

[Awww.Inf.22/23L](#) > Zadanie 1: makieta interfejsu aplikacji (UWAGA: termin do godziny 19:00)

ZADANIE

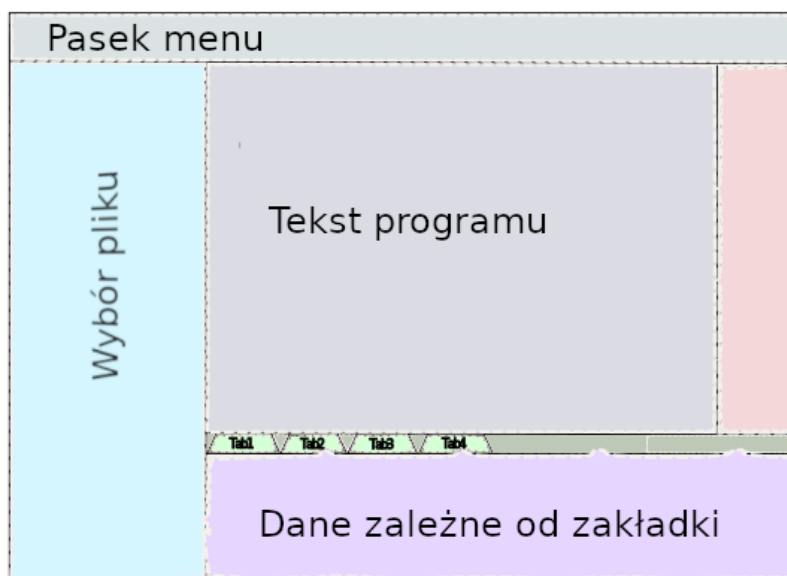
Zadanie 1: makieta interfejsu aplikacji (UWAGA: termin do godziny 19:00)

Otwarto: poniedziałek, 1 marca 2021, 00:00

Wymagane do: poniedziałek, 20 marca 2023, 19:00

Duże zadania tego kursu mają doprowadzić do stworzenia aplikacji webowej, która będzie pozwalała na kompilację programów na 8-bitowe procesory. Aplikacja docelowo ma pozwalać na wprowadzanie pliku napisanego w kodzie C (przez wgranie, ale też i przez edycję), ustawianie opcji kompilatora oraz na uzyskiwanie skompilowanej treści programu. Zakładamy, że kod przeznaczony do kompilacji będzie napisany w C, a kompilacja będzie odbywać się za pomocą kompilatora [SDCC](#)

Celem pierwszego zadania będzie stworzenia ogólnego widoku aplikacji, czyli de facto możliwej do wyświetlenia w przeglądarce jej makiety. Ogólny format strony aplikacji powinien zgadzać się z poniższym rysunkiem.



Pamiętać należy, aby stosowne pola strony były wypełnione przykładowymi danymi. Trochę różnych programów przeznaczonych do kompilacji pod SDCC można znaleźć w repozytoriach:

- [dla 8051](#)
- [dla STM8](#)
- [dla Z80](#)

Widoczne na obrazku pola w intencji mają mieć następującą funkcjonalność.

- Pasek menu - ma służyć na miejsce, gdzie będą pojawiały się typowe dla różnego rodzaju aplikacji menu, np. Plik, Edycja, Opcje itp. Co najmniej te trzy menu powinny być widoczne w makiecie.

- Wybór pliku - ma służyć za miejsce, gdzie będzie się wybierać oraz wstawiać pliki z kodem programów. Obszar ten powinien pozwalać na umieszczenie w nim plików oraz typowej drzewiastej struktury katalogów.
- Tekst programu - ma służyć za miejsce, gdzie będzie wyświetlany pełny kod programu, który ma zostać skompilowany.
- Fragment kodu - ma służyć za miejsce, gdzie będzie wyświetlany wybrany fragment kodu. Na późniejszych etapach działania aplikacji będziemy chcieli, aby np. wyróżnienie w obszarze tekstu programu identyfikatora procedury powodowało wyświetlenie jej kodu w obszarze fragment kodu.
- Tab1 - Tab4 - zakładki z różną zawartością. Z zakładkami tymi wiążemy taką intencję, że będą one grupowały różne kategorie opcji kompilatora, np. w jednym z tabulatorów będzie można wybrać procesor, dla którego kod będzie kompilowany, drugim opcje zależne od wybranego procesora, trzecim standard kodu źródłowego (np. C99 czy C11), czwartym opcje doprecyzowujące format generowanego kodu (np. czy stos ma być zewnętrzny, czy mają być generowane dane pomocnicze do profilowania itp.).
- Dane zależne od zakładki - w polu tym mają pojawiać się wartości zależne od wybranej zakładki. Na przykład po wybraniu zakładki sterującej wyborem procesora w tym miejscu powinna się pojawić możliwość wybrania jakiegoś z dostępnych procesorów.

