

První projekt v Code::Blocks

1 IDE

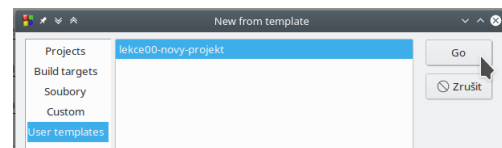
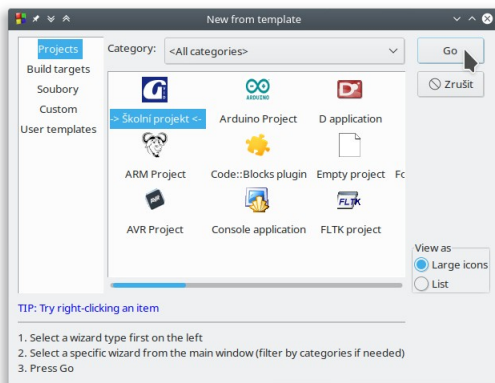
Program Code::Blocks je tzv. IDE (Integrated Development Environment), tedy integrované vývojové prostředí. Je dostatečně jednoduché a nenáročné, aby se v něm začátečník neztratil, ale zároveň má zakomponovány nástroje, které pomáhají při programování, jako je skvěle navržený debugger.

2 Instalace

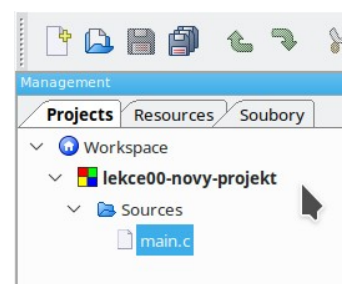
Popis instalace není součástí tohoto dokumentu. Budu předpokládat, že program máte nainstalován a ve funkčním stavu. Pomoci vás tím mohou třeba instruktážní videa s popisem [instalace](#) a [základního nastavení](#).

3 Založení nového projektu

1. File → New → Project...
2. Máte-li nainstalovanou šablonu „Školní projekt“, vyberte ji → Go.
3. Nemáte-li ji, použijte šablonu „Console application“ → Go.
4. Máte-li uloženu vlastní šablonu (viz dále), najdete ji v levém seznamu pod položkou „User templates“ → Go.
 - Pozor! Při této volbě je třeba vyrobit novou složku pro nový projekt.



5. Při pojmenování projektu vyrobí průvodce novou složku stejného jména. Zde budou uloženy všechny soubory, které k projektu patří.
 - **Pozor!** Ve jméně projektu, ani nikde v cestě, která vede ke složce projektu se **nesmí** vyskytovat mezery ani znaky s českou (nebo jinou) diakritikou!
 - Působí to různé problémy - nejčastěji kvůli tomu nefunguje správně debugger, ale problematické to je třeba i při balení do ZIP archivu - na jiném systému bude takový ZIP archiv obsahovat soubory se zmršenými názvy.
6. V okně Management, v záložce Projects je nyní tučně zvýrazněný aktuálně otevřený nový projekt. Pokud je jich otevřeno více, dvojklikem na vybraný projekt se tento stane aktivním - tlačítka Build a Run vždy překládají a spouští aktivní zvýrazněný projekt.
 - **Pozor!** Vždy editujte soubory z aktivního projektu. Pokud editujete soubor z neaktivního projektu a stisknete tlačítko Build, budete překládat něco jiného.
 - **Moje rada:** Ze začátku neotvírejte více projektů zároveň.

