**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**Факультет прикладной математики и информатики  
  
  
  
  
  
  
  
  
  
  
  
  
  
КРИВЛЕНЯ АНАСТАСИЯ ВАЛЕРЬЕВНА  
  
**Реферат для защиты лабораторной работы № 1**Студента 2 курса, 13 группы

Минск, 2021

**Оглавление**

[**UNICODE** 3](#_Toc90881978)

[**Строковые типы Windows API** 5](#_Toc90881979)

[**Настройка проекта** 5](#_Toc90881980)

[**Безопасные строковые функции** 7](#_Toc90881981)

[**Заключение** 7](#_Toc90881982)

# **UNICODE**

**Юнико́д**  (чаще всего) или **Унико́д** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Unicode*) — стандарт [кодирования символов](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D0%B1%D0%BE%D1%80_%D1%81%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB%D0%BE%D0%B2), включающий в себя знаки почти всех письменных [языков](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA) мира. В настоящее время стандарт является преобладающим в [Интернете](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82).

Стандарт предложен в [1991 году](https://ru.wikipedia.org/wiki/1991_%D0%B3%D0%BE%D0%B4) некоммерческой организацией «Консорциум Юникода» ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Unicode Consortium, Unicode Inc.*). Применение этого стандарта позволяет закодировать очень большое число символов из разных систем письменности: в документах, закодированных по стандарту Юникод, могут соседствовать китайские [иероглифы](https://ru.wikipedia.org/wiki/%D0%98%D0%B5%D1%80%D0%BE%D0%B3%D0%BB%D0%B8%D1%84), математические символы, буквы [греческого алфавита](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B5%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BB%D1%84%D0%B0%D0%B2%D0%B8%D1%82), [латиницы](https://ru.wikipedia.org/wiki/%D0%9B%D0%B0%D1%82%D0%B8%D0%BD%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BB%D1%84%D0%B0%D0%B2%D0%B8%D1%82) и [кириллицы](https://ru.wikipedia.org/wiki/%D0%9A%D0%B8%D1%80%D0%B8%D0%BB%D0%BB%D0%B8%D1%86%D0%B0), символы музыкальной нотной нотации, при этом становится ненужным переключение [кодовых страниц](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0).

Стандарт состоит из двух основных частей: универсального набора символов ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Universal character set, UCS*) и семейства кодировок ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA" \o "Английский язык) *Unicode transformation format, UTF*). Универсальный набор символов перечисляет допустимые по стандарту Юникод символы и присваивает каждому символу код в виде неотрицательного целого числа, записываемого обычно в шестнадцатеричной форме с префиксом U+, например, U+040F. Семейство кодировок определяет способы преобразования кодов символов для передачи в потоке или в файле.

Коды в стандарте Юникод разделены на несколько областей. Область с кодами от U+0000 до U+007F содержит символы набора [ASCII](https://ru.wikipedia.org/wiki/ASCII), и коды этих символов совпадают с их кодами в ASCII. Далее расположены области символов других систем письменности, знаки пунктуации и технические символы. Часть кодов зарезервирована для использования в будущем. Под символы кириллицы выделены области знаков с кодами от U+0400 до U+052F, от U+2DE0 до U+2DFF, от U+A640 до U+A69F.

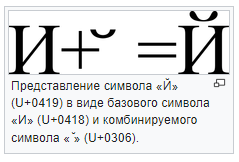
Универсальная система кодирования (Юникод) представляет собой набор графических символов и способ их кодирования для [компьютерной](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) обработки текстовых данных.

Графические символы — это символы, имеющие видимое изображение. Графическим символам противопоставляются управляющие символы и символы форматирования.

Графические символы включают в себя следующие группы:

* **буквы, содержащиеся хотя бы в одном из обслуживаемых**[**алфавитов**](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D1%84%D0%B0%D0%B2%D0%B8%D1%82)**;**
* **цифры;**
* **знаки пунктуации;**
* **специальные знаки (**[**математические**](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)**, технические,**[**идеограммы**](https://ru.wikipedia.org/wiki/%D0%98%D0%B4%D0%B5%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0)**и пр.);**
* **разделители.**

Юникод — это система для линейного представления текста. Символы, имеющие дополнительные над- или подстрочные элементы, могут быть представлены в виде построенной по определённым правилам последовательности кодов (составной вариант, composite character) или в виде единого символа (монолитный вариант, precomposed character). С 2014 года считается, что все буквы крупных письменностей в Юникод внесены, и если символ доступен в составном варианте, дублировать его в монолитном виде не нужно.

