

**Европейская экономическая комиссия****Конференция европейских статистиков****Шестидесятая пленарная сессия**

Париж, 6–8 июня 2012 года

Пункт 8 предварительной повестки дня

**Группа высокого уровня по стратегическим
разработкам в бизнес-архитектуре статистики****Принципы и руководящие положения построения
многоязычных приложений для официальной
статистики****Записка секретариата***Резюме*

В настоящей записке изложены принципы и руководящие положения построения многоязычных приложений для официальной статистики, разработанные Консультативным советом по совместному использованию элементов статистических информационных систем.

Национальные статистические организации обратились в Консультативный совет по совместному использованию элементов статистических информационных систем с просьбой предоставить информацию о стандартах разработки многоязычного программного обеспечения (ПО). Таких стандартов для статистических приложений не существовало, хотя отсутствие многоязычной поддержки является одним из главных препятствий сотрудничеству в области статистического программного обеспечения. В связи с этим Консультативный совет разработал ряд принципов и руководящих положений, излагаемых в настоящей записке.

Эти принципы и руководящие приложения предназначены для сотрудников и подрядчиков, занимающихся разработкой статистических приложений. По возможности они опираются на более общие стандарты информационных технологий, однако решают конкретные вопросы, с которыми сталкиваются статистические организации. Они были широко изучены в рамках статистического сообщества, занимающегося информационными технологиями. Они призваны дать практические советы в отношении того, как сделать приложения независимыми от языка с самого начала, что естественно намного эффективнее, чем пытаться добавлять многоязычные возможности на более позднем этапе.

Эти принципы и руководящие положения предназначены для распространения исключительно в электронном виде, и рассматриваются в качестве развивающегося документа. Поскольку информационные технологии постоянно эволюционируют, этот документ будет регулярно обновляться с учетом новых технологий и стандартов. Группа высокого уровня по стратегическим разработкам в бизнес-архитектуре статистики утвердила эти руководящие положения и рекомендовала всем статистическим организациям применять их. Для облегчения этой задачи данные принципы и руководящие положения представляются Конференции для информации.

1. Настоящие принципы и руководящие положения подготовлены Консультативным советом по совместному использованию элементов статистических информационных систем Конференции европейских статистиков и секретариатом Европейской экономической комиссии Организации Объединенных Наций (ЕЭК ООН) при содействии и экспертной оценке участников состоявшегося в 2011 году совместного совещания ЕЭК ООН/Евростата/Организации экономического сотрудничества и развития (ОЭСР) по управлению статистическими информационными системами (УСИС). Настоящая версия была опубликована в феврале 2012 года.

I. Введение

2. Усилия по глобальной стандартизации таких моделей, как Типовая модель производства статистической информации (ТМПСИ) и Типовая модель статистической информации (ТМСИ), а также успехи в разработке стандартов обмена данными и метаданными заставили поставщиков статистического ПО обратить внимание на возможности обмена ПО на международном уровне. Это поставило на повестку дня вопрос о том, как можно активно использовать эту перспективу при разработке ПО начиная с первого этапа.

3. Статистическое ПО часто разрабатывается в национальном контексте, а многоязычная поддержка может добавляться позднее. Часто добавление поддержки других языков на наиболее позднем этапе сопряжено с огромными трудностями, если язык с самого начала не считается ключевой составляющей архитектуры ПО.

4. Однако создание ПО, которое будет использоваться во многих странах, не ограничивается исключительно проблемой языка. Применение эффективной практики по интернационализации ПО означает расширение возможностей совместного использования данных в рамках статистического сообщества. Применение этих руководящих положений повысит переносимость и расширит возможности многократного использования ПО.

5. Целями настоящих руководящих положений являются обзор широко распространенных видов передовой практики в деле разработки интернационализованного ПО и описание некоторых существующих в этой области ресурсов и общих стандартов с уделением основного внимания темам, которые имеют конкретное отношение к статистическому ПО, таким как обработка чисел, форматирование, даты, обработка формул и нотация.

II. Определения

A. Интернационализация и локализация

6. Процесс разработки ПО для использования на разных языках, в различных культурных контекстах и странах может восприниматься как два различных процесса: интернационализации (I18N) и локализации (L10N). Интернационализация связана с процессом проектирования ПО таким образом, чтобы оно могло адаптироваться к меняющимся условиям без изменения кода. Локализация связана с более индивидуальным проектированием, например для конкретной страны, языка или региона. Таким образом, если интернационализация призвана обеспечить возможность использования системой нескольких языков, то локализация связана с ее практическим применением, например перевод и т.д.