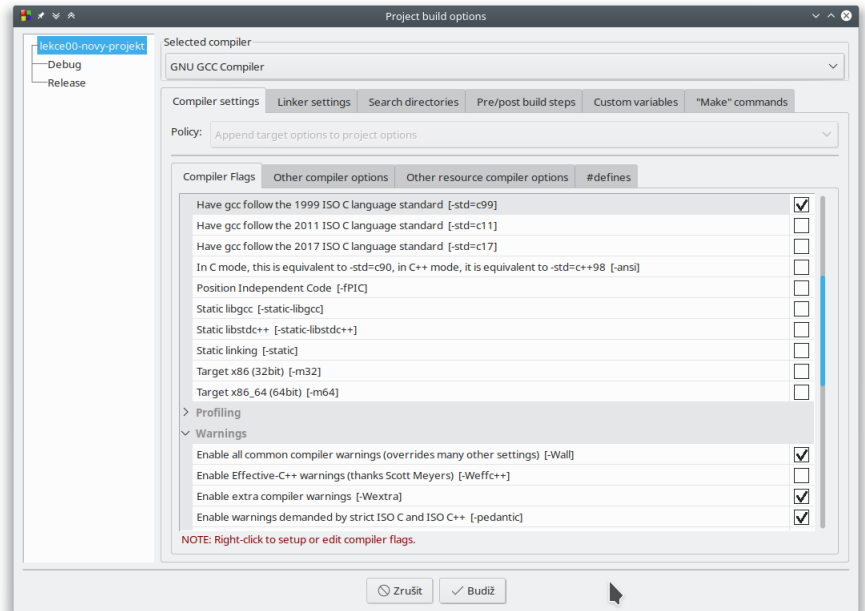


7. Zdrojový soubor nového projektu otevřete v okně Management → záložka Projects → Workspace → jméno nového projektu → Sources → main.c.
 - Poznámka: Zdrojový soubor main.c obvykle obsahuje funkci main (startovní bod programu), ale je to jen konvence, není to nutné. Soubor si můžete přejmenovat.

4 Nastavení překladače pro projekt

1. Pokud je projekt připraven ze šablony „Školní projekt“, je vše připraveno k práci.
2. Jinak je potřeba zkontrolovat nastavení překladače: Project → Build options...

1. Code::Blocks umí překládat jeden projekt více způsoby. Říká tomu překládací cíle (targets). Standardně použijete cíl Debug, kde se specifikuje překlad programu, který budeme chtít ladit pomocí zabudovaného debuggeru (to chceme). Další standardní cíl je Release, pro překlad hotového, odladěného programu, kdy chceme zapnout optimalizace a vypnout generování informací pro ladící program.



2. V okně se objeví nastavení pro celý projekt (pod jeho názvem) a pro cíle Debug a Release.
3. Pro naše potřeby je potřeba mít zapnuty tyto přepínače (pro celý projekt nebo klidně i v cílech Debug a Release):
 - ✓ `-std=c99` - Zapíná modernější variantu jazyka C, která umí například řádkové komentáře `//`, lokální proměnné v cyklech `for` aj.
 - ✓ `-Wall` - Zapíná generování všech možných varovných hlášení při překladu - to chcete, protože vás to upozorňuje na možné problémy.
 - ✓ `-Wextra` - Zapíná další varovná hlášení.
 - ✓ `-pedantic` - Brání používání nestandardních rozšíření jazyka C - ty nechcete používat, protože jinak váš program nemusí fungovat s jiným překladačem.
4. V cíli Debug se pak obvykle přidává parametr `-g` pro generování ladících informací pro debugger. Nedoporučuji zde zapínat optimalizace - některé věci pak v debuggeru mohou být zobrazeny matoucím způsobem.
5. V cíli Release dává smysl zapnout optimalizace, třeba `-O3` a vypnout `-g`.

5 Vytvoření své vlastní šablony

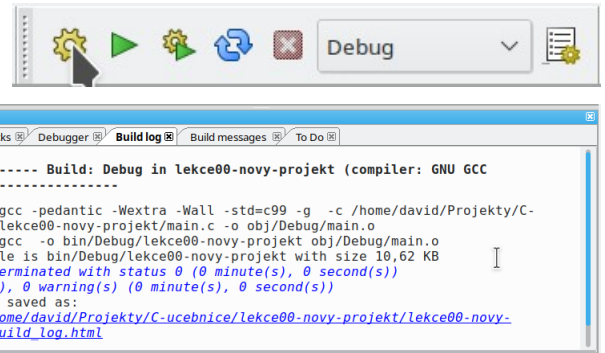
Pokud máte založen nový projekt, pečlivě nastavené přepínače pro překladač, vytvořen zdrojový soubor, ze kterého chcete odvozovat nové projekty (například v něm máte připravenou hlavičku s podpisem, nějaké své často používané podprogramy nebo rovnou moduly), můžete svůj projekt uložit jako šablonu pomocí menu File → Save project as template...

Při vytváření nového projektu pak šablonu stačí vybrat ze seznamu "User templates", viz výše popsáný postup.

