



**ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ,
ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ**

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ(21)(22) Заявка: **2008123786/08**, 17.11.2006(24) Дата начала отсчета срока действия патента:
17.11.2006

Приоритет(ы):

(30) Конвенционный приоритет:
15.12.2005 US 11/304,240(43) Дата публикации заявки: **27.12.2009** Бюл. № 36(45) Опубликовано: **27.05.2011** Бюл. № 15(56) Список документов, цитированных в отчете о
поиске: **US 6847706 B2**, 25.01.2005. **US 2002/0149569**
A1, 17.10.2002. **RU 47120 U1**, 10.08.2005.(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: **11.06.2008**(86) Заявка РСТ:
US 2006/044741 (17.11.2006)(87) Публикация заявки РСТ:
WO 2007/070223 (21.06.2007)

Адрес для переписки:

**129090, Москва, ул.Б.Спасская, 25, стр.3,
ООО "Юридическая фирма Городисский и
Партнеры", пат.пов. А.В.Мишу, рег.№ 364**

(72) Автор(ы):

КОТИПАЛЛИ Кришна В. (US)

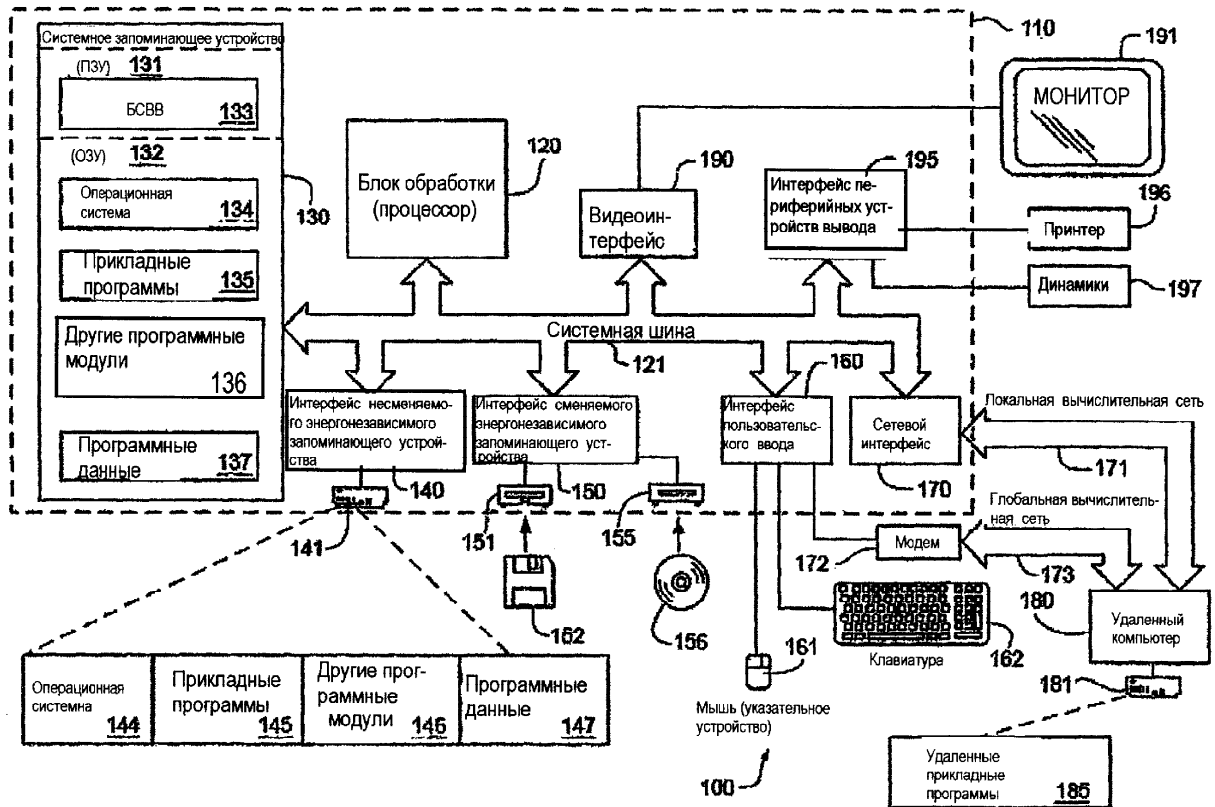
(73) Патентообладатель(и):

МАЙКРОСОФТ КОРПОРЕЙШН (US)**(54) ПРОГРАММИРУЕМАЯ КЛАВИАТУРА С РАЗВИТОЙ ЛОГИКОЙ**

(57) Реферат:

Изобретение относится к способам использования программируемой клавиатуры. Техническим результатом является возможность использования клавиатуры для различных приложений и аппаратных средств с помощью простых модификаций по отношению к экранному размеру программируемых клавиш и размеру шрифта, используемому для символов. Программируемая клавиатура имеет одну или несколько программируемых клавиш с наличием символа, отображенного на одной или нескольких программируемых клавишах. К

одному из символов, отображенному на одной из программируемых клавиш, может применяться модификация путем выбора программируемой клавиши и выполнения затем ввода по направлению. Пользователь может подтверждать ввод по направлению с помощью касания и направленной буксировки селектора по сенсорному экрану, множество символов, связанных с множеством клавиш модификации, становятся видимыми после того, как выбрана программируемая клавиша, так что пользователь может визуально определять, какой ввод по направлению будет обеспечивать требуемый символ. 3 н. и 17 з.п. ф-



ФИГ. 1а



**FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY,
PATENTS AND TRADEMARKS**

(19) RU⁽¹¹⁾ 2 419 833⁽¹³⁾ C2

(51) Int. Cl.
G06F 3/02 (2006.01)

(12) **ABSTRACT OF INVENTION**

(21)(22) Application: 2008123786/08, 17.11.2006

(24) Effective date for property rights:
17.11.2006

Priority:

(30) Priority:
15.12.2005 US 11/304,240

(43) Application published: **27.12.2009** Bull. 36

(45) Date of publication: **27.05.2011 Bull. 15**

(85) Commencement of national phase: **11.06.2008**

(86) PCT application:
US 2006/044741 (17.11.2006)

(87) PCT publication:
WO 2007/070223 (21.06.2007)

Mail address:

129090, Moskva, ul.B.Spasskaja, 25, str.3, OOO
"Juridicheskaja firma Gorodisskij i Partnery",
pat.pov. A.V.Mitsu, reg.№ 364

(72) Inventor(s):

KOTIPALLI Krishna V. (US)

(73) Proprietor(s):

MAJKROSOFT KORPOREJSHN (US)

(54) SMART PROGRAMMABLE KEYBOARD

(57) Abstract:

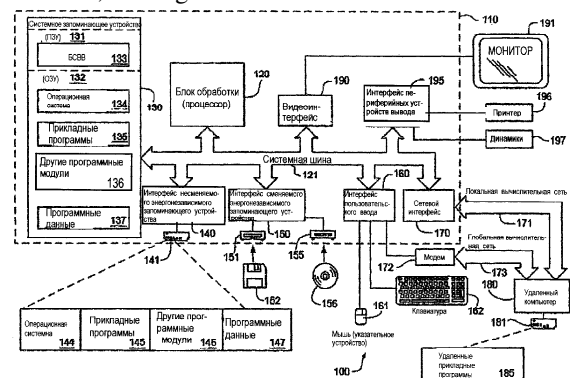
FIELD: information technology.

SUBSTANCE: programmable keyboard has one or more programmable keys with a character displayed on one or more programmable keys. One of the characters displayed on one of the programmable keys can be modified by selecting the programmable key and then performing directed input. The user can confirm directed input by touching and directed dragging of the selector on the touch screen. Several characters associated with the plurality of modification keys become visible after the programmable key is selected so that the user can visually determine which directed input will provide the required character.

EFFECT: possibility of using the keyboard for

various applications and devices using simple modifications to the screen size of the programmable keys and font size used for the characters.

20 cl, 22 dwg



ФИГ. 1а

УРОВЕНЬ ТЕХНИКИ

Портативные компьютеры, такие как планшетные персональные компьютеры (ПК), переносные ПК, персональные цифровые ассистенты (ПЦА, PDA) и карманные ПК, имеют ряд сложных конструктивных решений. С одной стороны, желательно, чтобы они были легкими по весу и простыми в эксплуатации. С другой стороны, желательно, чтобы они имели длительный срок службы аккумуляторной батареи, были мощными и предусматривали большой экран дисплея. Эти потенциально противоречивые принципы разработки конструктивных решений затрудняют создание идеального портативного компьютера.

Чтобы уменьшить вес и занимаемую площадь, один способ состоял в использовании на дисплее программируемой клавиатуры для ввода данных. Один способ обеспечения программируемой клавиатуры состоит в оснащении устройства отображения сенсорной функциональной возможностью. Часть дисплея, которой может быть доля в пределах от 1 до $1/N$, где N является положительным целым числом, может быть выполнена с возможностью отображения символов, таких как символ алфавита, на программируемых клавишах в виде конфигурации, которая напоминает клавиатуру. Символ может выбираться путем нажатия на соответствующее место на дисплее. Путем многократного выбора символов человек может вводить данные известным образом.

Как может быть понятно, однако, имеется ограниченный размер пространства, доступного на дисплее портативного устройства. Доступное пространство дополнительно уменьшается, если часть дисплея должна использоваться для отображения выборок пользователя. Кроме того, уменьшение размера символов ниже некоторой единицы является не пригодным для использования вследствие трудности визуального различения различия в символах. Следовательно, может быть трудным отображать все символы, которые в других отношениях возможно желательны для отображения. Эта сложность дополнительно усиливается при использовании некоторых языков, включающих в себя символы, которые могут включать в себя или не включать диакритический знак.

Одно решение состояло в том, чтобы обеспечивать клавиши модификации, такие как клавиша «shift» переключения регистра. Путем выбора соответствующей клавиши модификации могут изменяться варианты выбора, доступные для каждой программируемой клавиши. Однако это требует дополнительных выборок. Использование клавиши модификации на обычной аппаратно реализованной клавиатуре обычно не является трудностью, поскольку могут использоваться одновременно обе руки. Однако, чтобы выбирать последующие программируемые клавиши, программируемой клавиатурой обычно оперируют с использованием того же пальца или координатно-указательного устройства. Таким образом, необходимость, чтобы пользователь сначала выбирал клавишу модификации и затем выбирал требуемую программируемую клавишу, может нежелательно снижать скорость ввода данных.

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

На устройстве отображения (дисплее) с сенсорным экраном может обеспечиваться программируемая клавиатура, имеющая некоторое количество программируемых клавиш, и каждая программируемая клавиша может включать в себя символ. Пользователь может выбирать одну из программируемых клавиш, нажимая селектор на сенсорном экране в соответствующей позиции. Пользователь может выбирать модифицированную версию первоначально выбранного символа путем буксировки

селектора в направлении, ориентированном в зависимости от модификатора. Выборка может быть завершена или введена путем подъема селектора от сенсорного экрана. Чтобы обеспечивать пользователю обратную связь, потенциально возможные символы, связанные с первоначально выбранным символом, могут отображаться

5 вокруг начальной выборки, так что буксировка селектора в направлении требуемого символа будет давать возможность пользователю выбрать этот символ.

Данное краткое описание предусмотрено, чтобы в упрощенной форме представить подборку концепций, которые дополнительно описаны ниже в подробном описании.

10 Это краткое описание не предназначается для определения ключевых признаков или существенных признаков заявленного объекта изобретения, также оно не подразумевает, что будет использоваться для ограничения объема заявленного объекта изобретения.

КРАТКОЕ ОПИСАНИЕ ФИГУР ЧЕРТЕЖЕЙ

15 Настоящее изобретение иллюстрируется с целью примера и не ограничивается сопроводительными чертежами, на которых сходные числовые ссылочные позиции указывают сходные элементы и на которых:

фиг.1а - иллюстрация схематического представления универсальной цифровой

20 вычислительной среды, в которой могут быть осуществлены некоторые аспекты настоящего изобретения;

фиг.1в-1м - изображение универсальной вычислительной среды, поддерживающей один или несколько аспектов настоящего изобретения;

фиг.2 - иллюстрация схематического представления вычислительной среды

25 планшетного ПК, в которой могут быть осуществлены один или несколько аспектов настоящего изобретения;

фиг.3 - иллюстрация схематического представления вычислительной среды портативного ПК, в которой могут быть осуществлены один или несколько аспектов

30 настоящего изобретения;

фиг.4 - иллюстрация варианта осуществления программируемой клавиатуры на устройстве отображения, в котором могут быть осуществлены один или несколько аспектов настоящего изобретения;

фиг.5 - иллюстрация отображающей символ программируемой клавиши, в которой

35 могут быть осуществлены один или несколько аспектов настоящего изобретения;

фиг.6а-6б - иллюстрация вариантов осуществления набора символов, обеспечиваемых по периметру первого символа, в соответствии с одним или несколькими аспектами настоящего изобретения;

фиг.7-8 - иллюстрация способов выбора символов, связанных с первым символом, в

40 соответствии с одним или несколькими аспектами настоящего изобретения;

фиг.9 - иллюстрация способа ввода символа в соответствии с одним или несколькими аспектами настоящего изобретения.