В. Локали

7. Локаль представляет собой набор установок пользователя, применимых к конкретному языку, стране и/или культуре. Идентификаторы локали обычно состоят из языка, часто в сочетании со страной и иногда с дополнительным параметром, указывающим кодовый набор и модификатор; например, en_GB – это локаль для английского языка в Соединенном Королевстве. Это означает возможность разграничения между американским английским (US) и британским английским (UK). Локаль состоит из ряда элементов, включающих, например, имя и идентификатор языка ИСО, валюту, параметры сортировки, числовые установки, такие как разделители тысяч, используемые календари и другие элементы, такие как направление текста (слева направо или справа налево, горизонтально или вертикально) и т.д.¹.

III. Принципы

A. Принцип 1: ПО должно быть многоязычным по своему замыслу

8. ПО предпочтительно должно проектироваться и тестироваться с целью реализации многоязычного приложения, а не приложения, подвергаемого доводке на последующих этапах. Рекомендуется включать в исходные требования к программному обеспечению требование о содействии использованию нескольких языков. Интернационализацию легче реализовать, если она предусмотрена с самого начала. Если она включена в требования, ее разработка и тестирование являются частью всего процесса разработки. Это позволит получить более качественный продукт, чем в случае добавления языковых требований в конце процесса.

9. В спецификациях требований необходимо прописать, что функции хранения, представления и вывода результатов должны осуществляться одновременно на нескольких языках, например в двуязычной стране. Это оказывает влияние на методику реализации многоязычных приложений.

B. Принцип 2: многоязычные приложения должны поддерживать дополнительные языки без перепроектирования

10. В случаях, когда приложение было спроектировано таким образом, чтобы позволить использование нескольких языков, это должно способствовать добавлению новых языков без необходимости его существенной переделки. По возможности в программе должна быть заложена способность к адаптации для добавления новых языков. Этому также должно содействовать хранение элементов интерфейса пользователя и данных приложения. В изложенных в настоящем документе руководящих положениях анализируются некоторые вопросы, требующие учета.

11. По возможности рекомендуется оснастить многоязычную систему одним набором дистрибутивов для обеспечения целостности кода и снижения стоимости инсталляции, поддержки и т.д. В дальнейшем потребуются вести лишь одну версию. Однако при этом возникает проблема проектирования элементов отоб-

¹ <http://www.jbase.com/new/support/41docs/jBASE%20internationalization.pdf>.

ражения таким образом, чтобы они поддерживали различные языки и длины. Это может также увеличить размер приложения.

12. Там, где возможно, это позволит сократить временной лаг разработки международных версий и повысить качество продукта в целом.

С. Принцип 3: интерфейс пользователя всегда должен быть отделен от кода

13. Отделение интерфейса пользователя от кода позволяет остальной части приложения быть фактически независимой от языка, что имеет ряд преимуществ.

а) Добавление нового языка или внесение изменений в интерфейс пользователя не требует перекомпиляции или перепрограммирования кода.

б) Переводчики могут работать без знания кода или логики программы.

14. Одним из примеров такого обособления может служить использование строковых переменных, а не строковых констант внутри программного кода. В таком случае строковые переменные загружаются на основе выбранной локали или языка.

Д. Принцип 4: определять и адаптировать все элементы интерфейса пользователя

15. Когда речь идет об интерфейсе пользователя, необходимо учитывать, что это подразумевает не только текст, используемый в интерфейсе пользователя. Изменение языка может также влиять на структуру интерфейса пользователя, в частности на кнопки, размер окна приложения и т.д.

16. При проектировании интерфейса разработчик должен учитывать возможные изменения длины текстов в конструкции форм. Длина слов варьируется в зависимости от языка. Это, прежде всего, затрагивает названия элементов, поскольку вариативность длины текста в различных языках сильнее всего проявляется в коротких предложениях или текстах. По мере включения все большего количества слов в предложения различия постепенно уравниваются в пользу более длинного текста. Например, "Click here" ("нажмите здесь" по-английски) (10 символов) соответствует "Klicken Sie hier" по-немецки (16 символов).

17. Ниже приводится руководство по размеру пространства, необходимого для переведенного текста, из "Руководящих принципов понимания вопросов разработки приложений" компании "Oracle".

