

**5A330601-Дастурий инжиниринг мутахассислиги бўйича магистратурага кирувчилар
учун имтихон саволлар**

Дастурий таъминотни тестлаш ва текшириш

1. Chegaraviy ma'lumotlar
2. Chegaraviy shartlarni tekshirish
3. Dastur kodini formal inspeksiyalash.
4. Dastur kodini qoplanganligi bo'yicha hisobotlar
5. Dastur kodini testlash usullari
6. Dastur kodini testlashni maqsad va vazifalari
7. Dastur kodini verifikatsiyalash bosqichlari
8. Dasturiy ta'minot sifati tushunchasi va tavsiflari
9. Dasturiy ta'minot validasiyasi
10. Dasturiy ta'minot verifikatsiyasi
11. Dasturiy ta'minot yaratishning modellari
12. Dasturiy ta'minot yaratishning hayot sikli
13. Dasturiy ta'minotni testlash asosiy tushuncha va ta'riflar
14. Dasturiy ta'minotni testlash turlari
15. Dasturiy ta'minotni testlashning hayot sikli
16. Dasturiy taminotni testlash uchun test misollarini shakillantirish:
17. Dasturning boshqaruv grafi
18. Loyihaning hayot siklida yaratiladigan xujjatlar va ularning vazifalari.
19. Ma'lumotlarni qayta kiritish. Noto'g'ri ma'lumotlar
20. Mezonlar sinfi
21. Modul tushunchasi va uning bosqichlari
22. Modulni testlashning maqsadi va vazifalari
23. Rasmiy inspeksiya bosqichlari va undagi ishtirokchilar roli
24. Rasmiy inspeksiya jarayonini hujatlashtirish
25. Rasmiy inspeksiyalarni o'tkazishning maqsad va vazifalari.
26. Ruxsat e'tilgan ma'lumotlar
27. Sinf (klass)larni testlash.
28. Test holatlari bo'yicha jadvallar
29. Test misollarini ketma-ketligini optimallashtirish
30. Test muhitini sozlash
31. Test muxiti
32. Test rejalar.
33. Test talab xossalari
34. Test talablarining loyiha xujjatlaridagi o'rni.
35. Test talablarining loyihaning boshqa turdagi hujjatlar bilan aloqasi
36. Testda o'tganlik bo'yicha hisobotlar
37. Testlarni tanlash mezoni
38. Testlashni takrorlanib turishi: maqsad va vazifalari
39. Testni bajarish uchun boshlang'ich shartlar
40. Verifikatsiyalash strategiya va rejalar.
41. Verifikatsiyalash va testlash jarayonlari hujatlari
42. Verifikatsiyalashning texnologik jarayonlari, uning loyihadagi o'rni
43. Dastur xatoliklari
44. Muammolarni nazorat qilish tizimlari
45. Testlarni ishlab chiqish
46. Adaptiv testlash
47. Hujjatlarni testlash
48. Testlovchining qurollar vositasi

