



Studio projektowe

Projekt kompilatora online

Autorzy:
Piotr Karaś
Patrycja Kopacz

Prowadzący
dr hab. Konrad Kułakowski

1. Cel systemu

Celem projektu jest stworzenie strony internetowej umożliwiającej napisanie kodu w przykładowym języku programowania, jego skompilowanie i uruchomienie. Dlatego też będzie ona musiała posiadać edytor przystosowany do programowania, w którym użytkownik będzie mógł napisać swój kod. Po stworzeniu programu będzie można go skompilować i uruchomić – zostaną wtedy wyświetlone ewentualne błędy, a jeśli takowych nie ma – wyjście. Program jak i wyjście będzie można pobrać na swój komputer.

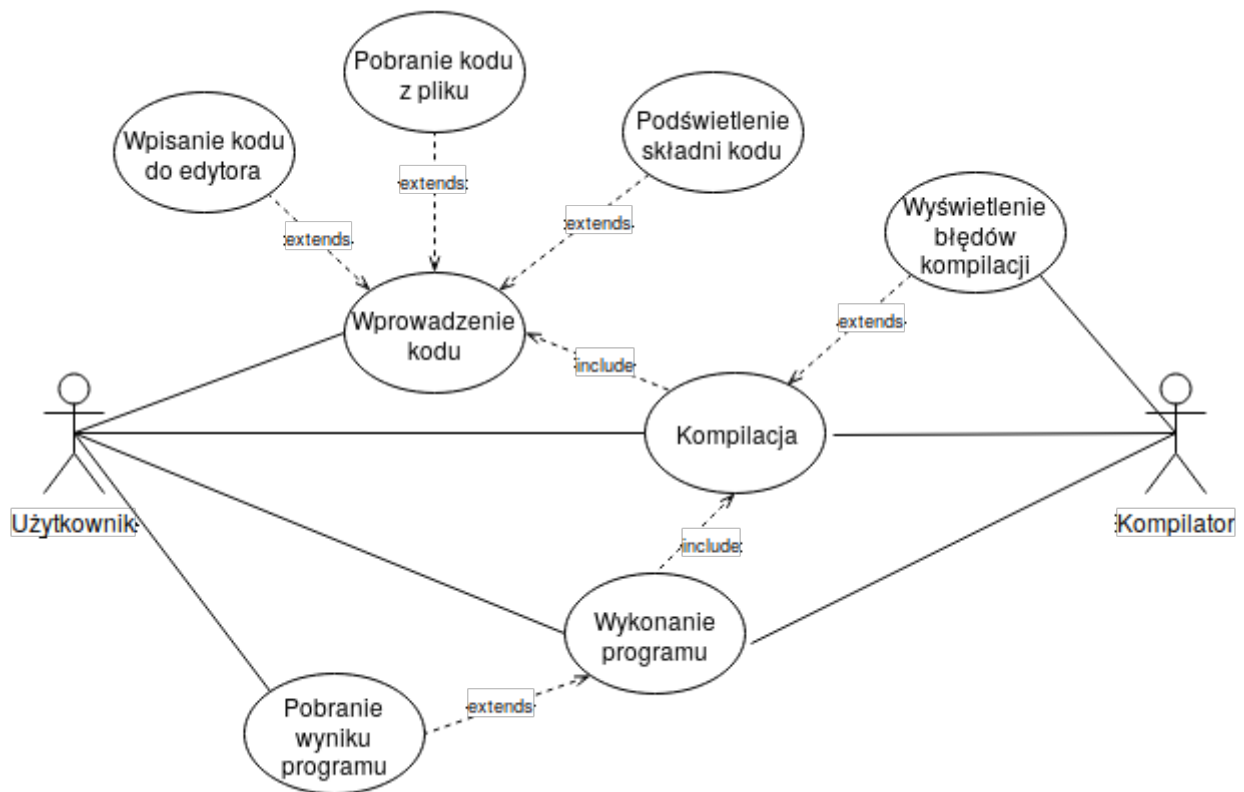
Program powstanie z myślą o uczniach dopiero rozpoczynających swoją przygodę z programowaniem. Będą oni mieli do swojej dyspozycji mały przewodnik po danym języku programowania, a w razie błędów program będzie wyświetlał im użyteczne porady oraz wskazówki. Sporym ułatwieniem będzie dla nich również fakt, że nie będą musieli instalować na swoim komputerze żadnego dodatkowego oprogramowania, co dla początkujących programistów bywa uciążliwe, a sama konfiguracja środowiska – trudna i skomplikowana. Program będzie też użyteczny dla programistów nie mających dostępu do swojego komputera. A dla osób, które szukają języka programowania odpowiedniego dla ich potrzeb, będzie możliwość utworzenia kilku przykładowych programów.