<i>Количество знаков в английском языке</i>	<i>Необходимое дополнительное пространство</i>
1 знак	400% или 4 знака
2–10 знаков	101–200%
11–20 знаков	81–100%

21–30 знаков	61–80%
31–70 знаков	31–40%
Более 70 знаков	30%

18. Необходимо предусмотреть дополнительное место в меню, названиях элементов и диалоговых формах. Все эти элементы следует повторно протестировать после перевода. Рекомендуется не перегружать текст в формах и приложениях. Это позволяет эффективно располагать переведенные элементы. По возможности для таких элементов, как текстовые поля, контейнеры и т.д., должно использоваться относительное задание размеров элементов, а не фиксированное.

19. Иконки и графические объекты должны также считаться частью интерфейса пользователя; для возможности перевода интерфейса необходимо ограничить использование или вообще не применять встроенного текста в иконках и графических объектах. Любой используемый текст должен быть срисован со строковых переменных.

20. Иконки и графические объекты должны проверяться и переводиться так же, как и строки текста для выявления любых возможных ошибок при толковании. Иконки и графические объекты в разных культурах воспринимаются по-разному. Указание пальцем, например, может считаться оскорблением в некоторых культурах. Цвета могут носить положительную или отрицательную коннотацию в различных культурах. Такие элементы нужно использовать очень осторожно.

21. Клавиши быстрого вызова, иконки меню и т.д. – это все элементы интерфейса пользователя, которые также требуют адаптации. Клавиши быстрого вызова необходимо адаптировать к различным языкам; это требует хранения комбинации клавиш быстрого вызова в качестве части интерфейса пользователя. Клавиши быстрого вызова должны быть доступны также с различных клавиатур. Это требует проверки раскладок клавиатур для различных локалей.

22. На определенном этапе тестирования весь интерфейс пользователя необходимо проанализировать в рамках работы по переводу и представить его группе по локализации.

Е. Принцип 5: планировать хранение многоязычных элементов интерфейса пользователя и управление ими

23. Элементы интерфейса пользователя необходимо поддерживать в доступном и поддающемся отслеживанию формате. Это может быть, например, формат ресурсного файла или таблица базы данных и т.д. Храниться должны, в частности, метаданные об элементах интерфейса, такие как комментарии, контекст, идентификаторы и т.д. Это можно использовать в общении с переводчиками, например для сообщения того, где используется определенный текст, когда он появляется, какую информацию он передает; кроме того это упрощает общение между переводчиками и разработчиками при выяснении конкретных вопросов.

24. Идентификаторы позволяют отслеживать переводимые элементы, а также повышают возможность их повторного использования между версиями и отдельными приложениями. Это позволяет стандартизировать переводы и может

сократить затраты на перевод, если один и тот же текст можно повторно использовать. Например, при работе с расширяемым языком разметки ("XML"), программа Набора тегов для интернационализации ("ITS") рекомендует хранить переводимый текст в элементах, а не в свойствах для обеспечения возможности использования уникальных общих идентификаторов, комментариев и т.д.

25. Строковые элементы, необходимые для разных языков, могут различаться по длине, и, следовательно, по размеру; необходимо обеспечить, чтобы элементы памяти могли расширяться для вмещения строк на различных языках².

26. Поскольку порядок блоков информации может меняться в зависимости от языка, блоки информации должны храниться в виде полных предложений, а не предложений, собираемых из ключевых слов. Также избегайте использования порядковых числительных с количественными, это не всегда легко перевести. Например, вместо "1-ой записью в таблице является" используйте "Запись 1:".

27. Проблему может представлять текст, содержащий переменные данные. Например, "В таблице 10 000 записей" по-румынски будет "Tabelul are 10.000 de înregistrări" с изменением положения переменной между началом и концом текста. Само предложение, например, можно упростить до 10 000 записей -> 10,000 înregistrări. или путем использования в тексте метки нахождения переменной в тексте: "В таблице содержится {1} записей", где {1} будет заменено величиной (10 000) на этапе выполнения.

28. Во всех случаях необходимо указывать язык по умолчанию с тем, чтобы хотя бы какое-то сообщение появлялось при отсутствии текста или изображений. Пользователь должен видеть какое-то уведомление об отсутствии элемента на запрашиваемом языке. В случае отсутствия перевода в логах должна создаваться запись типа "системная ошибка".