6 První spuštění a překlad projektu

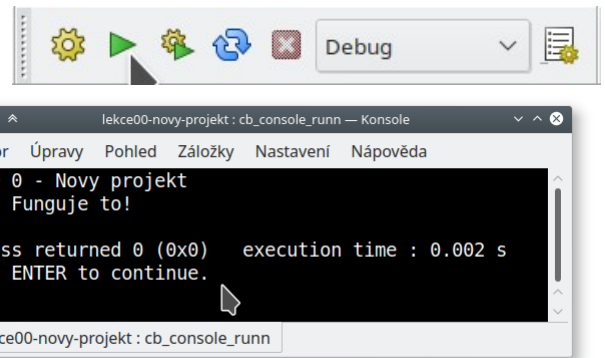
6.1 Překlad

1. Build → Build (Ctrl+F9), nebo ikona s ozubeným kolečkem
2. Po překladu se v záložkách Build log a Build messages (okno s nimi se otvírá/zavírá pomocí F2) objeví hlášení překladače. Pokud jsou v programu chyby nebo překladač tiskne varování, je potřeba program opravit a přeložit znovu.



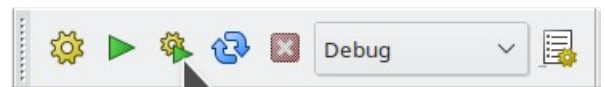
6.2 Spuštění

1. Build → Run (Ctrl+F10), nebo ikona se šipkou/trojúhelníkem.
2. U konzolové aplikace se otevře konzole s výstupem programu. Po skončení programu konzole čeká na stisk klávesy Enter. Nezavírejte ji uzavřením okna! Někdy to ve Windows způsobí zamrznutí přeloženého programu na pozadí a už jej nejde znovu z Code::Blocks spouštět (je to chyba ve Windows). Potom pomůže uzavření programu Code::Blocks a restart počítače (nebo zabití procesu ve správci procesů, když jej umíte používat).



6.3 Překlad a spuštění

1. Build → Build and run (F9), nebo ikona s ozubeným kolem a šipkou.
2. Dělá oba předchozí kroky zároveň.
3. Ne vždy chcete před spuštěním program znovu překládat - překlad může trvat dlouho.



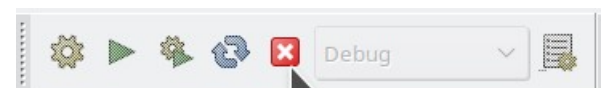
6.4 Složky obj a bin

Při překladu se ve složce projektu objeví složky obj a bin, kam se ukládají přeložené binárky. Samotný přeložený program s koncovkou .exe se uloží do složky bin/cíl/program.exe, kde cíl je buďto Debug nebo Release a jméno programu odpovídá jménu projektu.

Před balením projektu do ZIP archivu tyto složky smažte - zbytečně zabírají místo a ZIP archiv obsahující spustitelné soubory s koncovkou .exe nejde posílat jako přílohu e-mailem (poštovní server to obvykle odmítne z obavy před šířením virů).

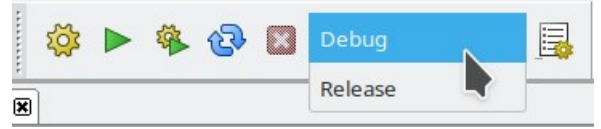
6.5 Ukončení neposlušného programu

Povedlo se vám napsat program, který se zacyklil a odmítá sám skončit? Použijte příkaz Build → Abort nebo ikonu s křížkem. Nevypínejte takový program zavřením konzole - program by mohl zůstat zaseknutý v paměti a Code::Blocks pak nemůže program znovu spustit. Takového zaseklého programu se pak jde zbavit jedině zabitím pomocí správce procesů operačního systému nebo restartováním počítače.



7 Překlad pro ladění a pro vytvoření hotového programu

Code::Blocks dovoluje překládat program s různým nastavením překladače. Říká tomu překladačové cíle - targets. Při vývoji běžně používáme cíl Debug, protože pak můžeme používat ladící program - debugger. Program je pak o něco větší a pomalejší protože přeložená binárka obsahuje dodatečné informace pro překladač. Tento cíl je ve výchozím stavu zapnutý (Build → Select target → Debug nebo v panelu s ikonami pro překlad).



Máme-li program hotový, odladěný a bez chyb, je možné přepnout překladač na cíl Release (Build -> Select target -> Release) a provést znovupřeložení - Build → Rebuild (Ctrl+F11). Tím se všechny části projektu přeloží znovu s novým nastavením překladače. Obvykle máme v tomto cíli zapnutu optimalizace a takto vyrobený program poběží rychleji a bude mít menší velikost. Hotový program se pak bude nacházet ve složce projektu bin/Release/projekt.exe.