49. Testlash guruhni boshqarish
50. Keng tarqalgan dasturiy xatoliklar

Объектга йўналтирилган дастурлаш фанидан саволлар

1. Дастурлашда ўзгарувчилар ва ўзгармаслар. Мисоллар ёрдамида тушунтиринг
2. Дастурлашда маълумотлар типлари. Ҳар бир типга изоҳ беринг
3. Дастурлашда цикл операторлари. Мисоллар ёрдамида тушунтиринг
4. Дастурлашда танлаш оператори. Мисоллар ёрдамида тушунтиринг
5. Дастурлашда инкремент ва декремент операторлар. Мисоллар ёрдамида тушунтиринг
6. Массивлар билан ишлаш. Мисоллар ёрдамида тушунтиринг
7. Тартибсиз (Нерегулярный) массивлар. Мисоллар ёрдамида тушунтиринг
8. Объектга йўналтирилган дастурлашни мисоллар ёрдамида тушунтиринг
9. Объектга йўналтирилган дастурлашда объектлар
10. Объектга йўналтирилган дастурлашда синфлар
11. Объектга йўналтирилган дастурлашда конструктор
12. Объектга йўналтирилган дастурлашда конструкторнинг хусусиятлари
13. Объектга йўналтирилган дастурлашда синф элементлари
14. Дастурлашда методлар. Мисоллар ёрдамида тушунтиринг
15. Синф элементларига мурожат ҳуқуқлари. Мисоллар ёрдамида тушунтиринг
16. Объектга йўналтирилган дастурлашда объектларни эълон қилиш ва улардан фойдаланиш
17. Метод прототипи. Мисоллар ёрдамида тушунтиринг
18. Методнинг қиймат қайтариши. Мисоллар ёрдамида тушунтиринг
19. Методларда параметрлар. Мисоллар ёрдамида тушунтиринг
20. Метод параметрларига қиймат узатиш. Мисоллар ёрдамида тушунтиринг
21. Method overloadingни мисоллар ёрдамида тушунтиринг
22. Method overridingни мисоллар ёрдамида тушунтиринг
23. Синфларда статик методлар ва унга мурожат. Мисол келтиринг
24. This кўрсаткичини мисоллар ёрдамида тушунтиринг
25. Сатрлар билан ишлаш
26. String синфи конструкторлари. Мисоллар ёрдамида тушунтиринг
27. Сатрлар билан ишлайдиган методлар. Мисоллар ёрдамида тушунтиринг
28. Объектга йўналтирилган дастурлаш тамойилларини сананг ва таъриф беринг
29. Объектга йўналтирилган дастурлашда ворислик
30. Объектга йўналтирилган дастурлашда ворислик турлари
31. Синф конструкторларида ворислик
32. Объектга йўналтирилган дастурлашда полиморфизм
33. Объектга йўналтирилган дастурлашда инкапсуляция
34. Math синфининг методлари. Мисоллар ёрдамида тушунтиринг
35. Объектга йўналтирилган дастурлашда “super” калит сўзи
36. Объектга йўналтирилган дастурлашда “extends” калит сўзи
37. Объектга йўналтирилган дастурлашда “implements” калит сўзи
38. Объектга йўналтирилган дастурлашда abstract синфлар
39. Объектга йўналтирилган дастурлашда interfacелар
40. Объектга йўналтирилган дастурлашда packagелар
41. Объектга йўналтирилган дастурлашда шаблонлар (template)
42. Дастурлашда файллар билан ишлаш
43. Объектга йўналтирилган дастурлашда дўст синфлар
44. Дастурлашда стандарт кутубхоналар
45. Дастурлашда хатolikлар. Истисноларни бошқариш
46. Дастурлашда фойдаланувчи график интерфейсини яратиш
47. Дастурлашда виртуал функциялар
48. Дастурлашда изоҳлар билан ишлаш
49. try ва catch блокидан фойдаланиш

50. throw, throws, finally операторлари
51. Дастурлашда операторларни қайта юклаш
52. Объектга йўналтирилган дастурлашда деструктор

