1 Поясніть поняття програмна бібліотека та різницю між динамічними і статичними бібліотеками

Програмна бібліотека — це набір реалізацій поведінки, написаних в рамках певної мови, який має чітко визначений інтерфейс для виклику цієї поведінки. Інакше кажучи, це програмний засіб, що містить реалізацію заданих функцій, доступних за чітко визначеним інтерфейсом, для їх подальшого використання.

Різниця між динамічними і статичними бібліотеками полягає у тому, що зв'язки цільового виконуваного файлу з функціями статичних бібліотек встановлюються у момент компіляції, а динамічних — у момент завантаження ресурсів або виконання програми.

2 Поясніть поняття «драйвер-фільтр». Для чого можуть бути потрібні такі драйвери?

Драйвер-фільтр — це необов'язковий, опціональний драйвер, який розширює або модифікує існуючу поведінку пристрою. Такі драйвери можуть знадобитись для розширення базового функціоналу, наприклад, управління живленням, прискорення пересування курсора мишою за допомогою відповідних трансформацій вхідних даних, впровадження додаткових перевірок безпеки вводу з клавіатури тощо.

Існує декілька типів таких драйверів: драйвери-фільтри шин, драйвери-фільтри нижчого рівня та драйвери фільтри вищого рівня. Останні два типи також поділяються на пристроєві та класові: для одного певного пристрою або цілого класу пристроїв відповідно.

Драйвери-фільтри шин зазвичай розширюють можливості шини та постачаються Microsoft або розробником пристрою. Такі драйвери можуть, наприклад, додавати пропрієтарні розширення та покращення до стандартного апаратного забезпечення шини.

Для пристроїв, що описуються ACPI BIOS, менеджер живлення вбудовує ACPI-фільтр (драйвер-фільтр шини) над драйвером шини для кожного пристрою. Такий ACPI-фільтр наглядає за політикою живлення пристрою та вмикає або вимикає їх. Такий ACPI-фільтр прозорий і видимий іншим драйверами і відсутній на машинах без підтримки ACPI.

Драйвери-фільтри нижчого рівня зазвичай змінюють поведінку апаратного забезпечення пристрою. Вони зазвичай необов'язкові, опціональні і надаються незалежними виробниками (вендорами) апаратного забезпечення. Для кожного пристрою можуть існувати і використовуватись будь-яка кількість драйверів-фільтрів нижчого рівня.

Пристроевий драйвер-фільтр нижчого рівня (lower-level device filter driver) відстежує та/або модифікує запити вводу-виводи для певного пристрою. Зазви-

чай, такі фільтри перевизначають апаратну поведінку під очікувану специфікацію.

Класовий драйвер-фільтр нижчого рівня відстежує та/або модифікує запити вводу-виводу для класу пристроїв. Наприклад, класовий драйвер-фільтр для пристроїв типу «миша» може реалізовувати прискорення за допомогою нелінійного перетворення даних про переміщення миші.

Драйвери-фільтри вищого рівня зазвичай надають подальші розширення функціоналу для певного пристрою. Такі драйвери зазвичай створюються незалежними вендорами апаратного забезпечення та є необов'язковими. Для певного пристрою може бути необмежена кількість драйверів-фільтрів вищого рівня.

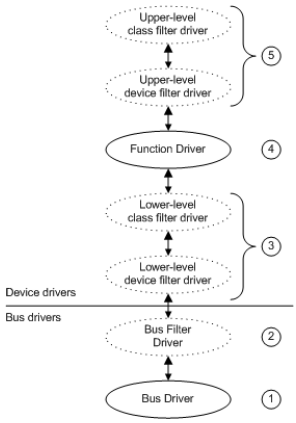

Рис. 1: Рівні драйверів

3 Охарактеризуйте вкладки Properties (Властивості) та Events (Події) Інспектора Об'єктів інтегрованого середовища візуальної розробки (CBuilder/RAD Studio etc)

Вкладка Properties Інспектора об'єктів дає змогу користувачу задавати властивості компонентів на формі під час проектування. Ця вкладка показує властивості вибраного на формі компонента і містить дві колонки: Property Name — назву властивості і Property Value — значення властивості. Редагування властивостей здійснюється за допомогою зміни необхідних значень властивостей. Сортування властивостей, перелічених у вкладці, можна виконувати як за алфавітним порядком, так і за категоріями.