Interfejsem końcowym naszego systemu będzie udostępniona strona internetowa umożliwiająca skompilowanie wprowadzonego kodu źródłowego. Użytkownicy będą mieli dostęp do opisanych poniżej funkcjonalności systemu.

2. Lista możliwości systemu

- ➔ ręczne wpisanie kodu do edytora
- ➔ pobranie kodu z wprowadzonego pliku
- ➔ kompilacja kodu
- ➔ przyjmowanie wejścia od użytkownika dla uruchomionego programu
- ➔ podświetlanie składni (słów kluczowych)
- ➔ więcej niż jeden dostępny język programowania
- ➔ podświetlanie linii, w której wystąpił błąd podczas kompilacji
- ➔ możliwość personalizacji wyglądu strony – zmiana tematu
- ➔ wybieranie flag dla użytkownika
- ➔ rodzaj samouczka dla uczących się: wskazówki, porady
- ➔ przykładowe kody źródłowe
- ➔ wyświetlenie wyniku programu
- ➔ pobranie programu wynikowego

3. Diagram przypadków użycia



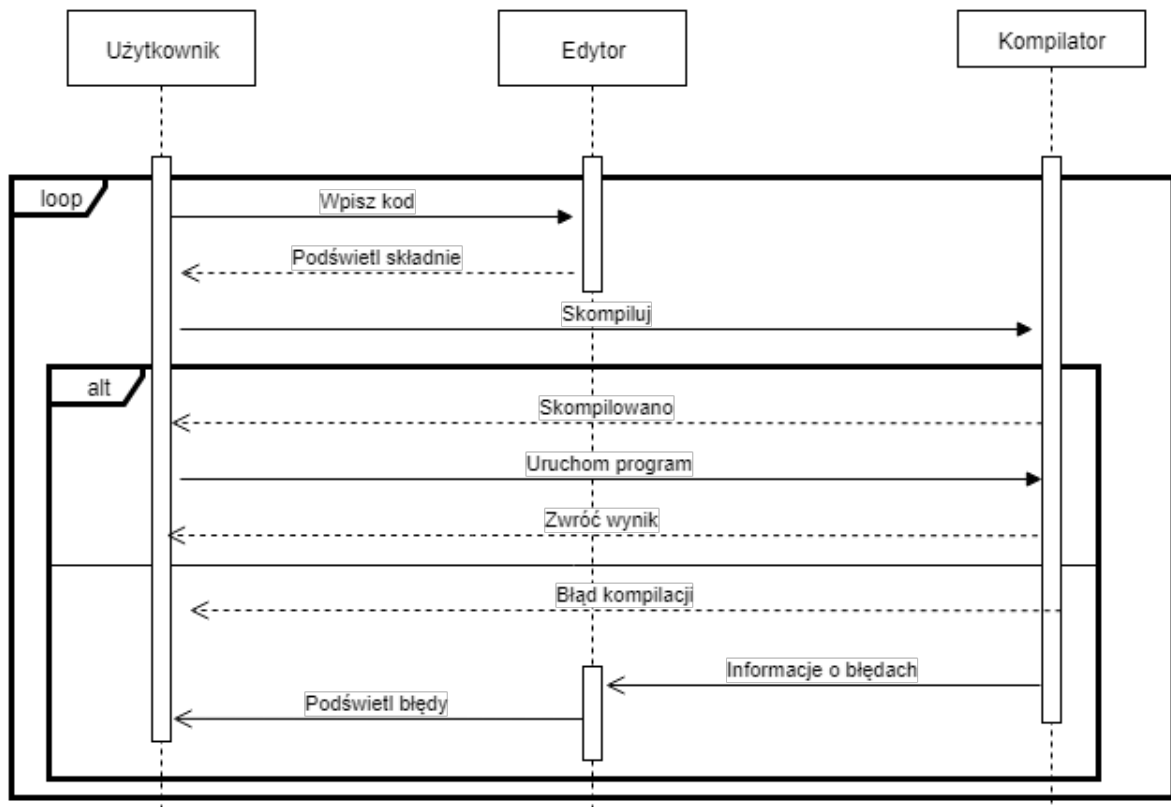
➔ Aktorzy

1. Użytkownik, czyli osoba korzystająca z kompilatora online
2. Kompilator, czyli nasz system

➔ Scenariusze przypadków użycia

1. Wprowadzenie kodu
2. Kompilacja
3. Wykonanie programu
4. Pobranie wyniku programu
5. Wyświetlenie błędów kompilacji
6. Wpisanie kodu do edytora
7. Pobranie kodu z pliku
8. Podświetlenie składni kodu

4. Poglądowy diagram sekwencji programu



5. Architektura systemu

Wykorzystamy technologię REST-ową. Komunikacja między serwisem webowym a serwerem będzie się odbywała za pomocą JSON-ów.

Po otrzymaniu zapytania o skompilowanie danego kodu serwer będzie musiał go zapisać w postaci pliku, następnie uruchomić proces kompilacji, z którego pobierze wyjście. W zależności od zwróconego wyniku pokaże użytkownikowi, w którym miejscu popełnił błąd lub też pozwoli mu uruchomić program.