Дастурий инжинирингга кириш

1. Унификацияланган моделлаштириш тили (UML) диаграммасини қисқача тарифланг.
2. SWEBOOK нима ва бу ўрганилаётган курсга қандай алоқаси мавжуд.
3. Қайта фойдаланишни режалаштиришда қандай факторлар эътиборга олинади?
4. Дастурий таъминот ишлаб чиқишни бошқаришда хавфлар идентификацияси қандай амалга оширилади?
5. Реал вақт ва Клиент-сервер тизимлари архитектуралари солиштирма характеристикаларини келтиринг.
6. Формал спецификацияни таърифини келтиринг.
7. Дастурий таъминотдан қайта фойдаланишга йўналтирилган лойиҳалаштириш авзалликларини келтиринг.
8. Фойдаланувчи интерфейсини лойиҳалаштириш хусусиятларини келтиринг.
9. Таксимланган ахборот тизимларини яратишда қандай масалалар ҳал қилинади?
10. Дастурий таъминот ишлаб чиқиш ҳаётий цикли (ХЦ). ХЦ асосий босқичлари, машхур моделларини кўрсатинг.
11. Дастурий таъминот прототиплаш мақсад ва вазифалари. Прототиплаш турлари ва улар орасидаги фарқлар.
12. Дастурий таъминот архитектураси таърифи. Мисоллар келтиринг.
13. Дастурий таъминот компоненталаридан файта фойдаланиш авзалликларини ва камчиликларини санаб ўтинг.
14. Хавфларни бошқариш жараёни таркибини келтиринг ва қисқа тавсифини келтиринг.
15. Таксимланган тизимлар нима? Таксимланган тизимларни лойиҳалаштиришда қандай муаммолар келиб чиқади?
16. Лойиҳа, маҳсулот ва бизнеснинг умумий хавфларига мисоллар келтиринг.
17. Фойдаланувчи интерфейсини лойиҳалаштириш хусусиятларини кўрсатинг.
18. Реал вақт тизимлари ва ўрнатилган тизимлар ишлаш принципларини тушунтиринг.
19. Дастурий таъминот талабларини шакллантиришни маъсул шахсларни ролларини айтинг.
20. Экстремал дастурлаш услугиёти асосий характеристикалари.
21. Лойиҳалаш эскизлари нима? Мисоллар келтиринг.
22. Фойдаланувчи интерфейси асосий турлари ва лойиҳалаш хусусиятларини асосий кўринишларини тавсифланг.
23. Дастурий таъминотдан қайта фойдаланишга йўналтирилган лойиҳалаш муаммоларини баён этинг.
24. Дастурий таъминот статик ва динамик моделлаштириш жараёнлари қандай фарқ қилади?
25. Дастурий таъминотни функционал бўлмаган талаблар тушунчасига таъриф беринг.
26. Фойдаланувчи интерфейсини тестлаш хусусиятларини келтиринг.
27. Дастурий таъминотни ишлаб чиқишда ишлаб чиқишни бошқаришда қандай хавфлар мавжуд
28. Дастурий воситалар ишончилиги қандай характеристикалар билан аниқланади?
29. Дастурий таъминот архитектурасини статик ва динамик моделлаштириш жараёнлари қандай қилади?
30. Дастурий таъминотга функционал бўлмаган талаблар тушунчасини таъриф беринг.
31. Дастурий таъминот яратишда хавфларни режалаштириш қандай амалга оширилади?
32. Қайта фойдаланишни кўллаб-қувватловчи асосий ёндашувларни санаб ўтинг.
33. Дастурий тизимлар статик таҳлили таркибини тавсифланг.
34. “Моҳият-боғланиш” моделининг асосий характеристикаларини айтинг. ER-модели тавсифига мисол келтиринг
35. СММ моделининг қандай даражасидаги корхоналарда “тестловчи” мутахассис ҳисобланади ва нима учун?

36. Архитектуралар лойиҳалаш қачон бошланади? “Тизимнинг контекстли модели” терминини қандай тушунасиз?
37. Дастурий таъминот ҳаётий цикли ва эволюцион модели асосий устунликларини айтинг.
38. Дастурий таъминот эволюцияси жараёни қандай босқичлардан иборат?
39. ДТ хавфсизлигини таъминлаш жараёнларига кирувчи фаолиятлар турларини санаб беринг.
40. Дастурий таъминотни ишлаб чиқиш каскад моделини тавсифлаб беринг.
41. Дастурий таъминотни лойиҳалаштиришда хавфлар таҳлили процедурасини баён қилинг.
42. Қайта фойдаланишга йўналтирилган дастурий таъминотни ишлаб чиқиш қандай турлари мавжуд? Уларнинг қисқача тавсифланг.
43. Иловалар тизимидан қайта фойдаланишга мисоллар келтиринг.
44. Дастурий таъминот ҳаётий цикли спирал ва каскадли моделлари муҳим фарқларини тавсифлаб беринг?
45. Компоненталардан қайта фойдаланишга мисоллар келтиринг.
46. Талабларни қўйиш учун маълумотлар манбаларини тавсифлаб беринг.
47. Декомпозиция нима ва дастурий инжинирингда нима учун қўлланилади?
48. Ишончилиқни ўлчаш муаммоларини характерланг.
49. Иловалар каркаслари (frameworks) нима? Мисоллар келтиринг.
50. Дастурларни визуал лойиҳалаштириш учун UML тилида қандай диаграммалар мавжуд? Сизга маълум диаграммаларга қисқача тавсиф беринг.
51. Дастурий таъминотни лойиҳалаштириш асосий концепциясини тасвирланг.
52. Объектга йўналтирилган лойиҳалашни асосий концепциясини санаб ўтинг. Уларнинг қисқача тавсифини келтиринг.
53. SCRUM услубиятини устунликлари ва камчиликларини тасвирланг.
54. WAF (Web application frameworks) асосий хусусиятларини тавсифланг.
55. ДТ ни лойиҳалашда рисклар индикаторини характерланг ва санаб ўтинг.
56. “Юмшоқ” ва “қаттиқ” реал вақт тизимлари орасидаги фарқларни тасвирланг.
57. ДТ ўзгаришига тезкор сўровлар нима билан асосланган?
58. Реал вақт тизимлари ДТ ни лойиҳалашгт асосий концепциясини санаб ўтинг. Уларнинг қисқача тавсифини келтиринг.
59. ДТ ни лойиҳалашда инкапсуляция қандай ва нима учун қўлланилади?
60. User interface лойиҳалашнинг асосий принципларини тасвирланг.