ОСУЩЕСТВЛЕНИЕ ИЗОБРЕТЕНИЯ

45 Использование программируемой клавиатуры имеет некоторые преимущества. С одной стороны, могут быть предоставлены программные интерфейсы (API) приложения, чтобы позволять другим приложениям управлять и контролировать ее (клавиатуры) использование на экранном устройстве отображения. Кроме того, является легко настраиваемым цвет и внешний вид программируемой клавиатуры.

50 Кроме того, подобная программируемая клавиатура может использоваться для различных приложений и аппаратных средств с помощью простых модификаций по отношению к экранному размеру программируемых клавиш и размеру шрифта,

используемому для символов. Дополнительные преимущества станут очевидными в свете нижеследующего раскрытия.

ВЫЧИСЛИТЕЛЬНАЯ СРЕДА ОБЩЕГО НАЗНАЧЕНИЯ

На фиг. 1a иллюстрируется пример подходящей вычислительной системной среды 100, в которой может быть осуществлено изобретение. Вычислительная системная среда 100 является лишь одним примером подходящей вычислительной среды и не подразумевает наложение какого-либо ограничения относительно объема использования или функциональности изобретения. Также не следует интерпретировать вычислительную среду 100 в качестве имеющей какую-либо зависимость или требование, относящиеся к какому-либо одному компоненту или комбинации таковых, проиллюстрированных в примерной операционной среде 100.

Изобретение является работающим со многими другими универсальными или специализированными вычислительными системными средами или конфигурациями. Примеры известных вычислительных систем, сред и/или конфигураций, которые могут быть подходящими для использования вместе с изобретением, включают в себя, без ограничения указанными, персональные компьютеры, серверные компьютеры, переносные или портативные устройства, многопроцессорные системы, микропроцессорные системы, телевизионные абонентские приставки, программируемую бытовую электронику, сетевые ПК, миникомпьютеры, большие ЭВМ (мэйнфреймы), распределенные вычислительные среды, которые включают в себя любое из вышеуказанных систем или устройств, и т.п.

Изобретение может быть описано в общем контексте машиноисполняемых команд, таких как программные модули, исполняемые посредством компьютера. В общем, программные модули включают в себя процедуры, программы, объекты, компоненты, структуры данных и так далее, которые выполняют конкретные задачи или реализуют особые абстрактные типы данных. Изобретение также может быть осуществлено на практике в распределенных вычислительных средах, в которых задачи выполняются посредством удаленных устройств обработки, которые связаны через коммуникационную сеть. В распределенной вычислительной среде программные модули могут находиться в запоминающей среде как локального, так и удаленного компьютера, включая запоминающие устройства для хранения данных.

Как показано на фиг. 1a, иллюстративная система для осуществления настоящего изобретения включает в себя вычислительное устройство общего назначения в виде компьютера 110. Компоненты компьютера 110 могут включать в себя, без ограничения указанным, блок 120 обработки, системную память 130 и системную шину 121, которая связывает различные системные компоненты, включая системное запоминающее устройство, с блоком 120 обработки. Системная шина 121 может быть любой из различных типов шинных архитектур, включая шину памяти или контроллер памяти, периферийную шину и локальную шину, использующую любую шинную архитектуру из множества шинных архитектур. В качестве примера, а не ограничения, такие архитектуры включают в себя шину стандартной промышленной архитектуры (ISA), шину микроканальной архитектуры (MCA), расширенную ISA шину (EISA), локальную шину стандарта Ассоциации (VESA) по стандартам в области видеoeлектроники и шину межсоединения (PCI) периферийных компонентов, известную также как шина расширения.

Компьютер 110 обычно включает в себя различные машиночитаемые носители. Машиночитаемые носители могут быть любыми имеющимися носителями, к которым может осуществлять доступ компьютер 110, и включают в себя энергозависимые и

энергонезависимые носители, съемные и несъемные носители. В качестве примера, а не ограничения, машиночитаемые носители могут содержать носители для запоминающих устройств компьютера и носители для передачи данных. Носители для запоминающих устройств включают в себя энергозависимые и энергонезависимые, 5 съемные и несъемные носители, реализованные любым способом или технологией, предназначенными для хранения информации, такой как машиночитаемые команды, структуры данных, программные модули или другие данные. Носители запоминающих устройств компьютера включают в себя, без ограничения указанным, 10 оперативное запоминающее устройство (ОЗУ, RAM), постоянное запоминающее устройство (ПЗУ, ROM), электрически стираемое программируемое ПЗУ (ЭСППЗУ, EEPROM), флэш-память или другую технологию памяти, ПЗУ на компакт-диске (CD-ROM), цифровые универсальные диски (DVD) или другое ЗУ на оптическом диске, ЗУ на магнитных кассетах, магнитной ленте, магнитном диске или другие ЗУ на 15 магнитных носителях, или любой другой носитель, который может быть использован для хранения требуемой информации и к которому может осуществлять доступ компьютер 110. Носитель (среда) передачи данных обычно реализует машиночитаемые команды, структуры данных, программные модули или другие 20 данные в виде модулированного сигнала данных, такого как несущая или другой транспортный механизм, и включает в себя любой носитель для доставки информации. Термин «модулированный сигнал данных» означает сигнал, у которого одна или несколько характеристик установлены или изменяются таким образом, чтобы в сигнале кодировать информацию. В качестве примера, а не ограничения, 25 среды передачи данных включают в себя проводные среды, такие как проводная сеть или непосредственное проводное соединение, и беспроводные среды передачи, такие как акустические, радиочастотные, инфракрасные и другие беспроводные среды передачи. Комбинации из любых из вышеуказанных сред передачи также следует 30 включать в объем машиночитаемых носителей.

Системная память 130 включает в себя компьютерные носители в виде энергозависимого и/или энергонезависимого запоминающего устройства, такого как постоянное запоминающее устройство (ПЗУ, ROM) 131 и оперативное запоминающее устройство (ОЗУ, RAM) 132. Базовая система 133 (BIOS) ввода-вывода, содержащая 35 базовые процедуры, обеспечивающие передачу информации между компонентами компьютера 110, например, при запуске, обычно хранится в ПЗУ 131. ОЗУ 132 обычно содержит данные и/или программные модули, которые непосредственно доступны для блока 120 обработки и/или обрабатываются им в данный момент времени. В качестве 40 примера, а не ограничения, на фиг.1а проиллюстрирована операционная система 134, прикладные программы 135, другие программные модули 136 и программные данные 137.

Компьютер 110 может также включать в себя другие съемные/несъемные, энергозависимые/энергонезависимые компьютерные носители для хранения данных. В 45 качестве примера на фиг.1а проиллюстрирован накопитель 141 на жестком диске, который осуществляет считывание с несъемных, энергонезависимых магнитных носителей или запись на них, накопитель 151 на магнитном диске, который осуществляет считывание со съемного, энергонезависимого магнитного диска 152 или 50 запись на него, и накопитель 155 на оптическом диске, который осуществляет считывание со съемного, энергонезависимого оптического диска 156 или запись на такой диск, например CD-ROM или другой оптический носитель. Другие съемные/несъемные, энергозависимые/энергонезависимые компьютерные носители

для хранения данных, которые могут быть использованы в иллюстративной операционной среде, включают в себя, без ограничения указанным, кассеты на магнитных лентах, карты флэш-памяти, цифровые универсальные диски, ленту цифрового видеомэгнитофона, твердотельное ОЗУ, твердотельное ПЗУ и т.п.

Накопитель 141 на жестком диске обычно соединен с системной шиной 121 через интерфейс несъемной памяти, такой как интерфейс 140, и накопитель 151 на магнитном диске и накопитель 155 на оптическом диске обычно соединены с системной шиной 121 посредством интерфейса съемной памяти, такого как интерфейс 150.

Накопители и связанные с ними компьютерные носители для хранения данных, описанные выше и проиллюстрированные на фиг. 1а, обеспечивают хранилище для машиночитаемых команд, структур данных, программных модулей и других данных для компьютера 110. На фиг. 1а, например, накопитель 141 на жестком диске проиллюстрирован в качестве хранящего операционную систему 144, прикладные программы 145, другие программные модули 146 и программные данные 147. Следует отметить, что эти компоненты могут быть либо теми же, либо отличными от операционной системы 134, прикладных программ 135, других программных модулей 136 и программных данных 137. Операционная система 144, прикладные программы 145, другие программные модули 146 и программные данные 147 обозначены отличающимися ссылочными позициями, чтобы проиллюстрировать, что, по меньшей мере, они являются различными копиями. Пользователь может вводить команды и информацию в компьютер 110 через устройства ввода, такие как клавиатура 162, микрофон 163 и указательное устройство 161, обычно называемое мышью, шаровым манипулятором или сенсорной клавиатурой. Другие устройства ввода (не показаны) могут включать в себя микрофон, джойстик, игровую панель, спутниковую антенну, сканирующее устройство и т.п. Эти и другие устройства ввода обычно соединены с блоком 120 обработки через интерфейс 160 пользовательского ввода, соединенный с системной шиной, но могут быть соединены посредством другого интерфейса и шинных структур, таких как параллельный порт, игровой порт или универсальная последовательная шина (USB). Монитор 191, или другой тип устройства отображения также соединен с системной шиной 121 через интерфейс, например видеоинтерфейс 190. В дополнение к монитору компьютеры также могут включать в себя другие периферийные устройства вывода, такие как динамики 197 и принтер 196, которые могут быть соединены через интерфейс 195 периферийных устройств вывода.

Компьютер 110 может действовать в сетевой среде, используя логические соединения с одним или несколькими удаленными компьютерами, например удаленным компьютером 180. Удаленный компьютер 180 может быть персональным компьютером, портативным устройством, сервером, маршрутизатором, сетевым ПК, одноранговым устройством или другим обычным сетевым узлом и обычно включает в себя многие или все элементы, описанные выше относительно компьютера 110, хотя только запоминающее устройство 181 было проиллюстрировано на фиг. 1а. Логические соединения, изображенные на фиг. 1а, включают в себя локальную вычислительную сеть (ЛВС, LAN) 171 и глобальную вычислительную сеть (ГВС, WAN) 173, но также могут включать в себя другие сети. Такие сетевые среды являются обычными для учреждений, вычислительных сетей масштаба предприятия, внутрикорпоративных локальных сетей и сети Интернет.

При использовании в сетевой среде LAN компьютер 110 соединен с локальной

сетью 171 через сетевой интерфейс или адаптер 170. При использовании в сетевой среде WAN компьютер 110 обычно включает в себя модем 172 или другое средство установления связи в WAN 173, такой как Интернет. Модем 172, который может быть внутренним или внешним, может быть соединен с системной шиной 121 через интерфейс 160 пользовательского ввода или другое подходящее средство. В сетевой среде программные модули, изображенные относящимися к персональному компьютеру 110, или их части могут храниться на удаленном запоминающем устройстве. В качестве примера, а не ограничения, на фиг.1а проиллюстрированы удаленные прикладные программы 185 в качестве постоянно хранимых на запоминающем устройстве 181. Будет понятно, что показанные сетевые соединения являются иллюстративными и могут использоваться другие средства установления линии связи между компьютерами.

СРЕДА ПРОГРАММИРОВАНИЯ

Интерфейс программирования (или просто интерфейс) может рассматриваться в виде любого механизма, процесса, протокола для предоставления возможности одному или нескольким сегментам(у) кода взаимодействовать с функциональностью или осуществлять доступ к функциональности, обеспечиваемой посредством одного или нескольких других сегментов(а) кода. В качестве альтернативы интерфейс программирования может рассматриваться в виде одного или нескольких механизмов(а), способов(а), вызовов(а) функции, модулей(я), объектов(а) и так далее, компонента системы, способного с возможностью взаимодействия связываться с одним или несколькими механизмами(ом), способами(ом), вызовами(ом) функций, модулями(ем) и так далее других компонентов(а). Подразумевается, что термин "сегмент кода" в предшествующем предложении относится, в том числе, к одной или нескольким командам или строкам кода и охватывает, например, модули кода, объекты, подпрограммы, функции и так далее, независимо от применяемой терминологии, или того, являются ли сегменты кода отдельно компилированными или являются ли сегменты кода предоставленными в виде исходного, промежуточного или объектного кода, являются ли сегменты кода используемыми во время исполнения системы или процесса или находятся ли они на одном или различных компьютерах или распределены по многим компьютерам, является ли функциональность, представленная посредством сегментов кода, реализованной полностью в виде программного обеспечения, полностью в виде аппаратных средств, или комбинации аппаратных средств и программного обеспечения.