F. Принцип 6: определять наилучший способ выбора локали

29. При стандартной реализации локали должна выбираться изначально наиболее подходящая локаль для пользователя. Может использоваться системная или пользовательская локаль по умолчанию операционной системы, символы, выявленные в документе, или, в соответствующих случаях, стандартная опция по умолчанию.

30. Затем пользователям должна предоставляться возможность выбирать и менять свою локаль. Например, язык по умолчанию определяется по стране IP-адреса, однако большинство международных вебсайтов допускают, что пользователь может предпочесть другой язык, и позволяют ему изменять параметр по умолчанию.

31. Необходимо запомнить выбранную локаль и продолжать использовать ее в ходе сессии. В случае, когда пользователя можно идентифицировать, следует сохранить предпочитаемую им локаль и использовать ее в будущем. Например, в случае вебсайта идентификационные файлы могут использоваться для установления по умолчанию предпочтительного языка будущих сессий.

32. В идеале пользователь должен иметь возможность менять локали на любом этапе работы, чтобы не приходилось перезагружать приложения для изме-

² См. руководящие положения 8 и 12, в которых более подробно обсуждаются вопросы сохранения многоязычных данных.

нения локалей. Однако изменение локалей после запуска может приводить к дополнительной нагрузке на сервер (запрос делается более одного раза), и это необходимо учитывать при создании спецификации.

33. Если разрешается менять локаль на последующих этапах, то существующие данные необходимо сохранять, а если это невозможно сделать, то должно появляться предупреждение о возможной утрате данных. Следует предусмотреть возможность отмены изменения локали перед обновлением введенных данных. В идеале статус страницы не должен меняться при обновлении локали; в установках выбора языка следует использовать их родные названия, например английский, французский, или их аббревиатуры ИСО: en, fr, de и т.д. Не используйте национальные флаги для идентификации языков.

34. Пользователи должны иметь возможность выбирать свой собственный метод ввода данных, в частности раскладку клавиатуры, чтобы вводить символы, используя привычные для них комбинации клавиш.

G. Принцип 7: представление данных должно соответствовать принятым правилам локали

35. Для разных локалей представление данных является различным. Ниже приведены самые типичные области, в которых встречаются различия. При разработке ПО необходимо знать и учитывать возможные различия. Следует обеспечить, чтобы приложение не опиралось на различные представления. Это касается, например, отказа воспринимать форматы почтовых индексов для различных локалей. Для обработки различий в представлении следует применять соответствующий формат, определенный согласно локали. В случае наличия нескольких форматов, необходимо сохранять предпочтения пользователя. Выходные данные должны отображать необходимые наборы символов.

1. Примеры различий локалей в форматах представления

- Числа:
 - различаются форматы чисел, такие как расположение десятичных разделителей и т.д.
 - порядковые числительные варьируются в зависимости от языков, например 1-ый и т.д.
 - "billion/trillion" (триллион) в длинной и короткой числовой оси
 - сколько значащих разрядов стоит отображать
- Даты: различные типы форматов, различные используемые календари, различные рабочие дни
- Единицы измерения:
 - британская /метрическая система мер
 - валюта, например символы, расположение условного обозначения валюты
- Ввод и формат текста: текст может отображаться в разных направлениях, например, справа налево
- Формат даты/времени:

- григорианский, тайский календари и т.д.
- часовые пояса
- рабочие дни, например начало недели
- форматы, например, в США мм/дд/гг
- Формат бумаги:
 - формат бумаги в США отличается от форматов бумаги в Европе
 - выходной документ не должен ограничиваться "стандартными" форматами бумаги
- Используемые форматы наименований, званий, например г-н, г-жа; формат имени или фамилии
- Правовые требования:
 - определить, используются ли свойственные для конкретных локалей товарные знаки, логотипы или символы, и узнать, могут ли или должны ли они использоваться в других локалях
 - соответствие регламентам
- Форматы адреса и номера телефона. Это актуально для многих статистических приложений; следует обращать особое внимание на:
 - возможность ввода адреса на желаемых языках
 - порядок полей для адреса, например, сначала указывается название улицы и т.д.
 - наличие и формат почтовых индексов
 - названия стран: локалы предоставят перечни названий стран, которые должны использоваться в отношении стран и географических районов

Н. Принцип 8: данные должны обрабатываться и сохраняться в соответствии с принятыми правилами локали

36. В процессе обработки данных в рамках вашего ПО должны учитываться возможные различия между локалями. В этой связи необходимо отметить, что следующие функции, которые могут использоваться при обработке данных, способны различаться в зависимости от локали.