Дастурий таъминотни тестлаш ва текшириш

1. Граничные данные
2. Проверка граничных условий
3. Формальная инспекция программного кода
4. Отчеты по покрытию программного кода
5. Методы тестирования программного кода
6. Цели и задачи тестирования программного кода
7. Этапы верификации программного кода
8. Понятия качества программного обеспечения и его описание
9. Валидация программного обеспечения
10. Верификация программного обеспечения
11. Модели создания программного обеспечения
12. Жизненный цикл создания программного обеспечения
13. Основные понятия и определения тестирования программного обеспечения
14. Виды тестирования программного обеспечения
15. Жизненный цикл тестирования программного обеспечения
16. Формирования тестовых примеров для тестирования программного обеспечения
17. Граф управления программы
18. Документы создаваемые в жизненном цикле проекта и их задачи
19. Повторный ввод данных. Неправильные данные

20. Класс критерий
21. Понятия и этапы модуля
22. Цели и задачи тестирования модуля
23. Этапы официальной инспекции и роль участников
24. Документирование процесса официальной инспекции
25. Цели и задачи проведения официальной инспекции
26. Допустимые данные
27. Тестирование классов
28. Таблицы по состоянию тестов
29. Оптимизация последовательности тестовых примеров
30. Настройка тестовой среды
31. Тестовая среда
32. Тестовые планы
33. Свойства тестовых спросов
34. Роль тестовых спросов в документации проекта
35. Связь тестовых спросов с другими типами документов проекта
36. Отчёты о прохождении теста
37. Критерии выбора тестов
38. Повторение тестов: цели и задачи
39. Начальные условия для выполнения тестов
40. Стратегии и планы верификации
41. Документы процесса верификации и тестирования
42. Технологические процессы верификации и роль в проекте
43. Программные ошибки
44. Система отслеживания проблем
45. Разработка тестов
46. Адаптационное тестирование
47. Тестирование документации
48. Инструментальные средства тестировщика
49. Управление группой тестирования
50. Распространенные программные ошибки

Вопросы по ООП

1. Назовите семь основных принципов ООП и прокомментируйте, как они использованы.
2. Что такое дружественные функции и дружественные классы? В каких случаях их использование целесообразно?
3. Что такое переопределение операций? Какие операции можно переопределять?
4. Чем отличаются компонентные и внешние функции-операторы?
5. Зачем нужны перегружаемые функции?
6. Что такое виртуальный деструктор и каковы особенности его использования?
7. Дайте определения понятия «поток». Чем отличаются текстовые и бинарные файлы?
8. Назовите основные классы потоков. Объектами каких классов являются потоки `cin` и `cout`?
9. Каково назначение деструктора? Укажите формат описания деструктора. Назовите варианты вызова деструктора.
10. Что определяют флаги состояния потока? Какие имеются флаги и переменные форматирования? Зачем и как анализируют флаги состояния потока?
11. В чем смысл перегрузки функций и на чем основан механизм перегрузки функций?
12. Приведите формат записи операторной функции. Укажите полную и сокращенную формы вызова операторной функции.
13. Укажите ограничения, накладываемые на перегрузку стандартных операций.
14. В каком файле объявлены классы файловых потоков? Чем отличается

форматированный ввод-вывод от строко-ориентированного и символного ввода-вывода?