Абстрактно интерфейс программирования может рассматриваться в общем, как показано на фиг.1в или фиг.1с. На фиг.1в иллюстрируется интерфейс Interface1 в виде канала, через который взаимодействуют первый и второй сегменты кода. На фиг.1с иллюстрируется интерфейс в виде содержащего интерфейсные объекты I1 и I2 (которые могут являться или не являться частью первого и второго сегментов кода), которые дают возможность первому и второму сегментам кода системы взаимодействовать через среду М. Принимая во внимание фиг.1с, можно рассматривать интерфейсные объекты I1 и I2 в качестве отдельных интерфейсов той же системы и можно также рассматривать, что объекты I1 и I2 плюс среда М содержат интерфейс. Хотя на фиг.1в и 1с показаны двунаправленный поток и интерфейсы на каждой стороне потока, некоторые исполнения могут иметь поток информации только в одном направлении (или без информационного потока, как описано ниже) или могут иметь интерфейсный объект только на одной стороне. В качестве примера, а не ограничения, термины, такие как программный интерфейс (API) приложения,

точка входа, способ, функция, подпрограмма, удаленный вызов процедуры и интерфейс модели (СОМ) компонентных объектов попадают в рамки определения интерфейса программирования.

5 Аспекты такого интерфейса программирования могут включать в себя способ, посредством которого первый сегмент кода передает информацию (где "информация" используется в своем самом широком смысле и включает в себя данные, команды, запросы и так далее) на второй сегмент кода; способ, посредством которого второй сегмент кода принимает информацию; и структуру, последовательность, синтаксис, 10 организацию, схему, временную диаграмму передачи и содержимое информации. В этом отношении для действия интерфейса может не быть важной непосредственно лежащая в основе транспортная среда, является ли среда проводной или беспроводной или комбинацией обеих, если информация транспортируется способом, определенным согласно интерфейсу. В некоторых ситуациях информация может не пропускаться в 15 одном или обоих направлениях в традиционном смысле, поскольку передача информации может происходить либо через другое средство (например, информацию, помещенную в буфер, файл и так далее, отдельный от потока информации между сегментами кода), либо не происходить, если один сегмент кода осуществляет просто 20 доступ к функциональности, выполняемой посредством второго сегмента кода. Любой или все эти аспекты могут быть важными в данной ситуации, например, в зависимости от того, являются ли сегменты кода частью системы в архитектуре со слабыми связями или с непосредственными связями, и, таким образом, этот перечень следует рассматривать иллюстративным и неограничительным.

25 Это понятие интерфейса программирования является известным специалистам в данной области техники и ясным из вышеизложенного подробного описания изобретения. Однако имеются другие способы для реализации интерфейса программирования и, если не исключено явным образом, подразумевается, что эти 30 другие способы охватываются прилагаемой в конце описания формулой изобретения. Может оказаться, что такие другие способы являются в большей степени изоощренными или сложными, чем упрощенное представление по фиг.1в и 1с, но тем не менее они выполняют подобную функцию, чтобы достигать того же общего результата. Теперь кратко будут описаны некоторые иллюстративные 35 альтернативные осуществления интерфейса программирования.

А. РАЗЛОЖЕНИЕ НА ЭЛЕМЕНТАРНЫЕ ОПЕРАЦИИ

Процесс передачи информации от одного сегмента кода в другой может выполняться косвенно путем разбиения процесса передачи на множество дискретных 40 взаимодействий. Это изображено схематично на фиг.1d и 1e. Как показано, некоторые интерфейсы могут быть описаны в терминах делимых наборов функциональности. Таким образом, функциональность интерфейса по фиг.1в и 1с может быть разложена, чтобы достигать того же результата подобно тому, как можно математически 45 обеспечить 24, или 2 умножить на 2, умножить на 3 умножить на 2. Соответственно, как проиллюстрировано на фиг.1d, предоставляемая интерфейсом Interface1 функция может быть разделена, чтобы преобразовать взаимодействия интерфейса во множество интерфейсов Interface1A, Interface1B, Interface1C и так далее, при достижении того же результата. Как проиллюстрировано на фиг.1e, обеспечиваемая посредством 50 интерфейса I1 функция может быть разделена на ряд интерфейсов I1a, I1b, I1c и так далее при достижении того же самого результата. Подобным образом интерфейс I2 второго сегмента кода, который принимает информацию от первого сегмента кода, может быть разложен на ряд интерфейсов I2a, I2b, I2c и так далее. При разложении на

элементарные операции количество интерфейсов, включенных в 1-й сегмент кода, не соответствует количеству интерфейсов, включенных во 2-й сегмент кода. В любом из случаев по фиг.1d и 1e функциональная суть интерфейсов Interface1 и I1 остается той же, как по фиг.1b и 1c соответственно. Разложение интерфейсов на элементарные операции также может следовать ассоциативному, коммутативному и другому математическому правилу, так что разложение на элементарные операции может быть трудным для понимания. Например, упорядочение операций может быть неважным, и следовательно, выполняемая интерфейсом функция может быть выполнена заранее до обращения к интерфейсу посредством другой части кода или интерфейса или выполнена посредством отдельного компонента системы. Кроме того, средний специалист в области программирования может понять, что имеются различные способы выполнения различных вызовов функций, которые получают тот же результат.

В. ПЕРЕОПРЕДЕЛЕНИЕ

В некоторых случаях является возможным игнорировать, добавлять или переопределять некоторые аспекты (например, параметры) интерфейса программирования, хотя при этом выполняется заданный результат. Это проиллюстрировано на фиг.1f и 1g. Например, если предположить, что интерфейс Interface1 по фиг.1b включает в себя вызов функции Square(input, precision, output), включающий в себя три параметра «входной», «точность» и «выходной» и который выдается из 1-го сегмента кода на 2-й сегмент кода. Если в данном сценарии средний параметр «точность» не принимается во внимание, как показано на фиг.1f, он также может быть игнорирован или даже заменен незначущим (в этой ситуации) параметром. Можно также добавлять дополнительный параметр, не принимаемый во внимание. В любом случае может достигаться функциональность функции «square» (квадрат) при условии, что выходной параметр является возвращаемым значением после возведения в квадрат входного параметра посредством второго сегмента кода. Точность может быть весьма значущим параметром для некоторого нисходящего потока (обработки) или другой части вычислительной системы; однако если понято, что точность не является необходимой для узкоцелевого вычисления значения квадрата, она может быть заменена или игнорирована. Например, вместо передачи действительного значения точности может передаваться незначущее значение, например дата рождения, без неблагоприятного воздействия на результат. Подобным образом, как показано на фиг.1g, интерфейс I1 заменяется интерфейсом I1', переопределенным, чтобы игнорировать или добавлять параметры к интерфейсу. Интерфейс I2 может быть подобным образом переопределен в виде интерфейса I2', переопределенного, чтобы игнорировать ненужные параметры, или параметры, которые могут быть обработаны в другом месте. Вопрос состоит в том, что в некоторых случаях интерфейс программирования может включать в себя аспекты, например параметры, которые не являются необходимыми для некоторой цели, и поэтому они могут игнорироваться или переопределяться, или обрабатываться в другом месте для других целей.

С. ВСТРОЕННОЕ КОДИРОВАНИЕ

Также может быть допустимым объединение некоторой или всей функциональности двух отдельных модулей кода, так что "интерфейс" между ними изменяет форму. Например, функциональность по фиг.1b и 1c может быть преобразована в функциональность по фиг.1n и 1i соответственно. На фиг.1n предыдущие 1-й и 2-й сегменты кода по фиг.1b объединены в модуль, содержащий и

тот, и другой. В этом случае сегменты кода все еще могут взаимодействовать друг с другом, но интерфейс может быть приспособлен к форме, которая является более подходящей для отдельного модуля. Таким образом, например, формальные предложения Call (вызов) и Return (возвращение) могут более не являться необходимыми, но подобная обработка или ответное действие(я) в соответствии с интерфейсом Interface1 все еще могут быть действительными. Подобным образом показанная на фиг.1i часть интерфейса (или весь интерфейс) I2 по фиг.1с может быть написана встроенной в интерфейс I1, чтобы образовать интерфейс I1". Как проиллюстрировано, интерфейс I2 разделен на I2a и I2b, и порция I2a интерфейса была кодирована встроенной вместе с интерфейсом I1, чтобы образовать интерфейс I1". Для конкретного примера можно полагать, что интерфейс I1 по фиг.1с выполняет вызов функции square(input,output), принимаемый интерфейсом I2, который после обработки значения, передаваемого с помощью входного параметра (для возведения его в квадрат), посредством второго сегмента кода передает назад возведенный в квадрат результат с помощью выходного параметра. В таком случае обработка, выполняемая посредством второго сегмента кода (возведение в квадрат входного параметра), может быть выполнена посредством первого сегмента кода без обращения к интерфейсу.

D. РАЗЪЕДИНЕНИЕ

Процесс передачи информации от одного сегмента кода в другой может выполняться косвенно путем разбиения процесса передачи на множество дискретных взаимодействий. Это схематично изображено на фиг.1j и 1k. Как показано на фиг.1j, обеспечиваются один или несколько элементов промежуточного программного обеспечения (разъединительный интерфейс(ы), поскольку они отъединяют функциональность и/или функции интерфейса от исходного интерфейса), чтобы преобразовывать процессы передачи в первом интерфейсе Interface1 для приведения их в соответствие другому интерфейсу, в этом случае интерфейсам Interface2A, Interface2B и Interface2C. Это может выполняться, например, там, где имеется установленная основа приложений, предназначенная для взаимодействия, например, с операционной системой в соответствии с протоколом Interface1, но затем операционная система изменяется, чтобы использовать другой интерфейс, в этом случае интерфейсы Interface2A, Interface2B и Interface2C. Вопрос в том, что исходный интерфейс, используемый 2-ым сегментом кода, изменяется так, что он более не является совместимым с интерфейсом, используемым 1-ым сегментом кода, и поэтому используется посредник, чтобы старый и новый интерфейсы стали совместимыми. Подобным образом, как показано на фиг.1k, с помощью интерфейса DI1 разъединения может быть представлен третий сегмент кода, чтобы принимать передачи информации от интерфейса I1 и чтобы с помощью интерфейса DI2 разъединения передавать функциональность интерфейса, например, на интерфейсы I2a и I2b, перепроектированные для работы с DI2, но обеспечивать тот же функциональный результат. Подобным образом DI1 и DI2 могут работать вместе, чтобы перенести функциональность интерфейсов I1 и I2 по фиг.1с на новую операционную систему при обеспечении того же или подобного функционального результата.

E. ПЕРЕОПРЕДЕЛЕНИЕ

Еще одним возможным вариантом является динамическое переопределение кода, чтобы заменять функциональность интерфейса чем-либо другим, но достигающим того же общего результата. Например, может иметься система, в которой сегмент кода, представленный на промежуточном языке (например, языке IL

компании Microsoft, байткодах языка Java и так далее), поставляется на оперативный (JIT) компилятор или интерпретатор в среде исполнения (такой, как предоставляется инфраструктурой .Net, Java-средой времени исполнения или другими подобного типа средами времени исполнения). JIT-компилятор может быть написан так, чтобы динамически преобразовывать передачу информации от 1-го сегмента кода на 2-й сегмент кода, то есть устанавливать их соответствие другому интерфейсу, как может требоваться 2-ым сегментом кода (либо исходным, либо отличающимся 2-ым сегментом кода). Это изображено на фиг.1l и 1m. Как может быть видно на фиг.1l, этот подход подобен сценарию разъединения, описанному выше. Это может выполняться, например, там, где установленная основа приложений предназначена для взаимодействия с операционной системой в соответствии с протоколом Interface1, но затем операционная система изменяется, чтобы использовать другой интерфейс. Может быть использован JIT-компилятор, чтобы оперативно привести в соответствие процессы передачи информации от установленных основных приложений на новый интерфейс операционной системы. Как изображено на фиг.1m, этот подход динамического представления в другом виде интерфейса(ов) может применяться для динамического разложения или иным образом также изменять интерфейс(ы).