Округление

- Различия существуют, например, в следующем:
 - сколько нужно использовать значащих разрядов? Например, округление валютных единиц
 - когда данные не сопоставляются между локалями, необходимо использовать округление, установленное в конкретной локали
 - если данные сопоставляются, тогда следует использовать стандартное округление.

- Строки³:
 - ввод и сохранение
 - все ли символы можно вводить и сохранять?
 - позволяют ли различные раскладки клавиатуры это делать?
 - используете ли вы индивидуальные символы для отдельных нажатий на клавиши, которых может не быть на всех клавиатурах?
- Сравнение строк:
 - используемый порядок поиска и сортировки должен соответствовать текущим установкам пользователя
 - при сравнении текста, зависящего от локали (такого как lstrcomp), следует использовать там, где это возможно, зависящие от локали функции сортировки строк. Стандартные функции сравнения строк могут не учитывать локализованные порядки сортировки
 - поиск и другие строковые функции должны осуществляться посимвольно, а не по длине байта ввиду переменной длины символов в байтах
 - когда текст не зависит от локали, следует использовать функции сортировки, не зависящие от локали
- Построчная обработка:
 - переносы строк, пробелы и т.д. различаются между собой; не следует использовать стандартные разбивки в ходе обработки
- Сортировка строк:
 - сортировки различаются в зависимости от культур и алфавитов
 - может существовать более одного порядка сортировки, например порядок сортировки немецкого телефонного справочника
 - порядок сортировки Юникода не подходит для лингвистического порядка сортировки или предполагаемого бинарного порядка сортировки. Не стоит опираться на предполагаемый порядок сортировки
 - необходимо решить, стоит ли разрешать поиск данных по разным языкам. Это может потребоваться в многоязычных странах, таких как Швейцария
 - следует определить основной порядок поиска
 - как обращаться с символами с диакритическими знаками при поиске?
 - необходимо учитывать возможные различия в используемых порядках сортировки баз данных и методах упорядочивания
- Валюта
 - следует хранить количество валюты вместе с идентификаторами валюты
 - следует ли вместе хранить разные виды валюты?

³ См. справочный раздел "Строки" для получения более подробной информации о строках.

- Единицы измерения:
 - следует хранить единицу измерения вместе со значением измерения

37. Самое главное: уделяйте особое внимание скрытым предположениям относительно того, как что работает.

I. Принцип 9: повторно использовать такие стандарты, как библиотеки программ для локализации

38. Следует использовать стандартные локали вместо определения специальных настроек. Стандартную информацию о локализации можно получить через многочисленные библиотеки и ресурсы поддержки разработок. Приложение Юникода "Common Locale Data Repository" (общий репозиторий данных локализации) является примером обширной библиотеки скачиваемой информации о локалях (см. более подробную информацию в справочном разделе).

39. Большинство ПО для разработки содержит стандартные программы локализации, через которые вы можете получить доступ к информации о локалях; например, "setlocale" в "Microsoft Visual studio"; "International Components for Unicode (ICU)" for Java и "C++development".

J. Принцип 10: включить смету расходов на обеспечение многоязычной поддержки в общую смету расходов по разработке

40. Для разработки и тестирования многоязычного приложения потребуется больше времени. Если переводы включаются в ПО, то тогда следует предусмотреть время на перевод в планах разработки ПО. Необходимо включать затраты не только на перевод текста, но и на проверку интерфейса пользователя. Как указывалось ранее, проверьте правильность отображения текстовых строк, правильность их разрыва и приемлемость иконок и графических объектов.

K. Принцип 11: включить подготовку документации в работу по локализации

41. Пользовательская документация должна быть доступна на языке локали. Это необходимо считать частью интерфейса пользователя. Ее перевод может быть включен в работу по локализации. Необходимо, по крайней мере, перевести руководства для пользователей и, по возможности, файлы администратора.

42. Кроме того, в работу по локализации следует включить другие текстовые объекты, такие как справочные файлы, а также системные сообщения.

L. Принцип 12: понимать Юникод и, по возможности, использовать его

43. Внутри символы сохраняются и хранятся в качестве двоичных чисел, а увязка этих чисел с их знаковым значением называется кодировкой. "ASCII" является одним из основных стандартов кодирования. Первоначально знаки хранились в 7 битах памяти и должны были представлять собой стандартный латинский алфавит, цифры и пунктуацию. С доступными мэппингами $2^7=128$.