15. Что общего между форматированным и строко-ориентированным вводом-выводом?

16. Укажите назначение операторов `try`, `catch` и `throw`.

17. Каково назначение пространства имен?

18. Что понимается под термином «технология визуального программирования»?

19. Перечислите отличия виртуальных и перегружаемых функций.

20. Поясните понятия статического и динамического полиморфизма.

21. Каковы различия модульного и объектно-ориентированного программирования?

22. Назовите виды полиморфизма в C++. Определите понятие виртуальных и абстрактных функций.

23. Что такое абстрактный класс? Назовите особенности использования абстрактного класса.

24. Каким образом осуществляется доступ к элементам данных и методам класса или объекта?

25. Укажите важнейшие факторы, влияющие на эволюцию методов проектирования и создания программных продуктов.

26. В чем суть наследования? Какие разновидности наследования имеются в языках программирования.

27. Каковы особенности одиночного и множественного наследования? Как описывается производный класс? Что такое множественное и виртуальное наследование?

28. Укажите, для чего предназначена типизация данных. Что дает использование структурных типов данных?

29. Назовите основные этапы разработки программных систем с использованием ООП и расскажите о каждом из них.

30. Охарактеризуйте свойства инкапсуляции.

31. Чем отличается описание компонентных функций внутри и вне определения класса?

32. Как соотносятся между собой и что означают понятия класс объекта и объект?

33. Какими факторами определяется целесообразность применения технологии ООП?

34. Назовите способы установления связи и обмена информацией между объектами.

Предложите вариант иерархии в области транспортных средств.

35. Что такое объект и каким образом объекты соединяются в систему для решения задачи? Чем характеризуется объект? Какого назначения зарезервированное слово `this`?

36. Назовите операции, которые могут быть выполнены над программными объектами.

37. Назовите варианты размещения описаний методов класса. В чем заключается достоинство внешнего описания методов класса?

38. Какие функции имеют доступ к собственным элементам данных класса?

39. Какие сложности возникают при работе с динамическими объектами?

40. В чем основное преимущество использования шаблонов; в чем их недостатки?

41. Что такое форматированный ввод-вывод? Какие средства используются для управления форматированным вводом-выводом?

42. Когда требуется явно разрывать связь потока с файлом?

43. Что случится, если несколько обработчиков соответствуют типу сгенерированного объекта?

44. Что происходит с автоматическими объектами, которые были созданы в блоке `try`, когда этот блок генерирует исключение?

45. Как можно написать обработчик `catch`, чтобы обрабатывать родственные типы ошибок без использования наследования классов исключений?

46. Какой тип указателя надо использовать в обработчике `catch`, чтобы перехватывать любое исключение типа указатель?

47. Как программист ограничивает типы исключений, которые могут генерироваться в функции?

48. Что происходит, если функция генерирует исключение типа, не допускаемого спецификацией исключений этой функции?

49. В чем смысл виртуальных функций? Как осуществляется вызов виртуальных функций?

50. Как можно написать обработчик `catch`, чтобы обработать родственные типы ошибок

без использования наследования классов исключений?

51. В каком случае обращение к компонентам объекта можно выполнить без указания имени класса?

52. Что такое копирующий конструктор? Назовите случаи, когда использование такого конструктора обязательно.