Также отмечается, что вышеописанные сценарии достижения того же или сходного результата в виде интерфейса через альтернативные осуществления также могут комбинироваться различными способами, последовательно и/или параллельно, или с другим промежуточным кодом. Таким образом, представленные выше альтернативные варианты осуществления не являются взаимно исключающими и могут быть смешанными, согласованными и комбинированными, чтобы создавать те же или эквивалентные сценарии по отношению к основным сценариям, представленным на фиг.1v и 1с. Также отмечается, что, как в случае большинства конструкций программирования, имеются другие сходные способы достижения той же или сходной функциональности интерфейса, которые могут не описываться в документе, но тем не менее могут быть представленными согласно существу и объему изобретения, то есть отмечается, что в основе значения интерфейса лежат, по меньшей мере частично, функциональность, представленная посредством интерфейса, и полезные результаты, возможные согласно интерфейсу.

ВЫЧИСЛИТЕЛЬНЫЕ СРЕДЫ ПОРТАТИВНЫХ ПК

На фиг.2 приводится пример иллюстративного планшетного ПК 201, который может использоваться в соответствии с различными аспектами настоящего изобретения. Некоторые или все функциональные возможности, подсистемы и функции в системе по фиг.1a могут быть включены в состав компьютера по фиг.2. Планшетный ПК 201 включает в себя большую поверхность 202 отображения, например цифровое устройство отображения с плоским экраном, предпочтительно экран жидкокристаллического дисплея (LCD), на котором отображен ряд окон 203. Используя перо 204, которое является примером селектора, пользователь может осуществлять выбор, подсветку и/или запись на поверхности 202 цифрового дисплея. Примеры подходящих поверхностей 202 цифрового дисплея включают в себя электромагнитные перьевые устройства ввода, такие как перьевые устройства цифрового ввода компаний Mutoh или Wacom. Также могут использоваться другие типы перьевых устройств цифрового ввода, например оптические устройства цифрового ввода. Планшетный ПК 201 интерпретирует графические знаки («жесты»), выполненные с использованием пера 204, чтобы управлять данными, вводить текст, создавать рисунки и/или исполнять задачи традиционных компьютерных приложений,

таких как электронные таблицы, программы обработки текстов и т.п.

Перо 204 может быть оснащено одной или несколькими кнопками или другими средствами, чтобы повысить его возможности для выбора. В одном варианте осуществления перо 204 может быть реализовано в виде "карандаша" или "пера", в котором один конец образует пишущую часть, и другой конец образует конец "стирающего элемента", который при перемещении по дисплею указывает части отображения, подлежащие стиранию. Могут использоваться другие типы устройств ввода, такие как «мышь», шаровой манипулятор или подобное. Дополнительно, непосредственно палец пользователя может быть пером 204 и использоваться для выбора или указания порций отображенного изображения на сенсорном контактно или бесконтактно дисплее. Следовательно, термин "пользовательское устройство ввода" используется в документе в его самом широком значении, охватывает многие разновидности известных устройств ввода типа пера 204. Область 205 показывает область обратной связи или область контакта, разрешающую пользователю определять, где перо 204 входило в контакт с поверхностью 202 дисплея. Таким образом, в аспектах настоящего изобретения перо 204 может использоваться в качестве эффективного селектора.

В различных вариантах осуществления система обеспечивает платформу работы с «чернилами» в виде набора СОМ-услуг, которые приложение может использовать, чтобы осуществлять ввод с экрана, управлять и хранить чернила. Одна услуга дает возможность приложению считывать и записывать чернила, используя раскрытые представления для чернил. Платформа работы с чернилами также может включать в себя язык разметки, включая язык, сходный с расширяемым языком (XML) разметки гипертекста. Дополнительно, система может использовать распределенную модель (DCOM) компонентных объектов в качестве другого осуществления. Могут использоваться другие реализации, включая модель программирования Win32 и модель программирования .Net от корпорации Microsoft.

На фиг.3 показаны функциональные компоненты переносного вычислительного устройства ("П/ПК") 320. Как изображено, П/ПК 320 включает в себя процессор 360, запоминающее устройство 362, устройство 328 отображения и клавиатуру 332. Запоминающее устройство 362 может включать в себя и энергозависимое запоминающее устройство (например, ОЗУ), и энергонезависимое запоминающее устройство (например, ПЗУ), карты стандарта Международной ассоциации (МАКППК, PCMCIA) производителей карт памяти для персональных компьютеров и так далее. Операционная система 364 может постоянно находиться в запоминающем устройстве 362 и может исполняться на процессоре 360. В варианте осуществления П/ПК 320 включает в себя операционную систему, такую как операционная система Windows® CE от корпорации Microsoft или другую операционную систему.

Одна или несколько прикладных программ 366 могут быть загружены в запоминающее устройство 362 и могут исполняться в операционной системе 364.

Примеры приложений включают в себя программы электронной почты, программы планирования, программы управления (PIM) персональной информацией, программы обработки текстов, программы электронных таблиц, программы Интернет-браузера, игры и другие известные приложения. П/ПК 320 также может включать в себя администратор 368 уведомлений, загруженный в запоминающее устройство 362, который также может исполняться на процессоре 360. Администратор 368 уведомлений может обрабатывать запросы на уведомления от приложений 366.

П/ПК 320 включает в себя источник 370 электропитания, который может быть

реализован в виде одной или нескольких аккумуляторных батарей или некоторого другого источника электропитания, такого как конденсаторный или топливный элемент или подобный. Источник 370 электропитания может дополнительно
 5 включать в себя внешний источник электропитания, который заменяет или повторно заряжает встроенные аккумуляторные батареи, такой как адаптер переменного тока или питаемая стыковочная шина.

Как изображено, П/ПК 320 включает в себя три типа средств внешнего уведомления: светоизлучающий диод (СИД, LED) 340, колебательное устройство 372 и
 10 звуковой генератор 374. Эти устройства могут быть непосредственно связаны с источником электропитания 370, так что при активировании они могут оставаться включенными в течение длительности, предписанной средством уведомления, даже если П/ПК процессор 360 и другие компоненты могут отключаться, чтобы снизить
 15 потребление энергии. В варианте осуществления СИД 340 может оставаться включенным, пока пользователь не предпримет действие. Существующие в настоящее время версии колебательного устройства 372 и звукового генератора 374 используют слишком много электроэнергии для современных батарей П/ПК и поэтому они могут
 20 быть выполнены с возможностью выключения, когда это делает остальная часть системы, или в некоторый конечный промежуток времени после активации.

ПРИЛОЖЕНИЯ

На фиг.4 изображен вариант осуществления портативного компьютера 401. Портативный компьютер, который может иметь конфигурацию, как обсуждено выше, включает в себя корпус 402, в котором предусмотрен дисплей 405. Как изображено,
 25 обеспечивается программируемая клавиатура 406, включающая в себя конфигурацию 408 из программируемых клавиш 410. Должно быть отмечено, что также могут обеспечиваться другие конфигурации 408.

Если требуется, конфигурация 408 программируемой клавиатуры 406 может быть
 30 управляемой посредством приложения через API, как обсуждено выше. В варианте осуществления может использоваться параметр, чтобы настраивать конфигурацию 408 программируемых клавиш 410. Параметр может включать в себя заданное по умолчанию значение, которое может быть модифицировано с тем, чтобы
 35 настроить конфигурацию 408, таким образом, различные приложения могут настраивать конфигурацию 408 программируемых клавиш 410, как требуется. В варианте осуществления программируемая клавиатура 406 может включать в себя набор конфигураций 408, которые могут выбираться посредством приложения, каждая конфигурация размещает программируемые клавиши 410 в различных заранее
 40 заданных местоположениях. В другом варианте осуществления конфигурация 408 может быть полностью настраиваемой так, что приложения могут размещать индивидуальные программируемые клавиши 410, как требуется.

Чтобы предоставлять возможность пользователям вводить данные в
 45 вычислительное устройство, каждая программируемая клавиша 410 может быть связана с символом, например символом "a", используемым в английском языке. В варианте осуществления, как изображено на фиг.5, программируемая клавиша 410 может включать в себя символ 505, позиционированный приблизительно около центра программируемой клавиши 410. Как только выбрана программируемая клавиша 410,
 50 на которую можно воздействовать путем нажатия селектора на поверхности сенсорного дисплея, может настраиваться отображение, связанное с программируемой клавишей. Как может быть понятно, если программируемая клавиша достаточно большая, на программируемой клавише может быть отображено

более одного символа, и выбор программируемой клавиши затем дает возможность пользователю выбирать, какой из отображаемых символов желателен для выбора.

В варианте осуществления, как изображено на фиг.6а, осуществление выбора программируемой клавиши 410 вызывает отображение дополнительных символов 505.

Следует отметить, что выбор может быть нажатием селектора на сенсорном дисплее в позиции, совпадающей с требуемой программируемой клавишей. В варианте осуществления могут отображаться символы 505а-505h. Символ 505 также может оставаться отображенным. Символы 505а-505h представляют символ 505 в сочетании с различными клавишами модификации. Таким образом, символ 505а является связанным с символом 505 и первой клавишей модификации.

На фиг.6b иллюстрируется альтернативный вариант осуществления. Символ 505 является представлением "5". Путем обеспечения ввода по направлению может быть выбрано другое число, например 1-4 или 6-9. Как может быть понятно, такое осуществление позволит, чтобы одиночная программируемая клавиша обеспечивала все цифры, кроме нуля. Один возможный способ обеспечения нуля состоит в том, чтобы потребовать от пользователя выполнить направленную буксировку сначала в одном направлении и затем в противоположном направлении. Таким образом, как может быть понятно, составные перемещения могут быть связаны с дополнительными клавишами модификации, если требуется. В качестве альтернативы один или несколько символов, связанных с различными модификаторами, можно изменять после интервала времени задержки. Таким образом, в варианте осуществления, если поддерживалось касание без выполненного ввода по направлению, символ 505 может изменяться из "5" в "0" после некоторого заранее заданного интервала времени.

В варианте осуществления пользователь может выбирать программируемую клавишу 505, и различные символы могут становиться видимыми немедленно. В альтернативном варианте осуществления различные символы могут появляться, только если пользователь осуществляет продолжительный контакт с программируемой клавишей 505. Очевидно, что интервал времени контакта с программируемой клавишей 505, требуемый для обеспечения появления дополнительных символов, может быть настроен через API, как обсуждено выше.

В варианте осуществления некоторые символы, представляющие наиболее обычные модификации для символа 505, могут появляться первыми. Таким образом может обеспечиваться многоуровневое ответное действие. Кроме того, некоторые позиции могут наполняться символом прежде других позиций. Как может быть понятно, являются возможными многие изменения в варианте осуществления и эти изменения могут быть настроены через API. Таким образом, например, но без ограничения, первым может появляться символ 505а, затем символ 505е и так далее. Как может быть понятно, если у символа 505 не имеется достаточного количества возможных модификаций, количество символов 505а-505е, которые могут отображаться, может соответственно уменьшаться. Кроме того, некоторые символы могут назначаться для определяемых пользователем событий и действий и настраиваться посредством приложения или пользователем, или и тем, и другим.

В варианте осуществления первый символ, отображенный на программируемой клавише, и символы, связанные с первоначально отображенным символом, могут быть определены в зависимости от приложения. Таким образом, некоторые первоначально отображенные символы могут иметь большее количество связанных символов, тогда как другие символы могут иметь меньшее количество связанных символов. В варианте осуществления первый символ может иметь единственный

второй символ, представляющий первый символ, подвергнутый модификации согласно связанной клавише модификации. В таком варианте осуществления контакт при времени свыше запрограммированного интервала, который может быть любым интервалом времени больше нуля, может вызывать появление второго символа.

Чтобы увеличить количество возможных вариантов, связанных с таким символом, пользователь может связывать дополнительные символы с первым символом. В варианте осуществления пользователь может выбирать позицию дополнительных символов либо путем перемещения имеющихся символов, либо путем выбора позиции, которая на настоящий момент не является связанной с первым символом и вводом по направлению. В альтернативном варианте осуществления позиция будет выбираться автоматически посредством приложения согласно алгоритму. Одним таким алгоритмом может быть наполнение сначала позиций, изображенных на фиг.6 в виде символов 505a, 505c, 505e и 505g, способом по часовой стрелке. Как может быть понятно, изменения в таком алгоритме являются многочисленными и могут включать в себя статистический анализ действий пользователя и также могут включать в себя ввод исходя из исследований применимости.