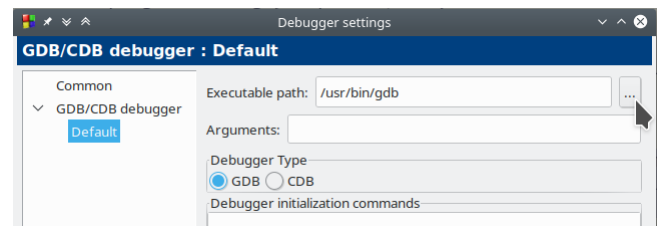
8 Ladění programu - debugger

Code::Blocks umí pomoci s hledáním chyb pomocí ladícího programu - debuggeru. Funguje to tak, že program takzvaně krokujete. Program se zastaví na vybraném řádku a vy si můžete prohlédnout obsahy proměnných, co se vytisklo na konzoli atd. Potom provedete další krok programu, tedy další řádek, a zase se to celé zastaví a vy takto postupně v klidu sledujete, co program dělá a jestli to dělá podle vašich představ. Pokud program nefunguje správně, obvykle takto dokážete celkem rychle najít problém.

Pokud s programováním začínáte, doporučuji používat debugger rutinně. Při sledování toho, co program skutečně dělá se toho naučíte nejvíce.

8.1 Kontrola debuggeru

V menu najděte Settings → Debugger... → Default → Executable path. Toto políčko musí obsahovat cestu k debuggeru. Ve Windows je to program gdb.exe nebo gdb32.exe, v Linuxu program gdb (bez koncovky). Pokud je toto políčko červené, buďto nemáte debugger nainstalovaný, nebo musíte cestu k němu najít a zadat ručně. Ve Windows byste jej měli mít někde ve složce C:\\Program Files\\Codeblocks\\MinGW\\bin.



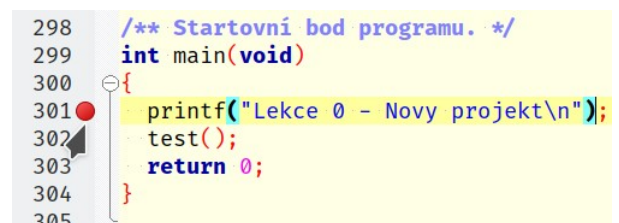
8.2 Kontrola projektu před laděním

Pozor! Ve jméně projektu, zdrojových souborů ani nikde v cestě ke složce projektu se nesmí vyskytovat mezery ani diakritika! V opačném případě debugger skončí, aniž by cokoli udělal.