Oto cennik punktowy za zrealizowanie poszczególnych elementów zadania.

- HTML określający układ interfejsu zgodnie z obrazkiem i wyjaśnieniami (do 5 punktów)
 1. Obecne wszystkie obszary wraz z przykładową zawartością (do 1 p.)
 2. Wygodny układ ekranu przy rozdzielczości 1024x768 lub większej przy założeniu proporcji ekranu 4:3 oraz wygodny układ ekranu przy rozdzielczości 1280x720 lub większej przy założeniu proporcji ekranu 16:9. (do 1 p.)
 3. Łagodne zmiany rozmieszczenia elementów przy zmianie rozmiaru okna przeglądarki (do 1 p.)
 4. Poprawna i pełna struktura dokumentu HTML sprawdzona za pomocą co najmniej jednego z narzędzi, przy czym narzędzie to nie powinno zgłaszać ostrzeżeń: `htmltest`, `tidy`, [html-validate](http://html-validate.com), validator@w3.org, (do 1 p.)
 5. Czysta i czytelna struktura dokumentu HTML (do 1 p.)
- CSS określający wygląd aplikacji (do 5 punktów)
 1. Zastosowane zmienne, które pozwalają na wyświetlanie co najmniej dwóch wersji kolorystycznych strony aplikacji (do 1 p.)
 2. Użycie określenia `medium`, tak aby była dostępna wersja układu aplikacji dostosowana do urządzeń mobilnych (do 1 p.)
 3. Dostępne dwie wersje pliku ze stylem: czytelna dla człowieka i skompresowana (np. z użyciem narzędzia `yuicompressor`) (do 1 p.)
 4. Poprawne i pełna struktura dokumentu CSS sprawdzona za pomocą co najmniej jednego z narzędzi, przy czym narzędzie nie powinno zgłaszać ostrzeżeń, które można wyeliminować: `csslint`, css-validator@w3.org, [stylelint](https://stylelint.io) (do 1 p.)
 5. Dobry styl pisania (DRY, czytelne konwencje nazewnicze, brak `!important`, komentarze wskazujące na nielokalne zależności) (do 1 p.)




[layout.png](#)

20 lutego 2023, 18:15 PM

Edytuj zadanie

Usuń zadanie

Status przesłanego zadania

Status przesłanego zadania	Przesłane do oceny
Stan oceniania	Ocenione
Pozostały czas	Zadanie zostało przesłane 7 dni przed terminem
Ostatnio modyfikowane	poniedziałek, 13 marca 2023, 18:00
Przesyłane pliki	<div> sb438247.zip 13 marca 2023, 18:00 PM</div>
Komentarz do przesłanego zadania	▶ Komentarze (0)

Informacja zwrotna

Ocena	10,00 / 10,00
Ocenione dnia	niedziela, 19 marca 2023, 12:35
Ocenione przez	KR Krzysztof Rogowski
Komentarz zwrotny	Ok

Skontaktuj się z nami



Follow us



Skontaktuj się z pomocą techniczną

Jesteś zalogowany(a) jako Stanisław Bitner (Wyloguj)

Podsumowanie zasad przechowywania danych

Pobierz aplikację mobilną

Pobierz aplikację mobilną

Wspierane przez Moodle

Motyw został opracowany przez

conecti.me

Moodle, 4.0.4+ (Build: 20220922) | moodle@mimuw.edu.pl