Для выбора символа пользователь может перемещать в направлении символа селектор, которым может быть перо или палец, или некоторое другое устройство, подходящее для обеспечения ввода по направлению. Как изображено на фиг.6, 8, направления перемещения после начального выбора могут обеспечивать восемь различных символов, связанных с первым символом 505. Однако, если пользователь не обеспечил ввод по направлению (или если ввод по направлению был ниже заранее заданного порогового значения), то будет выбран первый символ 505. Следует отметить, что тогда как проиллюстрированы 8 различных символов, могут отображаться дополнительные символы. Однако, вероятно, что человек затруднился бы согласованно обеспечивать точное перемещение в направлениях свыше восьми, таким образом могут быть полезными некоторые дополнительные этапы, если с отдельной программируемой клавишей должны быть связаны более 8 модификаторов.

В варианте осуществления дополнительные символы могут отображаться по периметру вне границы программируемой клавиши 505 и могут находиться на расстоянии друг от друга. Чтобы содействовать пользователю в выборе требуемого символа, перемещение в направлении символа может вызывать графическое изменение, например изменение в цвете или градиенте (цвета) потенциально выбранного символа. Пользователь затем может перемещаться в другом направлении, чтобы вызвать подсветку по периметру другого символа. Как может быть понятно, количество связанных символов, размер периметра, тип обратной связи и чувствительность являются лишь некоторыми из параметров, которые могут изменяться от одной программируемой клавиши к другой программируемой клавише и от приложения к приложению, и эти параметры могут быть управляемыми посредством приложений через API.

На фиг.7 иллюстрируется способ выбора символа в соответствии с аспектом настоящего изобретения. Как проиллюстрировано, на этапе 710 пользователь выбирает первый символ. В варианте осуществления выбор включает в себя нажатие селектором на поверхности сенсорного дисплея, касание совпадает с требуемой программируемой клавишей, отображающей соответствующий символ.

Затем на этапе 715 становится видимым набор символов, связанных с первым символом. В варианте осуществления набор содержит один символ. В альтернативном варианте осуществления набор содержит 8 символов. Как может быть понятно,

изменения в количестве обеспечиваемых в наборе символов могут изменяться, как требуется. В варианте осуществления символы могут постепенно становиться более видимыми или могут появляться некоторым другим образом, таким как выдвижение справа, выдвижение из центра или просто внезапное появление. Очевидно, что это графическое управление тем, как символы наполняются и делаются видимыми, может быть различным и управляемым через API. Постепенное появление (исчезновение) может быть полезным, поскольку оно дает возможность пользователю воспринимать напоминание относительно того, какие символы были связаны с первоначально выбранным символом, без создания на экране нежелательного загромождения.

На этапе 720 пользователь обеспечивает ввод по направлению, чтобы выбрать символ из набора символов, связанных с первоначально выбранным символом. Ввод по направлению можно обеспечивать путем перемещения селектора в направлении требуемого символа. В варианте осуществления ввод по направлению можно обеспечивать путем перемещения селектора вдоль поверхности сенсорного дисплея в направлении, ориентированном в зависимости от символа в наборе символов. Это скользящее перемещение может называться направленной буксировкой селектора. Как отмечено выше, ввод по направлению может вызывать, что станет подсвеченным символ, который связан с конкретным вводом по направлению. В варианте осуществления подсвеченный символ может изменять размеры, может изменять цвет или контрастность или может обеспечивать графическую обратную связь некоторого другого вида, включая мультипликацию. В варианте осуществления в наборе можно обеспечивать множество символов. В варианте осуществления при обеспечиваемых в наборе многих символах начальный ввод по направлению, выполненный пользователем, будет выбирать один из символов в наборе. В альтернативном варианте осуществления с наличием в наборе многих символов пользователь может свободно перемещаться в различных направлениях, чтобы подсвечивать и выбирать различные символы, отличные от символа, который был бы выбран на основе выполненного начального ввода по направлению.

Затем на этапе 725 после того, как подсвечен требуемый символ, пользователь завершает выбор и обеспечивается, что подсвеченный символ будет передан (например, требуемый символ входит или вводится в приложение). В варианте осуществления завершение выбора символа может производиться путем удаления селектора от поверхности дисплея. Очевидно, что также могут использоваться другие способы, такие как активация устройства ввода, чтобы завершать выбор подсвеченного символа. В варианте осуществления активация кнопки может вызывать завершенный выбор и передачу подсвеченного в текущий момент символа. Такой способ может быть полезным для портативного устройства в ситуации, когда пользователь испытывает значительную вибрацию или не способен удерживать селектор устойчиво по некоторой другой причине, например медицинскому состоянию.

На фиг.8 иллюстрируется другой вариант осуществления способа выбора символа. Как может быть понятно, отличием между способами, изображенными на фиг.7 и фиг.8, является то, что опущен этап становления видимых символов, связанных с первым символом. Таким образом, в варианте осуществления символы, связанные с первым символом, не станут видимыми, но все еще могут выбираться путем надлежащего выбора и ввода по направлению, выполненному пользователем. В альтернативном варианте осуществления выбор символа не будет служить причиной, что связанные символы станут видимыми, если только не имеется продленный интервал задержки (например, пользователь продолжает нажимать на сенсорный

дисплей в течение длительного интервала времени). В другом альтернативном варианте осуществления связанные символы не станут видимыми, если пользователь не обеспечивает второй входной сигнал. Как может быть понятно, условие становления связанных символов видимыми может быть управляемым посредством приложения и может быть задано так, чтобы было управляемым пользователем через включение или отключение функциональной возможности в приложении.

Преимущество изображенного на фиг.8 способа состоит в том, что отображение не загромождается дополнительными символами и в зависимости от типа устройства отображения может требоваться меньшая энергия, поскольку нет изменения, выполненного по отношению к устройству отображения. Кроме того, такой способ менее сложен с графической точки зрения. Как может быть понятно, однако, такой способ требует, чтобы пользователь знал, какие символы связаны с первым выбранным символом наряду с направлением перемещения, требуемым для выбора требуемого связанного символа. Для некоторых языков, где имеются много дополнительных символов, связанных с первым символом, конфигурация, не показывающая на экране дополнительные символы, может быть несколько более сложной. Следовательно, такая конфигурация может наилучшим образом подходить для квалифицированных или опытных пользователей.

Как может быть понятно из вышеизложенного описания, когда с каждым символом связано множество символов, становится более трудным обеспечивать достаточную информацию пользователю без нежелательного загромождения отображения. Одно потенциальное решение, которое может использоваться в соответствии с различными аспектами настоящего изобретения, состоит в том, чтобы связывать конкретную клавишу модификации с конкретным направлением. Например, если возможные направления связаны со значением в градусах, находящимся между 0 и 359 градусами (поскольку 360 градусов будут тем же, что и 0), перемещение вдоль вектора в 0 градусов может быть связано с клавишей «shift» переключения регистра. Следует отметить, что направление в 0 градусов зависит от опорной точки и, следовательно, может быть настроено, как требуется. Возвращаясь на фиг.6, 0 градусов могут быть установлены на одной линии с местоположением символа 505a. Таким образом, в варианте осуществления помещение селектора на символ и затем перемещение в направлении нуля градусов (например, вверх) будет эквивалентным нажатию программируемой клавиши "select" (выбор) и затем нажатию символа. Очевидно, что нажатие символа и обеспечение затем направленной буксировки является более эффективным и может быть выполнено более быстро, чем выбор сначала программируемой клавиши «shift» и затем выбор требуемой клавиши. Хотя возможны многие варианты, может быть полезным потребовать, чтобы пользователь имел целью области, по меньшей мере, в 45 градусов так, чтобы имелось достаточное разделение между выборками.

В варианте осуществления направление «вверх» (или 0 градусов) может быть связано с клавишей модификации "shift", «вниз» (или 180 градусов) - с клавишей модификации "control", «вправо» (или 270 градусов) - с клавишей модификации "alt" и «влево» (или 90 градусов) - с клавишей модификации "function". Как может быть понятно, могут обеспечиваться изменения этого, и направление, связанное с конкретной клавишей модификации, может быть настраиваемым пользователем. Кроме того, некоторые модификации для символа, такие как верхний или нижний индексы, также могут быть связаны с направленной буксировкой.

Следует отметить, что некоторые программируемые клавиши могут не иметь

достаточных разновидностей, которые возможны для наполнения всех указаний (направлений), связанных с символом, отображенным на программируемой клавише. Например, может не иметься что-либо, связанное с модификатором "alt" и символом "m" английского алфавита. Наряду с тем, что это может быть зависимым от приложения и настраиваемым через API, как обсуждено выше, в варианте осуществления первый символ может обычно иметь только один связанный с ним модификатор, такой как, но без ограничения указанным, модификатор «shift». Например, в варианте осуществления клавиатура, предназначенная для использования, чтобы набирать текст на английском языке, может связывать направление «вверх» с версией заглавных букв для 26 букв алфавита и любое другое перемещение может игнорироваться. Таким образом, заглавная "a" может вводиться путем выбора программируемой клавиши "a" и обеспечения затем направленной буксировки в соответствующем нулю градусов направлении или вектору. В альтернативном варианте осуществления заглавные буквы могут быть нормальным видом вводимого символа, и строчные буквы могут вводиться путем выбора заглавной буквы и затем буксировки в направлении вектора 180 градусов.

Независимо от количества символов, связанных с символом, изображенным на программируемой клавише, может быть полезным давать возможность пользователю связывать дополнительные символы с первым символом, как упомянуто выше. Таким образом, в варианте осуществления пользователь может назначать символ программируемой клавише и направлению перемещения. В таком варианте осуществления символ "§" может связываться с выбором программируемой клавиши, которая отображает символ "s", и перемещением «вниз». Пользователь может даже импортировать дополнительные символы, если требуется. Таким образом, как может быть понятно, творческие возможности пользователя являются основным ограничивающим фактором для количества возможных изменений.

В варианте осуществления также может назначаться последовательность действий конкретному выбору и направлению перемещения. Например, макроопределение, чтобы выполнять последующие символы полужирными, может быть связано с выбором символа "b" и перемещением вправо. Чтобы отключать функциональную возможность полужирного шрифта, может использоваться та же операция или пользователь может выбирать "b" и перемещение влево. Таким образом, как может быть понятно, могут существовать многочисленные изменения, некоторые или все из которых могут быть настраиваемыми в соответствии с приложением. Следует отметить, что более сложные макроопределения также могут быть связаны с выбором программируемой клавиши и вводом по направлению.

Следует отметить, что некоторые языки ограничивают ввод информации на основании того, что было введено предварительно. Кроме того, некоторые языки образуют комбинации букв. Например, выбор "n" в японской слоговой азбуке кана может предоставлять пользователю выбор из na, ni, nu, ne и no. Таким образом, после ввода или выбора начального символа человек будет способен вводить только некоторые символы, которые соответствуют контексту предварительно введенного или выбранного символа. Таким образом, в варианте осуществления символы, связанные с конкретным символом, могут настраиваться согласно грамматическим правилам правописания и предварительно введенному символу(ам).

На фиг.9 иллюстрируется вариант осуществления такого способа. Сначала на этапе 910 пользователь вводит первый символ. Ввод символов может обеспечиваться путем использования селектора, которым может быть перо, чтобы касаться и

отпускать программируемую клавишу, предусмотренную на программируемой клавиатуре на сенсорном дисплее. Затем на этапе 915 пользователь выбирает программируемую клавишу, которая отображает второй символ; второй символ связан с первым символом или предназначен, чтобы использоваться вместе с ним. В варианте осуществления выбор выполняется путем продолжения нажатия программируемой клавиши с помощью селектора. Поскольку второй символ может быть связан с рядом потенциальных изменений, на этапе 920 набор символов, относящихся, по меньшей мере, к некоторым из потенциальных изменений во втором символе, отображается вокруг второго символа. В варианте осуществления все символы, которые связаны с первым символом и вторым символом, отображаются вокруг второго символа требуемым образом.

На этапе 925 пользователь обеспечивает ввод по направлению. Поскольку символы обеспечиваются вокруг второго символа, перемещение в сторону от начальной точки выбора будет обычно в направлении одного из символов в наборе. В варианте осуществления ввод по направлению может быть выполнен путем буксировки селектора по сенсорному дисплею в направлении одного из символов из набора после первоначального нажатия программируемой клавиши, которая отобразила второй символ. На этапе 930 пользователь останавливает процесс выборки и ввода по направлению, и выборка выбранного символа из набора символов является завершенной (например, введенной). В приложении обработки текстов ввод может иметь следствием, что выбранный символ является отображаемым рядом с предварительно введенным символом. В варианте осуществления пользователь может завершать выборку путем подъема селектора от сенсорного экрана после перемещения селектора в направлении требуемого символа. Завершение выборки также может именоваться осуществлением ввода или введением или передачей избранного символа. В варианте осуществления, как обсуждено выше, устройство отображения может подсвечивать текущий избранный символ, чтобы обеспечивать обратную связь пользователю относительно символа, который будет введен, если пользователь осуществит подъем селектора от дисплея в этот момент.