8.3 Spuštění debuggeru

1. Předpokládám, že pracujete s cílem Debug (tedy program je přeložen s přepínačem -g).

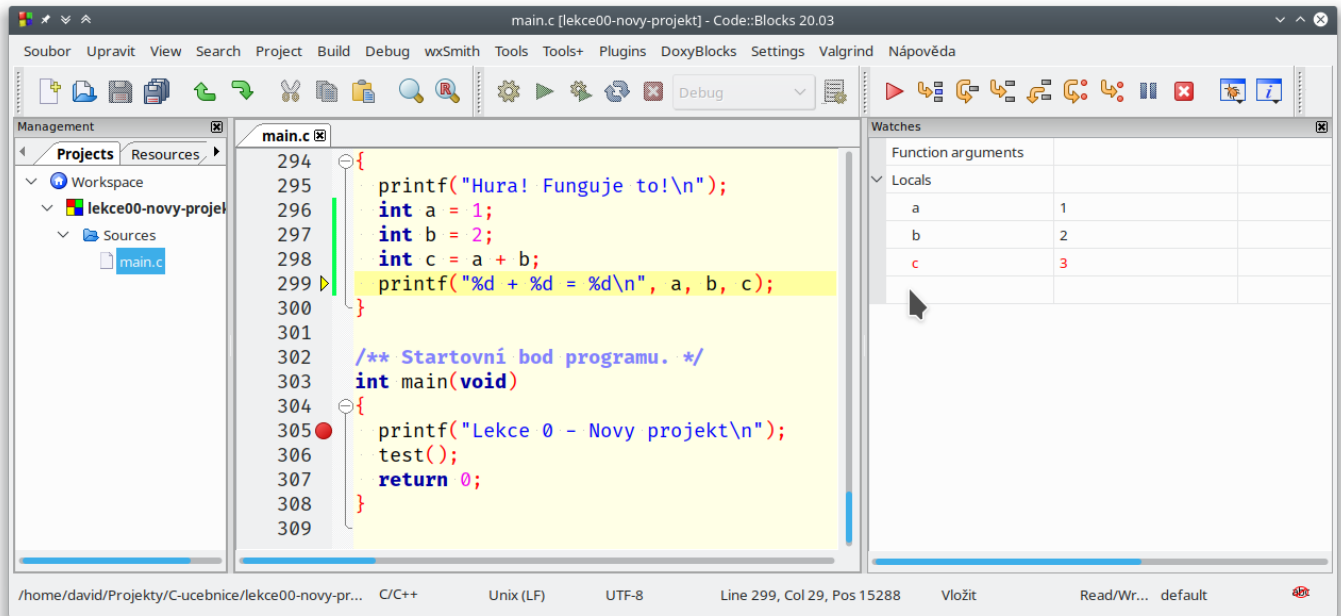
2. Na řádek s kódem, kde chcete zastavit vykonávání programu musíte vložit zarážku. Chcete-li třeba sledovat program od začátku, přejděte na první řádek těla funkce main a klikněte před něj myší (mezi číslo řádku a jeho začátek). Před řádkem by se mělo objevit červené kolečko (stopka).



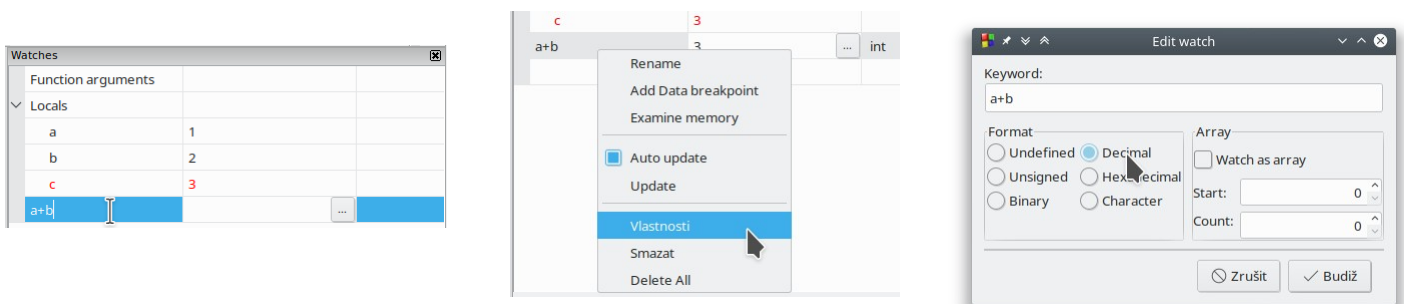
3. Debug → Start / Continue (F8)



- Při spouštění debuggeru se přepne perspektiva (uspořádání oken v Code::Blocks). Pokud si své rozložení uložíte (C::B vám to nabídne dialogovým oknem), při příštím spuštění budete mít okna rozložena stejně.
4. Zapněte okno Watches pro sledování hodnot proměnných Debug → Debugging windows → Watches.
- Toto okno můžete "utopit" v prostředí chycením za jeho záhlaví a posunutím k okraji hlavního okna.

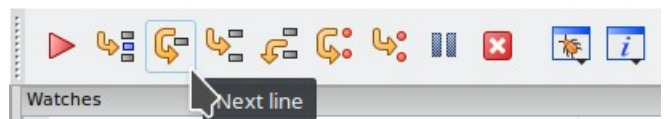


- Lokální proměnné a argumenty funkcí se zobrazují automaticky, ale ne vždy ve vhodném formátu. Pak je možné je přidat na konec seznamu znovu a vyrobit na ně nový pohled.
- Nový pohled na proměnnou nebo dokonce výraz jde do seznamu přidat vyplněním posledního volného řádku. Formát pohledu se mění z kontextového menu: pravé tlačítko myši nad položkou → Properties. Tady jde specifikovat, jak chcete proměnnou zobrazit (binárně, decimálně, hexa, jako pole od určitého indexu).



