Laboratorul 1

Obiective

- ✓ Să ne configurăm mediul de dezvoltare;
- ✓ Să vedem cum putem folosi GitHub pentru a stoca și gestiona codul sursă;
- ✓ Să vedem unde găsim resurse utile de învățare;
- ✓ Să folosim debugger-ul pentru a rula codul nostru linie cu linie;

Configurarea mediului de lucru

Pentru a putea dezvolta și rula programe scrise în C++, veți avea nevoie de un **compilator** și de un **editor de text** sau **mediu de dezvoltare integrat** (*Integrated Development Environment*, pe scurt *IDE*).

Unele medii de dezvoltare includ deja un compilator ca parte din kit-ul de instalare, deci se poate să nu fie nevoie să instalați separat un compilator.

Compilatoare

- **GNU Compiler Collection (GCC)**

  Instrucțiuni de instalare:

  - Pe Windows, de obicei se instalează ca parte a distribuției de MinGW (*Minimalist GNU for Windows*). O versiune recentă și ușor de instalat a MinGW se poate găsi pe acest site; după extragerea fișierelor într-un director nou, va trebui să adăugați calea subdirectorului `bin` în variabila de mediu PATH.
  - Pe MacOS, se poate instala prin intermediul Xcode.
  - Pe Linux, de obicei se instalează folosind package manager-ul distribuției.

- **Clang**

  Instrucțiuni de instalare:

  - Pe Windows, se poate instala ca parte din suita LLVM.
  - Pe MacOS, se poate instala prin intermediul Xcode.
  - Pe Linux, de obicei se instalează folosind package manager-ul distribuției.

- **Microsoft Visual C/C++ Compiler (MSVC)**

  Instrucțiuni de instalare:

  – Este disponibil exclusiv pe Windows. Se instalează de obicei o dată cu Visual Studio, sau poate fi instalat și separat ca parte din package-ul de Build Tools for Visual Studio.

Editoare de text și IDE-uri

Recomandarea ar fi să vă uitați peste toate opțiunile disponibile pentru sistemul vostru de operare, să le încercați și să alegeți programul în care simțiți că lucrați cel mai bine:

- **CodeBlocks**: poate fi folosit cu oricare dintre compilatoarele de mai sus, sau (pe Windows) puteți instala direct pachetul care include MinGW.
- **VS Code** (împreună cu extensia de C/C++): nu vine cu un compilator inclus, va trebui să aveți deja instalat și configurat un compilator din lista de mai sus.
- **Visual Studio** (doar pentru Windows și MacOS): oferă suport în principal pentru compilatorul MSVC, poate fi folosit și cu GCC/Clang dacă folosiți sistemul de build bazat pe CMake.
- **CLion**: parte din suita de unelte de dezvoltare de la JetBrains, pe care le puteți obține gratuit aplicând pentru o licență de student.
- **Xcode** (doar pentru MacOS): mediul de dezvoltare oferit de Apple. Poate fi folosit împreună cu Clang sau GCC.

GitHub

GitHub este o platformă de găzduire și gestionare a codului sursă, folosită de multe organizații și proiecte pentru dezvoltarea colaborativă de software. GitHub se bazează pe Git, un sistem de versionare al codului sursă[1].

Veți folosi GitHub pe parcursul dezvoltării proiectului de la laborator. Recomandarea mea este să vă puneți pe GitHub și temele/proiectele de la celelalte materii — pe lângă păstrarea lor ca referință, pe viitor le puteți include și în CV.

Vă puteți crea gratuit un cont personal de GitHub apăsând pe butonul de *Sign up* de pe pagina lor principală.

[1] Sistem de versionare al codului sursă și nu numai. Git poate să gestioneze aproape orice tip de fișier, dar se descurcă cel mai bine cu cele text. Creatorul Git l-a numit *the stupid content tracker*

GitHub Student Pack

Pe perioada cât sunteți studenți puteți profita gratuit de GitHub Pro și de multe alte beneficii, ca parte din GitHub Student Developer Pack.

Puteți aplica pentru acest pachet prin intermediul site-ului lor. Va fi nevoie să vă confirmați statutul de student folosind adresa de e-mail instituțională și (posibil) încărcând o poză cu legitimația/carnetul de student.

Utilizarea Git și GitHub

Pentru a face primii pași pe GitHub, vă recomand să urmați instrucțiunile de Quickstart din documentația lor. Dacă vreți să aflați mai multe despre cum funcționează Git sau vreți să găsiți pașii pentru a face un anumit lucru, puteți consulta documentația oficială.

Pentru a sincroniza modificările de pe calculatorul vostru cu proiectul de pe site, puteți instala aplicația GitHub Desktop, care vă oferă o interfață grafică de gestionare a repository-urilor de pe GitHub.

Totodată, editoarele de text și IDE-urile menționate anterior (cu excepția CodeBlocks) oferă integrare cu Git/GitHub:

- Instrucțiuni pentru VS Code
- Instrucțiuni pentru Visual Studio
- Instrucțiuni pentru CLion
- Instrucțiuni pentru Xcode

Referințe

Tehnic vorbind, singura sursă *oficială* de documentație/specificație pentru limbajul C++ este standardul ISO/IEC 14882:2020(E) — însă majoritatea persoanelor nu vor avea nevoie să consulte standardul pentru a scrie un program.

O referință neoficială, dar de calitate, care acoperă aproape integral limbajul C++ (inclusiv C) și bibliotecile standard, este cppreference.com.

Un alt site unde puteți găsi informații și răspunsuri utile este StackOverflow.

Alte resurse de învățare găsiți pe această pagină de pe site-ul oficial al limbajului C++.

Debugging

În cazul în care programul vostru nu compilează, primul pas spre rezolvarea problemei este să citiți și să înțelegeți mesajul de eroare care vi-l dă compilatorul. Puteți încerca de asemenea să căutați pe internet (o parte din) mesajul de eroare primit, ca să vedeți în ce situații este generat și ce reprezintă.

Dacă programul vostru compilează și rulează, dar dă o eroare în timpul execuției sau nu produce output-ul așteptat, va trebui să faceți debugging. Toate editoarele de text / IDE-urile menționate anterior oferă suport pentru rularea programului cu un debugger atașat, care vă permite să executați codul rând cu rând și să vedeți cum evoluează valorile variabilelor la fiecare moment:

- Instrucțiuni pentru CodeBlocks
- Instrucțiuni pentru VS Code
- Instrucțiuni pentru Visual Studio
- Debugging
- Xcode