Следует отметить, что хотя приложения могут управлять раскрытыми функциональными возможностями через API, операционная система также может управлять функциональными возможностями, если требуется. Потенциально повышенные уровни гибкости обеспечиваются, однако, если функциональными возможностями управляют посредством приложения через API.

Настоящее изобретение было описано в терминах предпочтительных и примерных вариантов осуществления такового. Хотя предмет изобретения был описан на языке, конкретно определенном для структурных признаков и/или этапов способа, следует понимать, что объект изобретения, определенный в прилагаемой формуле изобретения, не обязательно ограничен конкретными признаками или действиями, описанными выше. Скорее конкретные, описанные выше признаки и действия раскрыты в качестве примерных форм реализации пунктов формулы изобретения. На основании анализа данного раскрытия многочисленные другие варианты осуществления, модификации и разновидности в рамках объема и сущности прилагаемой формулы изобретения могут быть реализованы специалистом в данной области техники.

Формула изобретения

1. Способ ввода данных в исполняющееся на компьютере приложение с помощью

программируемой клавиатуры на сенсорном экране, содержащий этапы, на которых:

принимают сигнал касания селектора на сенсорном экране;

отображают второй символ, связанный с первым символом, причем отображение содержит этапы, на которых генерируют на сенсорном экране графическое изображение второго символа, связанного с первым символом, в соответствующей позиции непосредственно после приема сигнала касания селектора;

принимают сигнал направленной буксировки селектора на сенсорном экране;

определяют, что касание селектора связано с первой программируемой клавишей, изображающей первый символ, и направленная буксировка селектора связана с первой клавишей модификации;

выбирают второй символ, причем второй символ связан с первым символом и первой клавишей модификации; и

передают второй символ в приложение.

2. Способ по п.1, в котором с первым символом связаны четыре клавиши модификации, и при этом каждый модификатор связан с отличающимся направлением, выбранным из набора направлений, состоящего из направлений 0° , 90° , 180° и 270° , причем направленная буксировка селектора направлена по направлению 0° .

3. Способ по п.1, в котором этап определения программируемой клавиши, связанной с касанием селектора, содержит этапы, на которых:

определяют, что селектор касается сенсорного дисплея в первой позиции; и

определяют, что первая программируемая клавиша связана с первой позицией на сенсорном экране.

4. Способ по п.1, в котором этап (а) приема сигнала направленной буксировки селектора содержит этапы, на которых:

определяют первую позицию, которой касался селектор на сенсорном экране;

определяют конечную позицию при прекращении селектором касания сенсорного экрана; и

определяют вектор, связанный с изменением в позиции между первой позицией и конечной позицией.

5. Способ по п.4, в котором этап определения, что первая клавиша модификации связана с направленной буксировкой селектора, содержит этапы, на которых:

определяют набор направлений, связанных с первой программируемой клавишей;

определяют из упомянутого набора направлений первое направление, с которым наиболее близко совпадает вектор направленной буксировки селектора; и

определяют, что первая клавиша модификации связана с первым направлением.

6. Способ по п.5, в котором этап приема сигнала направленной буксировки селектора дополнительно содержит этап, на котором:

подсвечивают второй символ, связанный с первой клавишей модификации.

7. Способ по п.1, дополнительно содержащий этап, на котором:

после приема сигнала касания селектора, но прежде приема сигнала направленной буксировки селектора, видимым образом позиционируют второй символ по вектору, связанному с первой клавишей модификации.

8. Способ по п.1, дополнительно содержащий этапы, на которых:

после приема сигнала касания селектора, отображают один или более дополнительных символов, связанных с первым символом,

генерируют графическое изображение каждого из упомянутых одного или более дополнительных символов в соответствующей позиции непосредственно после приема

сигнала касания селектора;

определяют, что касание селектора связано с первой программируемой клавишей, изображающей первый символ, и направленная буксировка селектора связана с клавишей модификации, относящейся к одному из упомянутых одного или более

дополнительных символов;
выбирают упомянутый один из упомянутых одного или более дополнительных символов; и

передают упомянутый один из упомянутых одного или более дополнительных символов в приложение.

9. Машиночитаемый носитель, включающий в себя машиночитаемые команды для выполнения этапов, на которых:

принимают сигнал первой выборки, связанный с первой программируемой клавишей, имеющей первый символ;

определяют набор символов, связанных с первой программируемой клавишей;
принимают уведомление, что селектор касается сенсорного экрана в течение заранее заданного интервала времени;

отображают набор символов вокруг первого символа;

принимают сигнал направленной буксировки, связанной с первым выбором; и
передают информацию второго символа из набора символов, причем второй символ связан с упомянутым сигналом направленной буксировки и первым символом.

10. Машиночитаемый носитель по п.9, дополнительно содержащий машиночитаемые команды для выполнения этапов, на которых:

после приема сигнала направленной буксировки подсвечивают второй символ, связанный с упомянутым вводом по направлению.

11. Машиночитаемый носитель по п.9, в котором этап отображения содержит этапы, на которых:

определяют периметр для позиционирования набора символов вокруг первого символа; и

обеспечивают команды для позиционирования по периметру в соответствующей позиции каждого символа из набора символов.

12. Машиночитаемый носитель по п.11, в котором этап обеспечения команд содержит этапы, на которых:

определяют набор символов, связанных с первым символом;

определяют направления, связанные с соответствующей, по меньшей мере, одной клавишей модификации, связанной с набором символов; и

обеспечивают команды для позиционирования каждого из символов из набора символов в соответствии с направлением соответственной, по меньшей мере, одной клавиши модификации.

13. Машиночитаемый носитель по п.9, дополнительно содержащий машиночитаемые команды для выполнения этапов, на которых:

обеспечивают команды для отображения набора символов вокруг первого символа, причем каждый из символов в наборе позиционирован в направлении, соответствующем клавише модификации, связанной с соответственным символом.

14. Машиночитаемый носитель по п.9, в котором этап приема сигнала

направленной буксировки содержит этапы, на которых:

определяют, что селектор активировал сенсорный экран в первой позиции; и

определяют, что селектор скользяще перемещен по сенсорному экрану во вторую позицию.

15. Способ ввода данных с использованием набора программируемых клавиш на устройстве отображения, в компьютерной системе с наличием графического пользовательского интерфейса, включающего в себя устройство отображения и устройство выбора в составе пользовательского интерфейса, причем способ содержит этапы, на которых:

принимают сигнал выбора первой программируемой клавиши из набора программируемых клавиш, причем первая программируемая клавиша включает в себя первый символ;

отображают второй символ, связанный с первым символом, прежде приема сигнала ввода по направлению, при этом отображение содержит этапы, на которых:

ожидают в течение заданного интервала времени после приема сигнала выборки; и формируют графическое изображение второго символа, связанного с первым символом, на устройстве отображения в заданной позиции;

принимают сигнал ввода направления, обеспеченный устройством выбора в составе пользовательского интерфейса; и

вводят второй символ, связанный с вводом по направлению и первым символом.

16. Способ по п.15, в котором этап формирования графического изображения содержит стадии:

после приема сигнала выборки первой программируемой клавиши определяют, что второй символ связан с упомянутой программируемой клавишей;

определяют ориентированное положение, связанное со вторым символом; и

отображают графическое изображение второго символа на устройстве отображения, причем второй символ позиционирован в соответствии с ориентированным положением, связанным со вторым символом.

17. Способ по п.15, в котором этап приема сигнала ввода направления содержит этапы, на которых:

отмечают начальную позицию селектора на сенсорном экране;

отмечают конечную позицию селектора на сенсорном экране; и

определяют направление перемещения селектора относительно заранее заданной опорной точки.

18. Способ по п.15, дополнительно содержащий этап, на котором:

обеспечивают набор программируемых клавиш на устройстве отображения в виде настраиваемой конфигурации, при этом приложение осуществляет выбор из, по меньшей мере, двух различных конфигураций.

19. Способ по п.15, дополнительно содержащий этап, на котором:

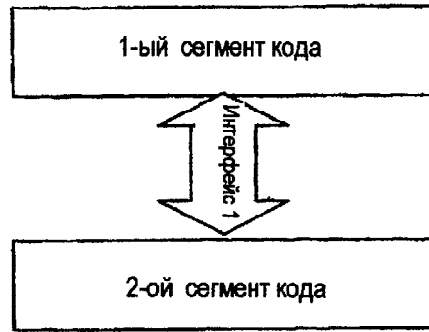
обеспечивают набор программируемых клавиш на устройстве отображения в виде настраиваемой конфигурации, при этом приложение осуществляет выбор конфигурации и набора программируемых клавиш.

20. Способ по п.15, дополнительно содержащий этапы, на которых:

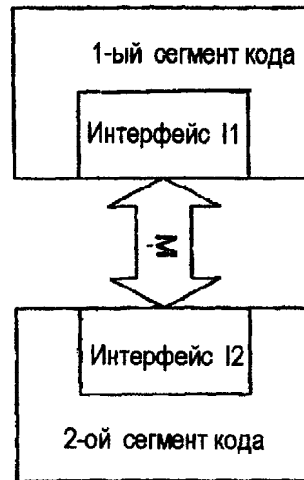
после приема сигнала выборки первой программируемой клавиши определяют один или более дополнительных символов связанных с упомянутой программируемой клавишей;

определяют ориентированное положение, связанное с каждым из упомянутых одним или более дополнительных символов; и

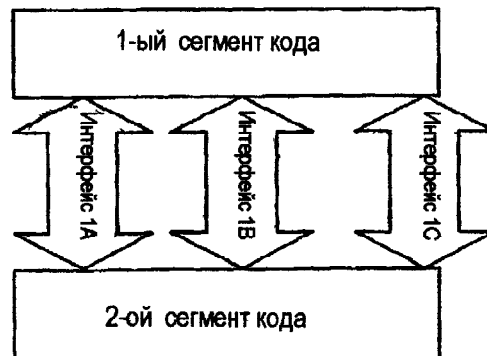
отображают графическое изображение упомянутых одного или более дополнительных символов на устройстве отображения, причем упомянутые один или более дополнительных символов позиционированы в соответствии с ориентированным положением, связанным со вторым символом.