 Cимволы в Юникоде подразделяются на базовые ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *base characters*) и комбинируемые ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *combining characters*). Комбинируемые символы обычно следуют за базовым и изменяют его отображение определённым образом. К комбинируемым символам, например, относятся [диакритические знаки](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%BA%D1%80%D0%B8%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5_%D0%B7%D0%BD%D0%B0%D0%BA%D0%B8), знаки ударения. Например, русскую букву «Й» в Юникоде можно записать в виде базового символа «И» (U+0418) и комбинируемого символа « ̆» (U+0306), отображаемого над базовым.

Комбинируемые символы помечены в таблицах символов Юникода особыми категориями:

* Nonspacing Mark — безынтервальный (непротяжённый) знак; таковые обычно отображаются над или под базовым символом и не занимают отдельной горизонтальной позиции (интервала) в отображаемой строке;
* Enclosing Mark — обрамляющий знак; эти символы также не занимают отдельной горизонтальной позиции (интервала) в отображаемой строке, но отображаются сразу с нескольких сторон базового символа;
* Spacing Combining Mark — интервальный (протяжённый) комбинируемый знак; таковые, как и базовый символ, занимают отдельную горизонтальную позицию (интервал) в отображаемой строке.

Особый тип комбинируемых символов — селекторы варианта начертания ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *variation selectors*). Они действуют только на те базовые символы, для которых такие варианты определены. К примеру, в версии Юникода 5.0 варианты начертания определены для ряда математических символов, для символов традиционного [монгольского алфавита](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%BD%D0%B3%D0%BE%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BB%D1%84%D0%B0%D0%B2%D0%B8%D1%82) и для символов [монгольского квадратного письма](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%BD%D0%B3%D0%BE%D0%BB%D1%8C%D1%81%D0%BA%D0%BE%D0%B5_%D0%BA%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D0%BF%D0%B8%D1%81%D1%8C%D0%BC%D0%BE).

# **Строковые типы Windows API**

В WINAPI, в зависимости от того, подключён Юникод или нет, используются два вида строк: UNICODE-ные или TCHAR-ные. Ниже описаны строковые типы данных:

**LPCSTR** – указатель на константную строку, заканчивающуюся нуль-терминатором. От фразы long pointer constant string.

**LPCTSTR** – указатель на константную строку, без UNICODE.  От фразы long pointer constant TCHAR string. Это надстройка функции LPCSTR.

**LPCWSTR** – указатель на константную UNICODE строку. От фразы фразы long pointer constant wide character string. Это надстройка функции LPCSTR.

**LPSTR** – указатель на строку, заканчивающуюся нуль-терминатором. От фразы long pointer string.

**LPTSTR** – указатель на строку, без UNICODE. От фразы long pointer TCHAR string.  Это надстройка функции LPSTR.

**LPWSTR** – указатель на UNICODE строку. От фразы long pointer wide character string. Это надстройка функции LPSTR.

**TCHAR** – символьный тип — аналог char и wchar\_t.

# **Настройка проекта**

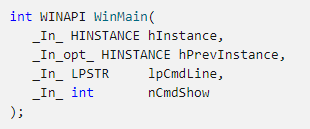
**Windows API** (также известный как API Win32, Windows Desktop API и Windows Classic API) — это платформа на основе языка C для создания Windows приложений. Он существовал уже в 1980-х и использовался для создания приложений Windows в течение десятилетий. Более сложные и удобные платформы были построены поверх Windows API, такие как MFC, ATL, фреймворки .NET. В нашем курсе необходимо иметь Visual Studio, чтобы использовать Windows API.