44. Таким образом, символ "a" может быть представлен следующим образом:

<i>Двоичное представление</i>	<i>Код "ASCII"</i>	<i>Символ</i>
110 0001	97	a

45. В целях представления стандартных символов английского языка были определены наборы кодовых символов от 1 до 127. Коды ниже 32 использовались для системных процессов, коды "ASCII" от 32 до 127 отводились для стандартного набора латинских символов, состоящего из 94 печатаемых символов вместе со знаком пробела.

46. Введение 8 битов для хранения символов означало, что отныне могло храниться до 255 разных символов, включая исходные кодовые мэппинги, т.е. коды от 128 до 255 были "свободными". Это позволило расширить набор символов и включить символы с диакритическими знаками или другие необходимые мэппинги. Таким образом, коды выше 128 по-разному использовались в разных странах. Такая свобода означала, что текст, отправленный с использованием кодировки "ASCII", мог иметь различные представления в различных локалях с одним и тем же двоичным кодом, присвоенным нескольким символам. Для определения и сообщения того, как использовались значения "ASCII" выше 128, были разработаны наборы кодов, например "Latin 5" ИСО (турецкий язык). Однако 8 битов и 255 символов было по-прежнему недостаточно для некоторых языков, таких как китайский, японский, корейский и т.д.

47. Юникод и "ISO 10646" (известный также как Универсальный набор символов (UCS)) были разработаны в качестве международного стандарта, позволяющего последовательно представлять и обрабатывать все символы. Каждый символ имеет определенное число или "кодovую точку", например код "U+0021" обозначает символ восклицательного знака "!". Юникод может сегодня поддерживать более 1 млн. символов с более 100 000 присвоенных символов в текущей версии. Первые 128 символов совместимы с набором символов "ASCII".

48. Внедрение и принятие Юникод позволило разделить кодирование символов и их хранение. Кодовые точки могут храниться с использованием разных систем кодирования.

49. Самые известные системы кодирования включают в себя "UTF-8", "UTF-16" и "UTF-32". При выборе системы кодирования важно учитывать ограничения, налагаемые памятью, запоминающим устройством и необходимостью обеспечить совместимость с предыдущими версиями. Соответствующий краткий обзор с дополнительной информацией об их сравнении представлен на странице "Wikipedia" по адресу: http://en.wikipedia.org/wiki/Comparison_of_Unicode_encodings.

50. В "UTF-8" каждая кодовая точка от 0 до 127 хранится как 1 байт. Кодовые точки со значением 128 и выше хранятся с использованием до 6 байтов. Это значит, что для первых 128 символов кодировка "ASCII" увязывается с "UTF-8". Первый байт в многобайтовом символе показывает, сколько байтов используется для данного символа. Такова стандартная установка для "XML".

51. Кодировка "UTF-16" – это стандарт, используемый для "Windows". Он использует стандартно по два байта для каждого символа. Большинство символов Юникод кодируются через свои кодовые точки. "UTF-16" распространен а зонах, использующих двухбайтовый размер символов, таких как Китай, Япония. "UTF-16" используется в "Java" и "Windows".

52. "UTF-32" использует по четыре байта для каждого символа.

53. Юникод позволяет также использовать более старые системы кодирования; там, где кода не существует в более старой системе кодирования, по умолчанию отображается отсутствующее значение; это устраняет проблему подмены символов в различных системах кодирования. Однако для правильного представления текста необходимо, по крайней мере, чтобы кодируемый текст был специфицирован.

1. Использование Юникод

54. Для использования Юникод в ваших приложениях:

а) применяйте совместимые наборы кодов и сообщайте это набор кодов. Вы должны быть в состоянии читать данные, генерируемые в различных наборах кодов, и правильно обрабатывать данные, например:

i) в адресах электронной почты система кодирования должна указываться в заголовке, как, например: **Content-Type: text/plain; charset="UTF-8"**;

ii) для страниц "HTML" кодировка должна всегда указываться в метатэге в заголовке `<head> <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />`. С декабря 2007 года стандарт "UTF-8" опередил "ASCII" и другие кодировки и стал самой распространенной системой кодирования для вебстраниц;

б) по возможности используйте совместимые с Юникод типы данных и функции.

