

стирование играет важную роль в тяжёлых приложениях с большим количеством функций.

Плюсы автоматизированного тестирования.

Возможность нагрузочного тестирования. Можно достаточно быстро смоделировать большое количество пользователей.

Экономия времени. Ручное тестирование больших приложений – долгий и трудоёмкий процесс, в то время как сценарии пишутся лишь один раз.

Возможность повторного использования. Тестовый сценарий, написанный один раз, может быть использован и в будущем при очередном обновлении проекта.

Минусы автоматизированного тестирования.

Дороговизна. Инструменты автоматизированного тестирования, а также обучение их использованию стоят недёшево, поэтому нужно тщательно оценивать бюджет.

UI-тестирование. Автоматизированное тестирование не может в полной мере покрыть требования к пользовательскому интерфейсу.

Отсутствие «человеческого взгляда». Возможно существование ошибок, которые заметит только человек.

Заключение.

Оба вида тестирования имеют как преимущества, так и недостатки. Комбинация обоих – идеальный способ получить от тестирования максимальный результат.

УДК 372

Шапочка С. А.

## **ПРОБЛЕМА РУССКОГО ЯЗЫКА В КОНСОЛЬНЫХ ПРИЛОЖЕНИЯХ**

*БНТУ, г. Минск*

*Научный руководитель: канд. техн. наук, доцент Дробыш А. А.*

Приложение – это компьютерная программа. Приложения бывают разными, и отличаются по принципу организации своего пользовательского интерфейса. Интерфейс же в свою очередь – это внешний вид программы. Он бывает нескольких типов, одним из которых является текстовый.

Текстовый интерфейс пользователя (англ. Text user interface, TUI; также Character User Interface, CUI) – это командная строка, то

есть система средств взаимодействия пользователя с компьютером, основанная на использовании текстового режима дисплея или аналогичных устройств например, командной строки. Приложения, использующие текстовый интерфейс командой строки, называют консольными. Консольных приложений много, и они используются часто для того, чтобы упростить действия пользователя в работе с компьютером.

Стандартной 8-битной кодировкой для русских версий в Microsoft Windows до 10-й версии является Windows-1251. Windows-1251 имеет существенный недостаток: отсутствуют символы псевдографики, имеющиеся в CP866 и KOI8 – расширенных версиях ASCII – таблицы, в которой некоторым распространенным печатным и непечатным символам сопоставлены числовые коды (хотя для самих Windows, для которых она предназначена, в них не было нужды, это делало несовместимость двух использовавшихся в них кодировок заметнее). Основное назначение псевдографики – графическое оформление программ с текстовым интерфейсом пользователя (в том числе и так называемых консольных) – отображение в них окон, меню, кнопок и прочих элементов интерфейса, создания рамок и таблиц. Следовательно, из-за этого недостатка при вводе текста программы в консольное приложение на русском языке будет отображаться бессмысленный с точки зрения читателя набор символов.

Существует несколько методов решения проблемы русского языка в консольных приложениях. Самым примитивным методом является использование транслитерации, т.е. передачи текста на русском языке с помощью английского алфавита. Этот метод хоть и решает данную проблему, но не является достаточно эффективным, поскольку, может вызвать некоторые затруднения в прочтении текста пользователем.

Более правильным, но весьма сложным методом является смена шрифта консоли на юникодный, это позволит пользователю работать с текстом на русском языке даже на английской Windows. Чтобы поменять шрифт только для одной единственной программы, в её консоли необходимо нажать на иконку в левом верхнем углу → Свойства → Шрифт. Если же нужно поменять для всех будущих программ, то следует нажать на иконку в левом верхнем углу → Умолчания → Шрифт.

В среде Microsoft Visual Studio правильным решением является уйти от однобайтных кодировок, и использовать Unicode в программе. При этом получится правильный вывод не только кириллицы, но и поддержки всех языков (изображение отсутствующих в шрифтах символов будет отсутствовать, но появится возможность с ними работать).

Метод, который часто рекомендуют – использование конструкции «setlocale (LC\_ALL, "Russian");» У этого варианта имеются свои недостатки. Например, в Visual Studio 2012 введённый текст передается в программу неправильно. Ещё один метод, не использующий Unicode – использование функций «CharToOem» и «OemToChar». Этот метод требует перекодировки каждой из строк при выводе, и (кажется) слабо поддаётся автоматизации.

В заключении следует отметить, что самым эффективным из всех рассмотренных методов решения проблемы русского языка в консольных приложениях является использование Unicode в программах.