Вкладка Events містить усі події для обраного компонента і форми. Ця вкладка містить дві колонки: Event Name — назву події і Event Handler — обробник події. Обробник події визначає, як компонент або форма реагує на певну подію. Щоб автоматично згенерувати декларацію обробника події, необхідно натисну-

ти на праву колонку Event Handler навпроти бажаної події.

4 Label, Panel

4.1 Охарактеризуйте компоненти Label, Panel інтегрованих середовищ візуальної розробки CBuilder RAD Studio

Label — це компонент, який виводить (додає) текст, що не може бути змінений користувачем на формі.

Panel — це компонент, який створює пусту панель на формі, на якій можна розмістити інші компоненти.

4.2 Розгляньте основні події цих компонентів

Основні події компонента Label виконують зазначені дії у таких випадках:

1. OnClick — при одноразовому натисканні на компонент.
2. OnDblClick — при подвійному натисканні на компонент.
3. OnDragDrop — при перетягуванні об'єкту в області компонента.
4. OnDragDrop — при відпущенні перетягуваного об'єкту в області компонента.
5. OnContextPopup — при виклику вспливаючого меню за допомогою правої клавіші миші або з клавіатури.
6. OnMouseActivate — при натисканні клавіші миші, коли курсор вказує на компонент і материнська форма неактивна.
7. OnMouseDown — при натисканні клавіші миші, коли курсор вказує на компонент.
8. OnMouseUp — при натисканні клавіші миші, коли курсор вказує на компонент.
9. OnMouseEnter — при наведенні курсору в область компонента.
10. OnMouseLeave — при виведенні курсору з області компонента.
11. OnMouseMove — при пересуванні курсору в області компонента.

Для компонента Panel характерні основні події компонента Label, доповнені такими подіями, що виконують задані дії у таких випадках:

1. OnCanResize — при спробі змінити розмір компонента.
2. OnEnter — при отриманні компонентом фокусу.
3. OnExit — при переході фокусу від одного компонента до іншого.
4. OnResize — при закінченні зміни розміру компонента.
5. OnUndock — при відкріпленні (undock) компонента.

4.3 Вкажіть компоненти, які є аналогами розглянутих в середовищі Visual Studio. Для кожної пари компонентів розгляньте їх спільні риси і відмінності.

В середовищі Visual Studio аналогічні компоненти мають назви Label, Panel відповідно. Оскільки ці компоненти є базовими для створення програм, за основним функціоналом вони ідентичні: Label відображає текст та дозволяє змінювати його розмір, шрифт, колір, а також інші елементи представлення в обох середовищах; Panel дозволяє вміщувати інструменти в обох середовищах, може підтримувати зміну розміру, відкріплення тощо.

До відмінностей компонентів відносяться наявність та назви властивостей: Visual Studio дозволяє налаштувати більшу кількість властивостей порівняно з C++ Builder. Наприклад, підтримку засобів для людей з обмеженими можливостями: відповідні параметри AccessibleDescription, AccessibleName, AccessibleRole наявні лише у Visual Studio. Наступною відмінністю є кількість і назви подій, пов'язаних з компонентами: Visual Studio знову надає більшу кількість обробників подій, і майже кожна назва змінюється, а в рамках даної модульної контрольної роботи наводити порівняльний список назв не несе користі і не має сенсу.

4.4 Для розв'язання яких задач, на Ваш погляд, доцільно використовувати ці компоненти? (Дайте загальну відповідь та наведіть приклад програми, в якій доцільно використати компонент)

Компонент Label зазвичай використовується щоб назвати та відмітити для користувача інший компонент на формі. Наприклад, його доцільно використати для позначення полів вводу та результату у програмі, що вираховує швидкість за пройденим шляхом та часом.

Як правило, компонент Panel використовується для створення панелі інструментів у програмі. Цей компонент доцільно застосувати у графічному редакторі для групування різноманітних інструментів малювання та редагування зображення.