

Nume:
Grupa:

Subiect Partial

14 decembrie

Aveți voie cu:

- Cartea de Rust - <https://doc.rust-lang.org/book/>
- Documentația bibliotecilor std, clap și cursive (inclusiv tutorialele lor), Tour of Rust
- Suportul de curs și laborator, exemplele de laborator
- Orice IDE **offline**, însă **fără AI** (Copilot, ChatGPT sau altele similare)
- Language server-ul rust-analyzer

Nu aveți voie cu nici un fel de altă documentație online sau offline și nici un fel de aplicație de colaborare. Acestea includ ChatGPT, Copilot, Google, Whatsapp, TikTok.

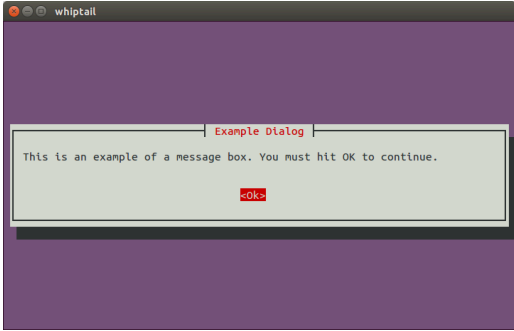
Fraudă:

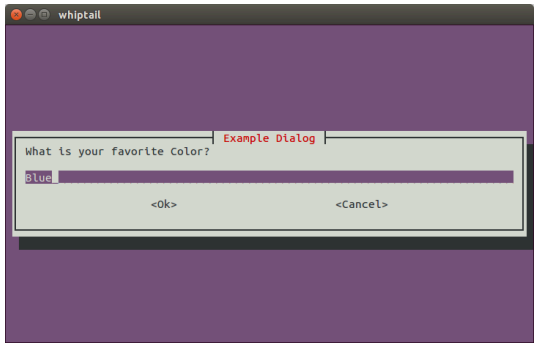
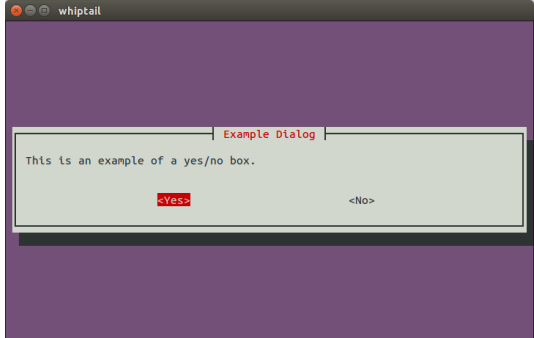
- orice tentativa de fraudare aduce după sine repetarea materiei
- la mai multe tentative de fraudă, se anulează tot examenul și se redă pe hârtie fără documentație (pentru toată seria, mai puțin studenții care au încercat fraudarea care vor repeta materia)

Enunț

Implementați un utilitar similar cu *dialog* (<https://manpages.org/dialog/1>) și *whiptail* (https://en.wikibooks.org/wiki/Bash_Shell_Scripting/Whiptail) în Rust folosind bibliotecile clap (<https://docs.rs/clap/latest/clap/>) și cursive (<https://docs.rs/cursive/latest/cursive/>). Utilitarul va trebui să știe următoarele comenzi:

Lista de comenzi

Comandă	Descriere	Exemplu
message	Afișează un dialog cu un mesaj text și cu un buton OK. 	<pre>\$ dialog-rs message <text_message> <height> <width> [--title <title>] \$ dialog-rs message "The message" 10 20 \$ dialog-rs message "The message" 10 20 --title "The Title"</pre>
input	Afișează un dialog cu un inputbox în care utilizatorul poate introduce un text. Când se apasă pe butonul OK, textul introdus va fi afișat în terminal ca eroare (stderr).	<pre>\$ dialog-rs input <text_message> <height> <width> [--title <title>] \$ dialog-rs input "Enter a string" 10 20 \$ dialog-rs input "Enter</pre>

		<code>a string" 10 20 --title "String"</code>
<code>yes_no</code>	<p>Afișează un dialog cu o întrebare și cu două butoane: YES și NO.</p> <p>Dacă se alege YES, programul va întoarce sistemului eroarea 0. Dacă se alege NO, programul va întoarce sistemului o valoare diferită de 0.</p> 	<pre>\$ dialog-rs yes_no <question> <height> <width> [--title <title>] [--yes <text_buton_yes>] [--no <text_buton_no>] \$ dialog-rs yes_no "Do you know Rust?" 10 20 \$ dialog-rs yes_no "Stii Rust?" 10 20 --yes "DA" --no "NU"</pre>

Notiță: Comanda `dialog-rs <params>` este înlocuită cu `cargo run -- <params>`.

Cerințe

Punctele 1, 2, 3 pot fi rezolvate independent de punctele 4, 5 și 6.

1. Adăugați biblioteca `clap` cu toate feature-urile necesare la proiectul vostru și verificați că afișează mesajul cu parametrii `--version` și `--help`. (1p)
2. Schimbați mesajul *about* afișat la comanda `--help`. (1p)
3. Implementați folosind `clap` parsarea comenzilor `message`, `input` și `yes_no`. Verificați că mesajul de help afișat (de către `clap`) este corect. (1.5p - 0.5p / comandă)
4. Fără a ține seama de ce parametrii ați primit, implementați un message box/dialog folosind `cursive` (fereastră cu chenar și titlu) (1p)
5. Fără a ține seama de ce parametrii ați primit, implementați un input dialog folosind `cursive`. (1p)
6. Fără a ține seama de ce parametrii ați primit, implementați un question dialog cu două butoane folosind `cursive`. (1p)
7. Implementați comanda `message`. (1p)
8. Implementați comanda `input`. (1p)
9. Implementați comanda `yes_no`. (1p)
10. Implementați un parametru global `--title` care permite afișarea unui titlu în partea din stânga sus a ecranului (1.5p).