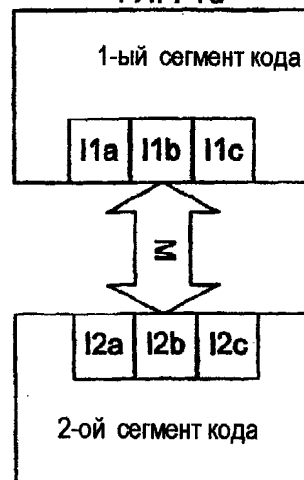
ФИГ. 1b



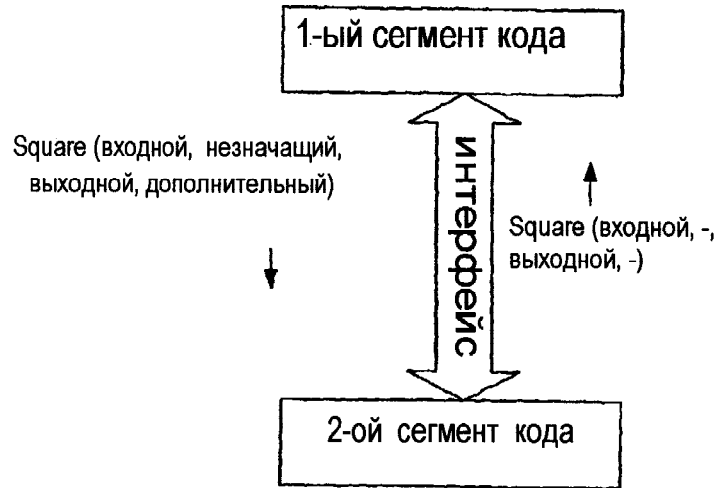
ФИГ. 1c



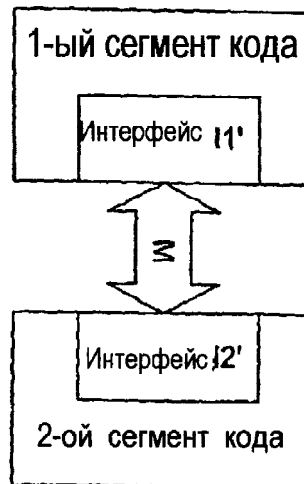
ФИГ. 1d



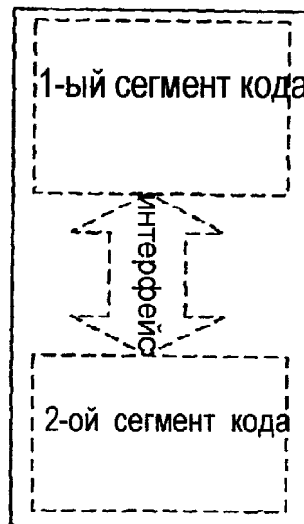
ФИГ. 1e



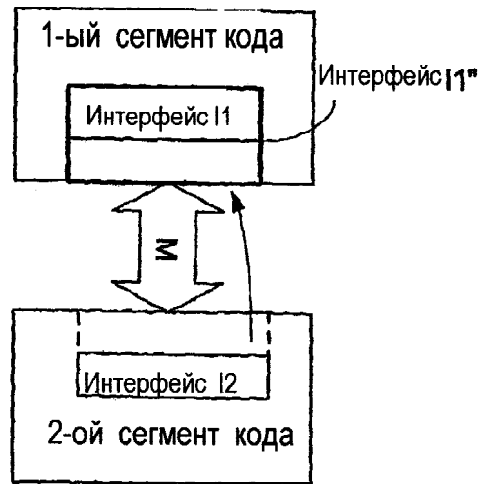
Фиг. 1f



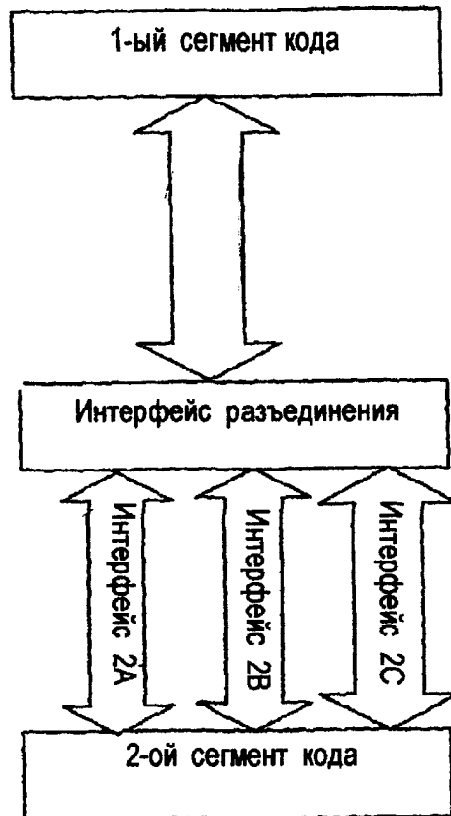
Фиг. 1g



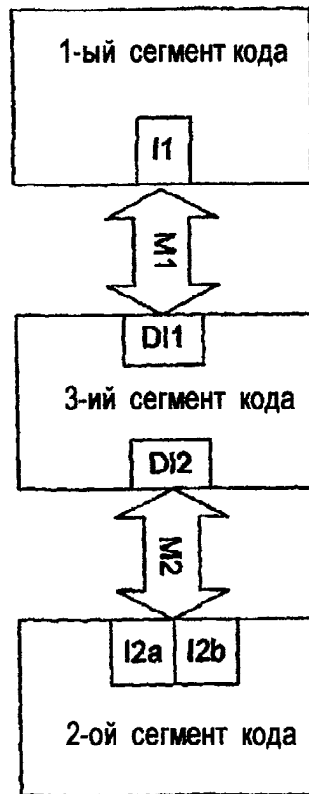
Фиг. 1h



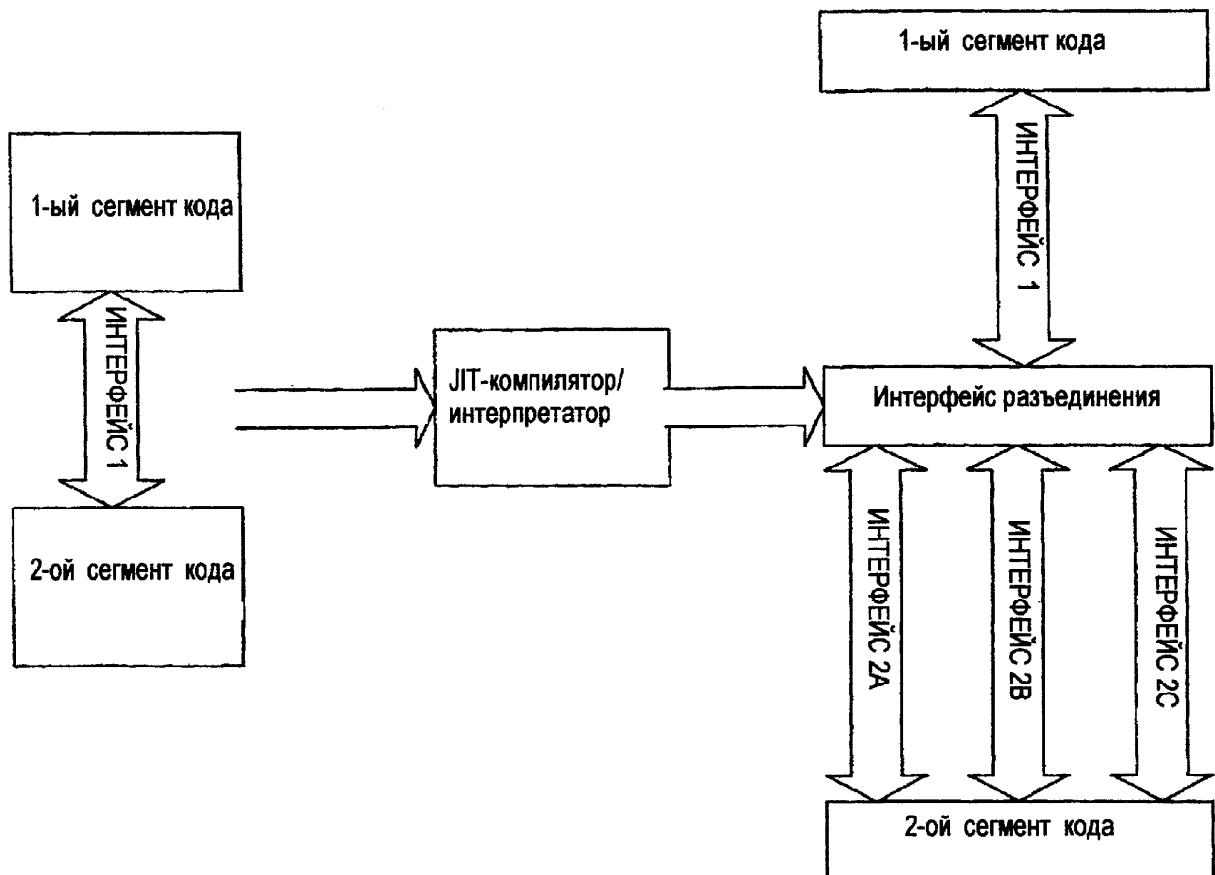
Фиг. 1i



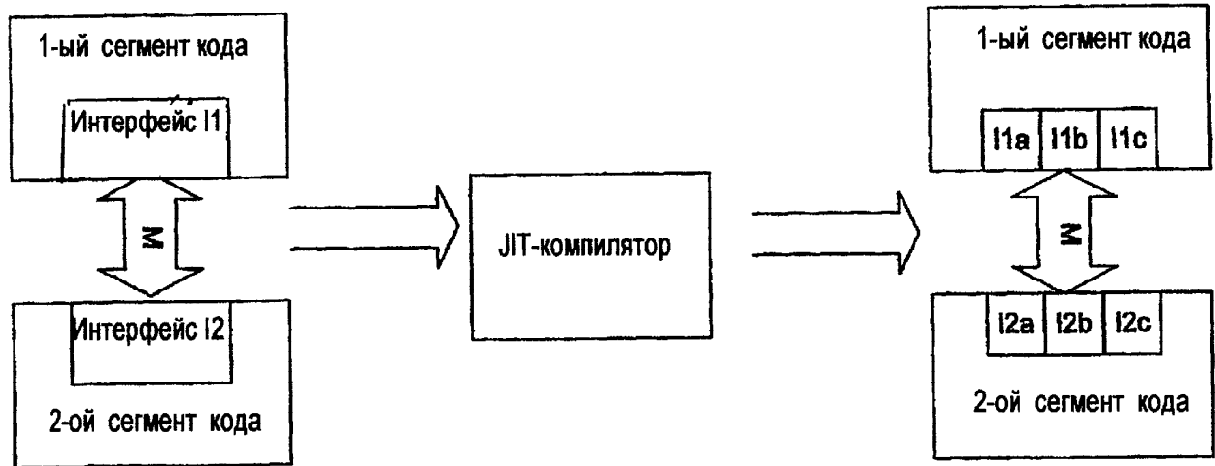
Фиг. 1j



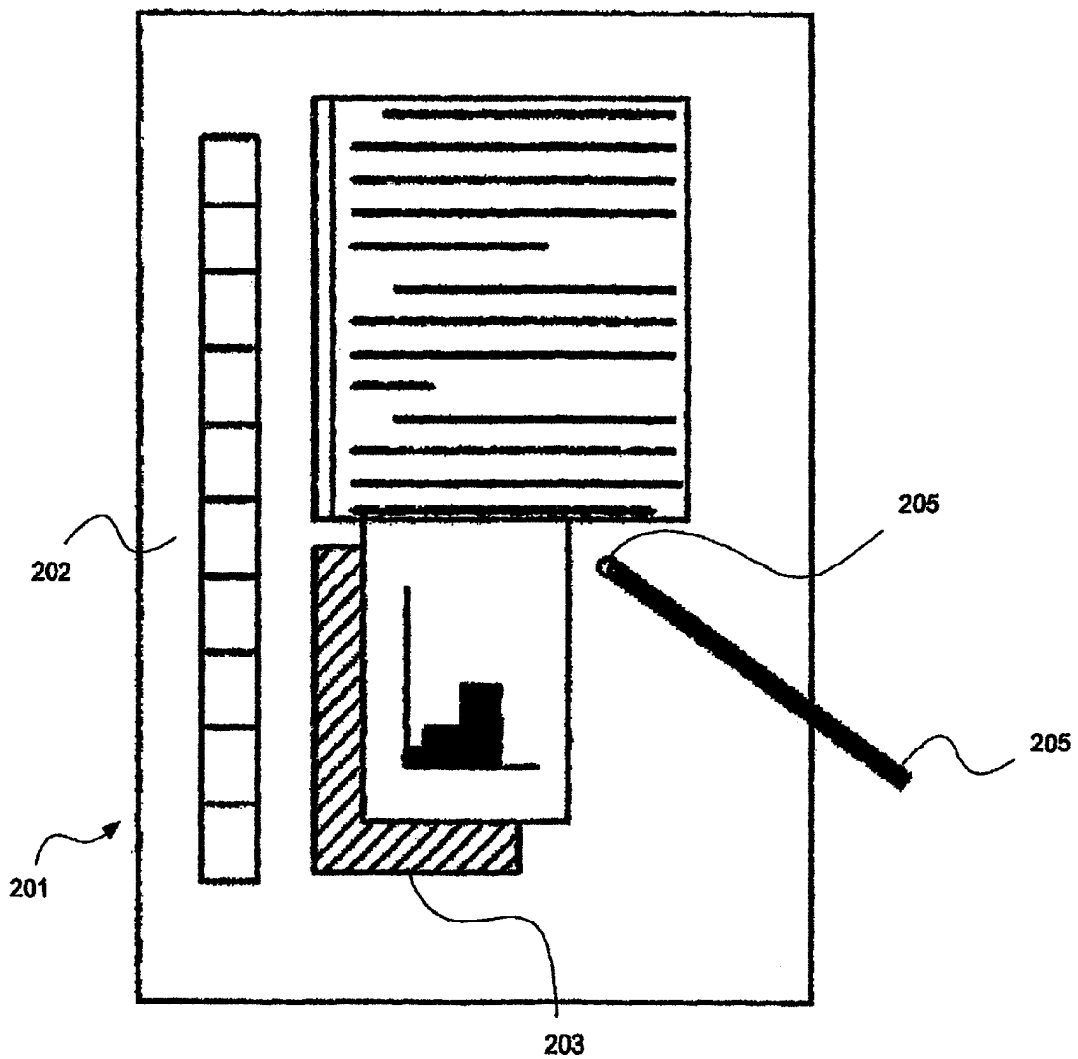
ФИГ. 1k



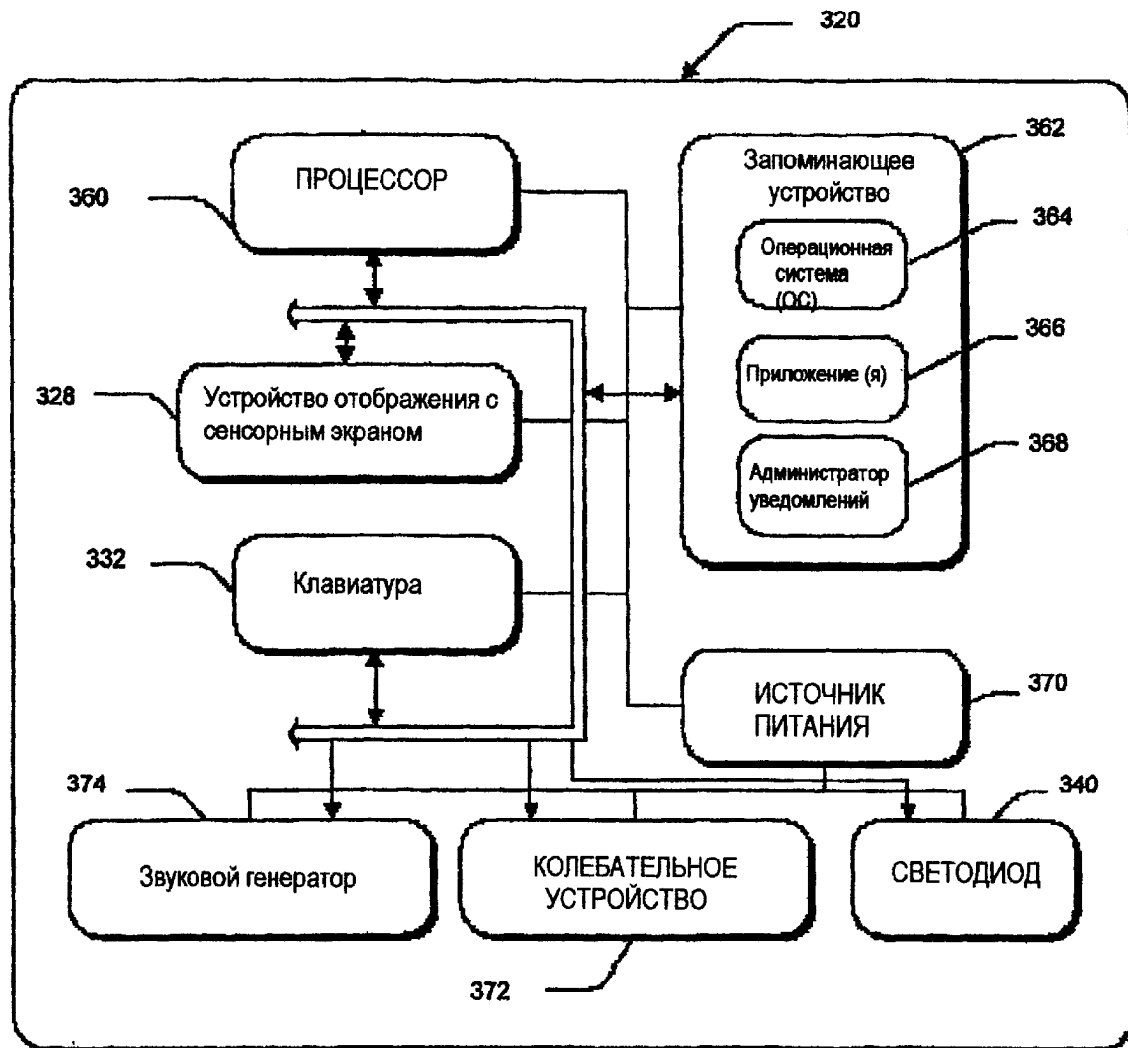