Введение в программную инженерию

1. Опишите кратко диаграммы унифицированного языка моделирования (UML).
2. Что такое SWEBOOK, и какое отношение это имеет к изучаемому курсу.
3. Какие факторы принимаются во внимание при планировании повторного использования.
4. Как осуществляется идентификация рисков при управлении разработкой ПО?
5. Приведите сравнительную характеристику архитектур систем реального времени и клиент-сервер.
6. Дайте определение формальной спецификации.
7. Опишите преимущества проектирования с повторным использованием ПО.
8. Опишите особенности проектирования интерфейса пользователя.
9. Какие задачи решаются при создании распределенных информационных систем?
10. Жизненный цикл разработки ПО. Перечислите основные этапы, назовите известные модели ЖЦ.
11. Каковы цели и задачи прототипирования ПО? Какие типы прототипирования вы знаете, в чем их существенное различие?
12. Сформулируйте определение архитектуры ПО. Приведите примеры.
13. Перечислите достоинства и недостатки повторного использования компонентов ПО. Дайте краткие пояснения.
14. Перечислите составляющие процесса управления рисками и дайте их краткую характеристику.
15. Что такое распределенная система? С какими проблемами сталкиваются разработчики при проектировании распределенных систем?
16. Приведите примеры общих рисков проекта, продукта и бизнеса.
17. Опишите особенности проектирования интерфейса пользователя.
18. Объясните принципы работы систем реального времени и встроенных систем.
19. Назовите роли действующих лиц формирования требований к ПО.
20. Опишите основные характеристики методологии экстремального программирования.
21. Что такое паттерны проектирования? Приведите примеры.
22. Опишите основные виды пользовательского интерфейса и особенности его проектирования.
23. Опишите проблемы проектирования с повторным использованием ПО.
24. Чем отличаются процессы статического и динамического моделирования архитектуры ПО
25. Определите понятие нефункциональные требования к ПО.
26. Опишите особенности тестирования пользовательского интерфейса.
27. О каких рисках идет речь при управлении разработкой ПО?
28. Какими характеристиками определяется надежность программных систем.
29. Чем отличаются процессы статического и динамического моделирования архитектуры ПО
30. Определите понятие нефункциональные требования к ПО.
31. Как осуществляется планирование рисков при создании ПО?
32. Перечислите основные подходы, поддерживающие повторное использование.
33. Опишите составляющие статического анализа программных систем.
34. Назовите основные характеристики модели «Сущности-связи». Приведите пример описания ER-модели.
35. В предприятиях какого уровня зрелости модели CMM «тестировщик» считается профессией и почему?
36. Расскажите об особенностях объектно-ориентированного проектирования ПО.
37. Когда начинается архитектурное проектирование? Как вы понимаете термин

“Контекстная модель системы”?

38. Дайте определение спецификации требований. Приведите примеры.
39. Назовите основные преимущества эволюционной модели ЖЦ разработки ПО.
40. Из каких этапов состоит процесс эволюции ПО?
41. Перечислите виды деятельности, входящие в процессы обеспечения безопасности

ПО.

42. Охарактеризуйте каскадную модель ЖЦ разработки ПО.
43. Опишите процедуру анализа рисков при проектировании ПО.
44. Какие виды разработки ПО на основе повторного использования вам известны? Дайте

их краткое описание.

45. Приведите примеры повторного использования системы приложений.
46. Опишите процесс эволюции ПО.
47. Опишите существенные отличия каскадной и спиральной моделей ЖЦ ПО?
48. Приведите примеры повторного использования компонентов.
49. Охарактеризуйте источники сведений для задания требований.
50. Опишите подходы, которые поддерживают повторное использование ПО.
51. Что такое декомпозиция и где и для чего она применится в программной инженерии?
52. Охарактеризуйте проблемы измерения надежности.
53. Что такое каркасы (frameworks) приложений? Приведите примеры.
54. Какие диаграммы имеются в языке UML для визуального проектирования программ?

Дайте краткое описание известных вам диаграмм.

55. Опишите ключевые концепции проектирования программного обеспечения.
56. Перечислите основные концепции объектно-ориентированного проектирования.

Приведите их краткое описание.

57. Опишите преимущества и недостатки методологии SCRUM.
58. Опишите основные свойства WAF (Web application frameworks).
59. Перечислите и охарактеризуйте индикаторы риска при проектировании ПО.
60. Опишите основные принципы проектирования UI.