2. Сохранение данных в формате Юникод

55. Данные должны храниться в типах данных Юникод; например в "SQL" сервер использует "nchar", "nvarchar" и "nvarchar(max)"; вместо их эквивалентов, которые не поддерживаются Юникод: "char", "varchar" и "text". Эти функции и тип хранения позволяют использовать более широкий набор символов. Данные должны сохраняться или преобразовываться соответствующим образом в формате Юникод.

56. Необходимо обращать внимание на случаи, когда данные, хранящиеся в не используемых Юникод типах данных, преобразуются в типы данных Юникод, поскольку это может приводить к проблеме потери данных, что должно помечаться как ошибка при обнаружении.

57. По возможности используйте функции, совместимые с Юникод.

58. Могут иметь место некоторые случаи ухудшения производительности и возникновения различий со строковыми функциями Юникод; хотя они и не носят запретительный характер, для активных операций с текстами может потребоваться оптимизация⁴.

⁴ <http://support.microsoft.com/kb/322112> Сопоставление комплектов "SQL" с комплектовками "Windows". Из-за разницы в правилах сопоставления для данных, не охватываемых Юникод и данных Юникод, при использовании комплектки "SQL" вы можете получить различные результаты при сравнении одних и тех же символов, в зависимости от основного типа данных.

М. Принцип 13: изучить вопрос о том, какие использовать шрифты и как их использовать

59. Существует ограниченный набор шрифтов Юникод для представления большинства, если не всех символов из набора Юникод, которые включают в себя "Arial Unicode MS". Обычно рекомендуется использовать наиболее подходящие шрифты для локали и языка вместо шрифтов Юникод. Шрифт должен быть правильно выбран, чтобы он поддерживал также широкий спектр символов с диакритическими знаками и при этом удовлетворял требованиям проектного решения. Следует использовать простые виды шрифта вместо рукописных или каллиграфических.

60. Некоторые опции для выбора шрифтов используют логические названия шрифтов: например, шрифт без засечек, который определяет шрифт по названию, использует шрифты "True Type" и увязывает шрифты с вашим приложением.

61. Необходимо определить резервный шрифт.

62. Элементы форматирования различаются в зависимости от языка, например, жирный шрифт обычно используется для выделения во многих рукописных шрифтах, но, например, в Кандзи и других рукописных шрифтах это может создать проблему, связанную с удобочитаемостью. Вместо этого можно использовать другие методы выделения, такие как подчеркивание.

IV. Выводы

63. Важность разработки программного обеспечения, которое можно было бы использовать на нескольких языках или в нескольких культурных контекстах, постоянно возрастает. Широкое распространение Интернета за последние десять лет стимулировало коллективное использование ПО. Для извлечения пользы из будущих разработок в рамках статистического сообщества ПО должно разрабатываться с учетом того, что оно будет использоваться в нескольких средах.

64. В настоящем документе изложены некоторые основные рекомендации по разработке ПО, которое может быть адаптировано для многоязычного использования. В нем не ставилась задача составить исчерпывающий перечень потенциальных проблем. В нем изложены широко встречающиеся проблемы и способы их предупреждения и преодоления.

65. Вышеизложенные руководящие положения опираются на анализ документов, касающихся как практических примеров интернационализации ПО, так и исследований в этой области. Мы признаем, что реализация каждого руководящего положения в каждом приложении вряд ли возможна, и поэтому рассматриваем их в качестве обзора передовой практики, позволяющей двигаться вперед.

V. Справочные ресурсы

A. Руководства

"IBM" опубликовала также всеобъемлющее руководство по разработке международного программного обеспечения, которое содержит углубленный анализ проблем, встречающихся при разработке ПО для использования в различных культурных контекстах. Это руководство размещено по адресу: <http://www-01.ibm.com/software/globalization/guidelines/>

"Пошаговая глобализация": <http://msdn.microsoft.com/en-us/goglobal/bb688121>

Проект "Microsoft": "Создание глобальных приложений" ": <http://msdn.microsoft.com/en-us/goglobal/bb895995.aspx>

"Wikipedia": http://en.wikipedia.org/wiki/Internationalization_and_localization

Интернационализация "jBASE". В данной публикации подробно описана работа по интернационализации программного обеспечения "Jbase":

<http://www.jbase.com/new/support/41docs/jBASE%20internationalization.pdf>

Передовая практика локализации "XML": <http://www.w3.org/TR/xml-i18n-bp/>

Часто задаваемые вопросы в отношении "UTF-8" и Юникод для "Unix/Linux": <http://www.cl.cam.ac.uk/~mgk25/unicode.html>

