

znakové konstanty, dlouhé literály (no)

- standardně typ char (8bitů – základní znaky)
 - znaková sada “interní systému”, překladače, programu
 - znaky (univerzální jména znaků) je možné zadávat pomocí ISO 10646 kódu “F\u00F6rster” (Förster)
 - typ pro ukládání znakových proměnných větších než char (UNICODE-snaha o sjednocení s ISO 10646) – tj. “dlouhé” znakové konstanty – dostatečně velký celočíselný typ (například unsigned int)
 - znaky (char) již proto nejsou konvertovány na int
 - v C je sizeof(‘a’) = sizeof(int) v C++ je = sizeof(char)
 - **klíčové slovo wchar_t** – dostatečně „široký“ typ pro uložení „nejširšího“ locale znaku
 - char16_t, char32_t – typy pro znaky jejichž „šířku“ známe
 - respektuje pravidla stejně velkého celočíselného typu (zarovnání, znaménko...)
-
- wchar_t b = L‘a’;
 - wchar_t c[20] = L”abcd”;
 - existují pro něj nové funkce a vlastnosti například wprintf(), wcin, wcout, MessageBoxW()...
 - řeší výstup zapsaný v “Unicode” do výstupu na základě lokálních nastavení
 - dán normou, ale stále v překladačích rozdílné implementace

znakové konstanty, dlouhé literály (no)

- třída `std::locale` s řešeními pro nastavování a správu informací o národním prostředí (a tedy národní specifika se neřeší přímo ve tvořeném programu)
- `setlocale()` – nastavení pro program – desetinná tečka, datum, čas
- `wcout.imbue(xxx)` napojení výstupního proudu na lokální prostředí (locale xxx)