Создание проекта:

1. В главном меню выберите **Файл,** **Создать,** **Проект**, чтобы открыть диалоговое окно **Создание проекта**.
2. В верхней части диалогового окна задайте для параметра **Язык** значение **C++**, задайте для параметра **Платформа** значение **Windows** и задайте для параметра **Проект** значение **Рабочий стол**.
3. В списке типов проектов выберите **Мастер рабочего стола Windows**, а затем нажмите кнопку **Далее**. На следующей странице введите имя проекта, например *DeskTest*.
4. Нажмите кнопку **Создать**, чтобы создать проект.
5. Откроется диалоговое окно **Project Windows Desktop**. В разделе **Тип приложения** выберите **Классическое приложение (.exe)**. В поле **Дополнительные параметры** выберите **Пустой проект**. Нажмите кнопку **ОК**, чтобы создать проект.
6. В **Обозреватель решений** щелкните правой кнопкой мыши проект*DeskTest*, выберите **Добавить**, а затем выберите **Новый элемент**.
7. В диалоговом окне **Добавление нового элемента** выберите **Файл C++ (.cpp)**. В поле **имя** введите имя файла, например test.cpp. Выберите **Добавить**.

Создание приложения:

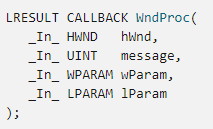
1. Точно так же, как у каждого приложения C и C++ должна быть main функция в качестве начальной точки, каждое Windows классическое приложение должно иметь **WinMain** функцию. WinMain имеет следующий синтаксис:



1. Требуется добавить *<windows.h> , <tchar.h>*



1. Наряду с WinMain функцией, каждое Windows классическое приложение также должно иметь функцию **window-procedure**. Эта функция обычно называется **WndProc**, но вы можете назвать ее по своему усмотрению. WndProc имеет следующий синтаксис:



В этой функции вы пишете код для управления сообщениями, получаемые приложением от Windows при возникновении событий . Например, если пользователь нажмет кнопку "ОК" в приложении, WinAPI отправит сообщение, и вы сможете написать код внутри WndProc функции, который совершает действия при действии пользователя. Это называется обработкой события. Вы обрабатываете только те события, которые относятся к вашему приложению.

# **Безопасные строковые функции**

К стандартным функциям библиотеки **cstring** относится **strcat() (от слова concatenation – соединение).** Если вы используете одну из последних версий среды разработки Microsoft Visual Studio, возможно возникновение следующей ошибки: “error **C4996: ‘strcat’**: This function or variable may be unsafe. Consider using **strcat\_s** instead. To disable deprecation, use \_CRT\_SECURE\_NO\_WARNINGS. See online help for details.” Так происходит потому, что уже разработана новая более безопасная версия функции  **strcat**– это **strcat\_s**.

Она заботится о том, чтобы не произошло переполнение буфера (символьного массива, в который производится запись второй строки). Среда предлагает вам использовать новую функцию, вместо устаревшей. Подобная ошибка может появиться, если вы будете применять функцию **strcpy**.

# **Заключение**

Были рассмотрены строковые типы данных Windows API, развёрнута тема кодировки UNICODE, настроили первый проект WinAPI и было определено, почему strcat\_s более безопасная функция, чем strcat. Автором реферата были усвоены данные темы.

xxxxx

xXXXXXXXXXx

XXXXXXXXXXXXX

xXXXXXXXX XXXx

XXXXXXXXX 0XXXX\\\\\\

xXXXXXXXXXxxXXXX\\\\\\\

XXXXXXXXXXXXXXXX////// \

XXXXXXXXXXXXXXXXX

XXXXX|\XXX/|XXXXX

XXXXX| \-/ |XXXXX

xXXXXX| [ ] |XXXXXx

xXXXX | /-\ | XXXXx

xXXXXX |/ \| XXXXXx

xXXXXXX XXXXXXx

xXXXXXXX XXXXXXXx

xXXXXXXXX XXXXXXXXx

xXXXXXXXXX XXXXXXXXXx

xXXXXXXXXXX XXXXXXXXXXx

xXXXXXXXXXXX XXXXXXXXXXXx

xXXXXXXXX XXX XXX XXXXXXXXx

XXXXXXXX XXX XXX XXXXXXXX

xXXXXXXX XXX XXX XXXXXXXx

XXXXXX XXX XXX XXXXXX

XXXX XXX XXX XXXX

XX XXX XXX XX

XXX XXX

XXX XXX

XXX XXX

XXX XXX

XXXx xXXX

XXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXX XXXXXXX

\_\_\_\_XXXXXX XXXXXX\_\_\_\_

/\_\_\_\_\_\_\_\_/ \\_\_\_\_\_\_\_\_\