ФИГ. 1l



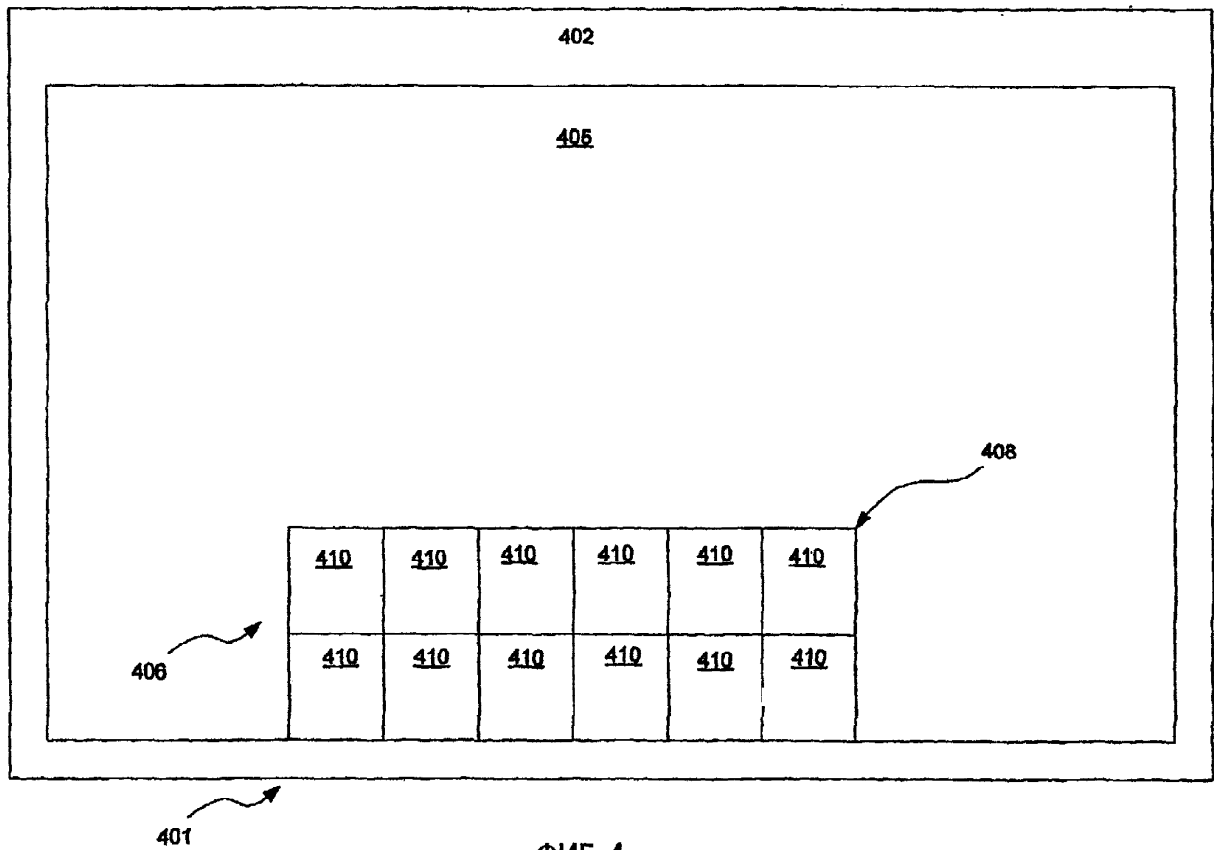
ФИГ. 1m



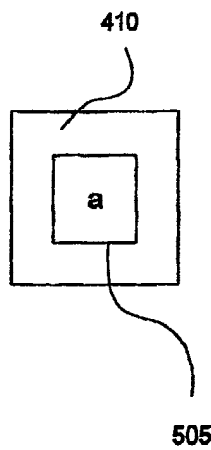
ФИГ. 2



ФИГ. 3



ФИГ. 4



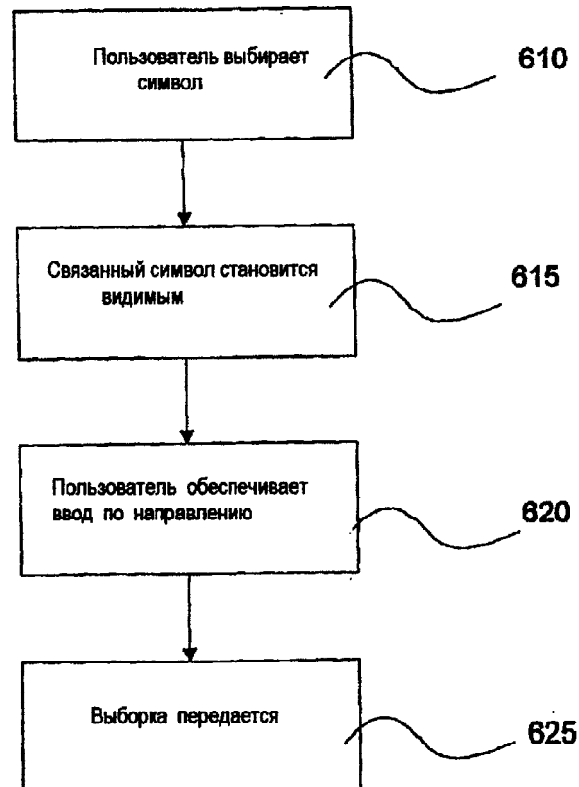
ФИГ. 5

<u>505b</u> À	<u>505a</u> A	<u>505h</u> Â
<u>505c</u> à	<u>506</u> a	<u>505g</u> Ã
<u>505d</u> â	<u>505e</u> ä	<u>505f</u> ã

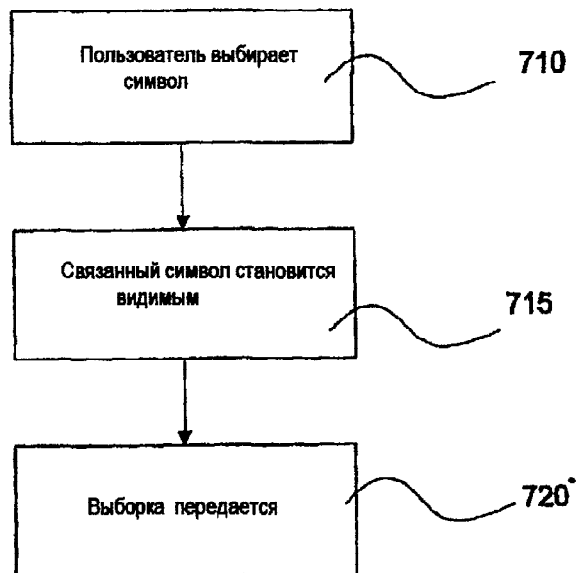
ФИГ. 6a

<u>505b</u> 1	<u>505a</u> 2	<u>505h</u> 3
<u>505c</u> 4	<u>505</u> 5	<u>505g</u> 6
<u>505d</u> 7	<u>505e</u> 8	<u>505f</u> 9

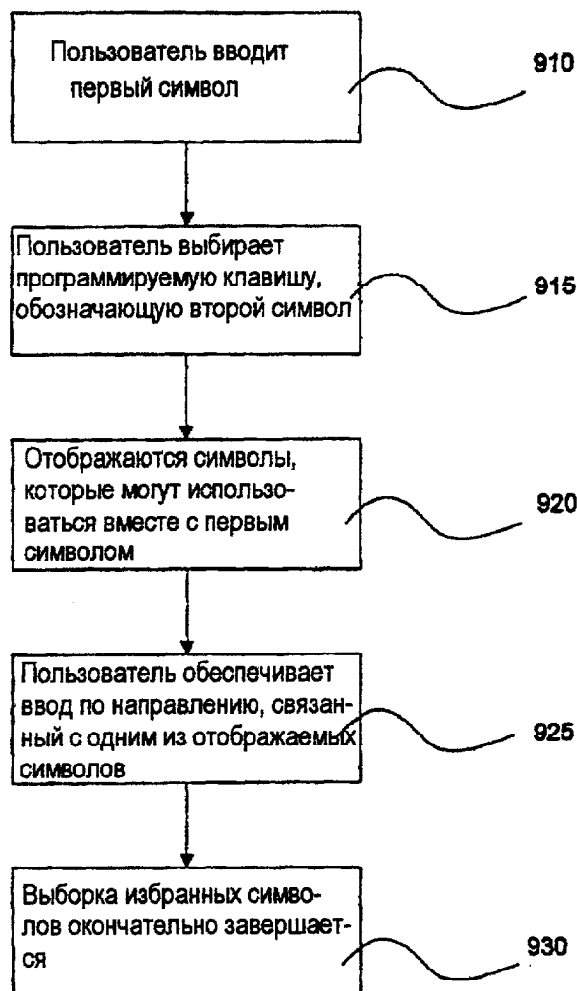
ФИГ. 6b



ФИГ. 7



ФИГ. 8



ФИГ. 9