована на постійну зміну вимог до продукту, пропонує 12 підходів для досягнення ефективних результатів у подібних умовах = **XP**

Це спосіб виділити набір значущих характеристик об'єкта, виключаючи з огляду незначимі = **Абстрагування**

## **OPTION**

11) Сукупність засобів методів і правил взаємодії (управління контрою і так далі) між елементами системи = **Інтерфейс**

Це сутність екземпляр класу, і який можна посилати повідомлення і яка може на них реагувати, використовувати свої дані = **Об'єкт**

Багаторазове виконання операції, або група операцій за якоїсь умови = **Цикл**

Основний спосіб опису статичної структури системи = **Діаграма класів**

## **Функція**

12) Створити копію (клонувати) існуючий каталог (репозиторій) в особистий комп'ютер = **git clone**

Ініціалізувати репозиторій = **git init**

Створити коміт = **git commit-em "add commit"**

Добавити в GIT якийсь файл чи проект = **git add**

**git pull**

13) Шаблони, форма, напрацьований ефектний підхід до створення програмного продукту = **Патерн**

Втілення абстрактної моделі окремої сутності (предмету або поняття), що має чітко виражене функціональне призначення в деякій області = **Об'єкт**

Спеціальна конструкція, яка використовується для групування пов'язаних змінних та функцій = **Клас**

Функція або процедура, що належить якомусь класу чи об'єкту = **Метод**

**Змінна**