Абсолютный минимум, который должен абсолютно и точно знать каждый разработчик о Юникоде и наборе символов: популярный справочник по Юникод для программистов: <http://www.joelonsoftware.com/articles/Unicode.html>

Описание основных принципов интернационализации, как создавать интернационализированное ПО и как модифицировать и интернационализировать ПО:

<http://www.debian.org/doc/manuals/intro-i18n/>

Основные проблемы, связанные с интернационализацией программного обеспечения: <http://www.acs.org.au/documents/public/crpit/CRPITV32Hogan.pdf>

B. Строки

Сортировка и сравнение строк "Microsoft": <http://msdn.microsoft.com/en-us/goglobal/bb688122>

Руководство "Oracle": лингвистическая сортировка и поиск по строкам: http://download.oracle.com/docs/cd/B19306_01/server.102/b14225/ch5linsort.htm#NLSPG005

C. Контрольные перечни

Контрольный перечень интернационализации "Win32" компании "Microsoft": <http://msdn.microsoft.com/en-us/library/cc194756.aspx>

Контрольный перечень "Oracle" для локализации ПО: <http://developers.sun.com/dev/gadc/i18ntesting/checklists/allquestions/allquestions.html>

Руководящие принципы, контрольные перечни и ресурсы: <http://www.i18nguy.com/guidelines.html>

D. Общие библиотеки и программные ресурсы

"CLDR": Общее хранилище данных локали Юникод: Приложение Юникод "CLDR" представляет собой стандартное хранилище данных о локализации в формате "xml". <http://cldr.unicode.org/>

"GNU C Library": " GNU C Library", широко известная как "glibc", является библиотекой стандарта "C", созданная "GNU Project"; <http://www.gnu.org/software/libc/> Например п. 7.6 касается доступа к информации о локализации: http://www.gnu.org/s/libc/manual/html_node/Locale-Information.html

Платформа "Java": "java.util.Locale"

<http://java.sun.com/developer/technicalArticles/J2SE/locale/>

"ICU": "сложившийся, широко распространенный набор "C/C++" и библиотеки "Java", которые обеспечивают поддержку Юникод и глобализации для программных приложений": <http://userguide.icu-project.org/i18n>

E. Стандарты

Юникод: "Консорциум Юникод" является некоммерческой организацией, занимающейся разработкой, поддержкой и распространением стандартов и данных, касающихся интернационализации ПО, прежде всего "Стандарт Юникод", который описывает представление текста во всех современных программных продуктах и стандартах: <http://www.unicode.org>

Последним опубликованным стандартом Юникода является версия 6, которая содержит 109 000 символов: <http://www.unicode.org/versions/Unicode6.0.0/ch01.pdf>

Домашняя страница Базовой рабочей группы по интернационализации: дает советы по интернационализации другим группам, разрабатывающим вебстандарты: <http://www.w3.org/International/core/>

Статья "Wikipedia" о "ASCII": <http://en.wikipedia.org/wiki/ASCII>

Статья "Wikipedia" о Юникод: <http://en.wikipedia.org/wiki/Unicode>

F. Руководства национальных органов власти

РУКОВОДЯЩИЕ ПРИНЦИПЫ И СТАНДАРТЫ ДЛЯ ДВУЯЗЫЧНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ – Совет валлийского языка: <http://www.byig-wlb.org.uk/english/publications/publications/3963.pdf>

Создание двуязычных приложений в Статистическом управлении Канады (Карен Доэрти, Статистическое управление Канады) <http://www1.unece.org/stat/platform/download/attachments/22478904/issue+4.pdf?version=1>

G. Примеры

Уроки, извлеченные из интернационализации программы оценки доступности вебсайтов: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.13.2698&rep=rep1&type=pdf>

Отдел статистики ЕЭК ООН разрешает скачивание, копирование и распространение данной публикации для ваших личных потребностей или потребностей

вашего работодателя, но исключительно на абсолютно некоммерческой основе. В случае цитирования любой части данной публикации должна указываться ЕЭК ООН в качестве источника. Коммерческое распространение настоящей публикации или любой ее части допускается лишь со специального разрешения. Для получения такого разрешения или за любой дополнительной информацией просьба обращаться в Отдел статистики ЕЭК ООН по адресу: support.stat@unece.org.