5. Další krok

- Debug → Next line (F7) - Tohle vykoná řádek a zastaví se na dalším.
- Debug → Step into (Shift+F7) - Pokud je na řádku váš podprogram, skočí do něj a zastaví se na jeho prvním řádku.
- Debug → Step out (Ctrl+F7) - Vykoná vše až do konce aktuálního podprogramu a zastaví se na řádku bezprostředně za jeho voláním.



- Debug → Run to cursor (F4) - Nejprve musíte nastavit textový kurzor na řádek, na kterém se chcete zastavit. Po tomto příkazu se vše před kurzorem provede rychle a zastaví se to na řádku s kurzorem.

6. Zastavení ladění

- Debug → Stop debugger (Shift+F8)
- Perspektiva se přepne zpět na výchozí.

9 Ruční spuštění přeloženého souboru

Uživatelé programy nespouštějí přes Code::Blocks. Výsledkem překladu programu je spustitelný soubor s koncovkou .exe, který najdete ve složce projektu v podsložce bin/cíl/projekt.exe (cíl je buď Debug nebo Release). Tento program jde nakopírovat kamkoli jinam, i na jiný počítač.

Konzolová aplikace se spouští z konzole (ve Windows je to program cmd). Nejprve se musíte pomocí konzolových příkazů dostat do složky, kde se program nachází a pak jej můžete spustit napsáním jeho jména. Prostředí Code::Blocks tohle dělá za vás, když použijete tlačítko/příkaz Run, ale po uživatelích vašeho programu asi nebudete chtít, aby pro spuštění vašeho programu museli používat vývojové prostředí, že?

Pro svůj .exe soubor si také můžete vyrobit zástupce na ploše vašeho systému, ale počítejte s tím, že po skončení programu se konzole automaticky zavře. Program Code::Blocks tomu při spuštění příkazem Run brání použitím vlastního skriptu, který po skončení programu čeká na klávesu Enter, abyste si mohli výsledek svého programu prohlédnout.

10 Soubory projektu

10.1 Přidání souboru do projektu

Prostředí umí fungovat jako obyčejný editor. Například datové textové soubory si můžete připravit přímo v něm a přidat si je do projektu.

10.2 Nový soubor

1. File → New → Empty file - Pak jej můžete pojmenovat s libovolnou koncovkou, třeba .txt.
2. File → New → File... - Tohle vám nabídne šablony pro zdrojové (.c) a hlavičkové (.h) soubory pro přidávání dalších modulů do projektu.
3. Takto vyrobené soubory se standardně ukládají do složky projektu.

10.3 Přejmenování souboru v projektu

Občas potřebujete změnit jméno souboru, který už je součástí projektu.

1. Soubor musí být zavřený.
2. Pravým tlačítkem vyvolat kontextové menu nad souborem v okně Management (v záložce Projects). Objeví se tam položka „Rename file...“.

10.4 Soubory ve složce projektu

Ve složce projektu se nachází vše, co s projektem souvisí. Význam jednotlivých druhů souborů podle jejich koncovek je tento:

- .cbp - Projektový soubor obsahuje nastavení celého projektu. Tento nikdy nemažte. Projekt vždy otevíráte otevřením tohoto souboru. Správně otevřený projekt je vidět v okně Management v záložce Projects.

- `.c`, `.h` - Zdrojové a hlavičkové soubory projektu. Mohou být umístěny i v podsložkách. Nikdy pomocí Code::Blocks neotevírejte zdrojové soubory samostatně - při překladu byste přišli o nastavení dané vaším projektem a nebude vám fungovat správně debugger.
- `.layout` - Soubor popisující vaše rozložení oken v prostředí. Pro funkčnost projektu jako takového není důležitý, ale uchovává například váš skládaný kód a komentáře (viz nastavení Settings → Editor → Folding).
- `.bak` - Záložní soubory, které prostředí automaticky ukládá pro případ nehody. Frekvenci ukládání záložních souborů jde nastavit v Settings → Environment → Autosave.
- `obj`, `bin` - Složky obsahující binárky vznikající při překladu. Ve složce `bin` najdete výsledný přeložený program (ve Windows s koncovkou `.exe`). Tyto složky můžete kdykoli odstranit. Při opětovném překladu se automaticky vytvoří znovu.

Poznámka: Pokud chcete projekt zabalit do ZIP archivu, odstraňte ze složky projektu nejprve složky `obj` a `bin`, případně také záložní soubory `.bak`.