Jeśli wszystko przebiegło zgodnie z planem, program działa i zostanie podjęta taka decyzja projektowa - użytkownik powinien mieć zapewnioną komunikację ze swoim dziełem, zatem będzie mógł wprowadzać swoje dane jako wejście i wyświetlony będzie miał rezultat. Serwer będzie musiał również zadbać o odpowiednie zatrzymanie programu, jeśli straci łączność z użytkownikiem.

6. Technologie, które zostaną wykorzystane do stworzenia systemu

- ➔ Backend naszej strony będzie zaimplementowany w Javie 10 przy pomocy Springa (oraz Maven'a)
- ➔ Frontend natomiast zostanie napisany przy użyciu HTML, Javascript, jQuery, CSS, Velocity, Bootstrap, Atlassian AUI

- ➔ Backend i Frontend będą komunikować się ze sobą za pomocą protokołu HTTP
- ➔ JSON jako format zapisu danych

7. Narzędzia, które zostaną wykorzystane przy tworzeniu systemu

- ➔ IntelliJ – zintegrowane środowisko programistyczne
- ➔ Brackets – darmowy edytor tekstu dedykowany tworzeniu dokumentów HTML, wyróżnia się wbudowaną opcją podglądu dokumentu na żywo podczas jego tworzenia
- ➔ Github – online repozytorium kodu źródłowego, które umożliwia współtworzenie kodu ze wsparciem gita
- ➔ Trello – narzędzie wykorzystywane przez zespoły do współpracy podczas projektów, umożliwia przypisanie zadań oraz śledzenie postępów w pracy

8. Harmonogram prac

- ➔ 15.10.2018 – stworzenie wstępnych założeń projektu
- ➔ 29.10.2018 – pełna specyfikacja projektu
- ➔ 05.11.2018 – zaimplementowany serwer wraz ze wstępnym widokiem strony
- ➔ 03.12.2018 – działający kompilator umożliwiający skompilowanie na już w pełni działającej stronie internetowej
- ➔ 10.12.2018 – wykonanie testów sprawdzających
- ➔ 17.12.2018 – gotowy projekt wraz z dokumentacją