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.

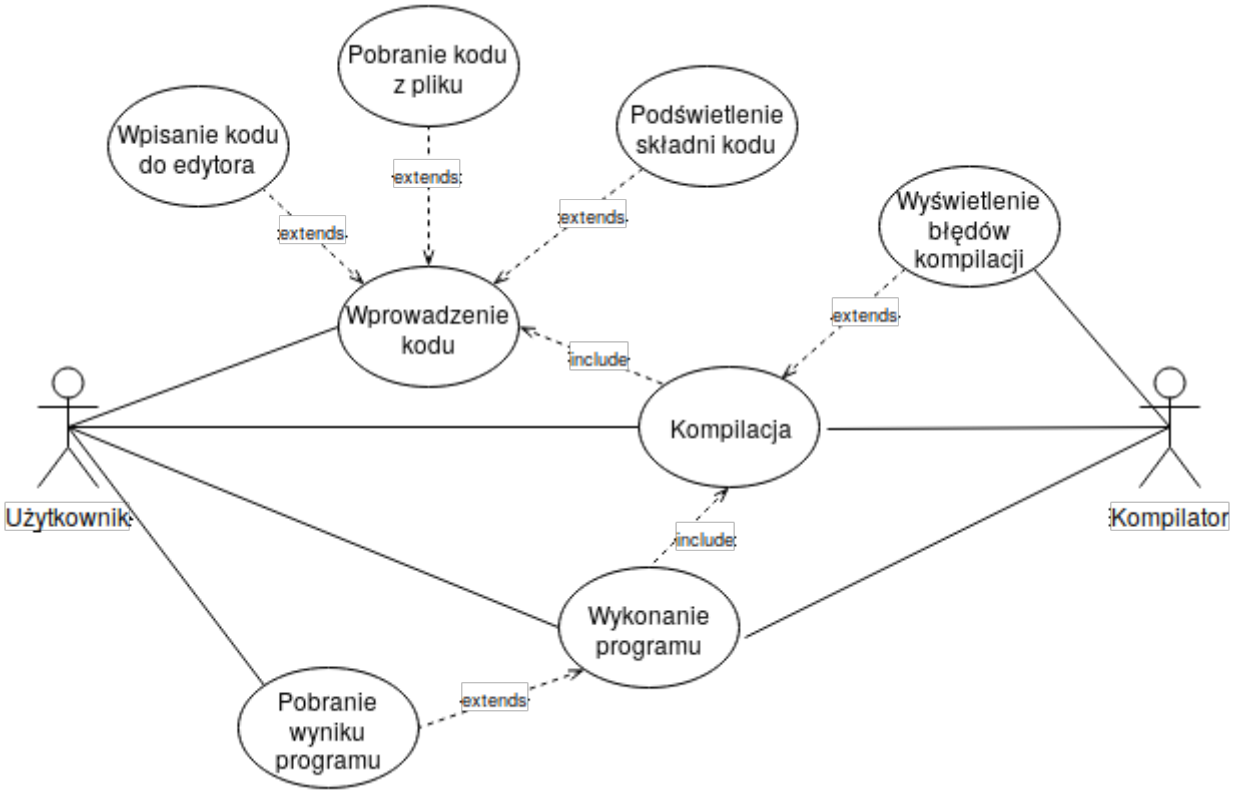
Program powstanie z myślą o uczniach dopiero rozpoczynających swoją przygodę z programowaniem. Będą oni mieli do swojej dyspozycji mały przewodnik po danym języku programowania, a w razie błędów program będzie wyświetlał im użyteczne porady oraz wskazówki. Sporym ułatwieniem będzie dla nich również fakt, 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 otworzenia 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 linijki, 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. W założeniu system jest przeznaczony dla osób uczących się, które wcześniej nie zetknęły się z danym językiem programowania.
2. Kompilator, czyli nasz szeroko pojęty system – warstwa webowa jak i serwer.

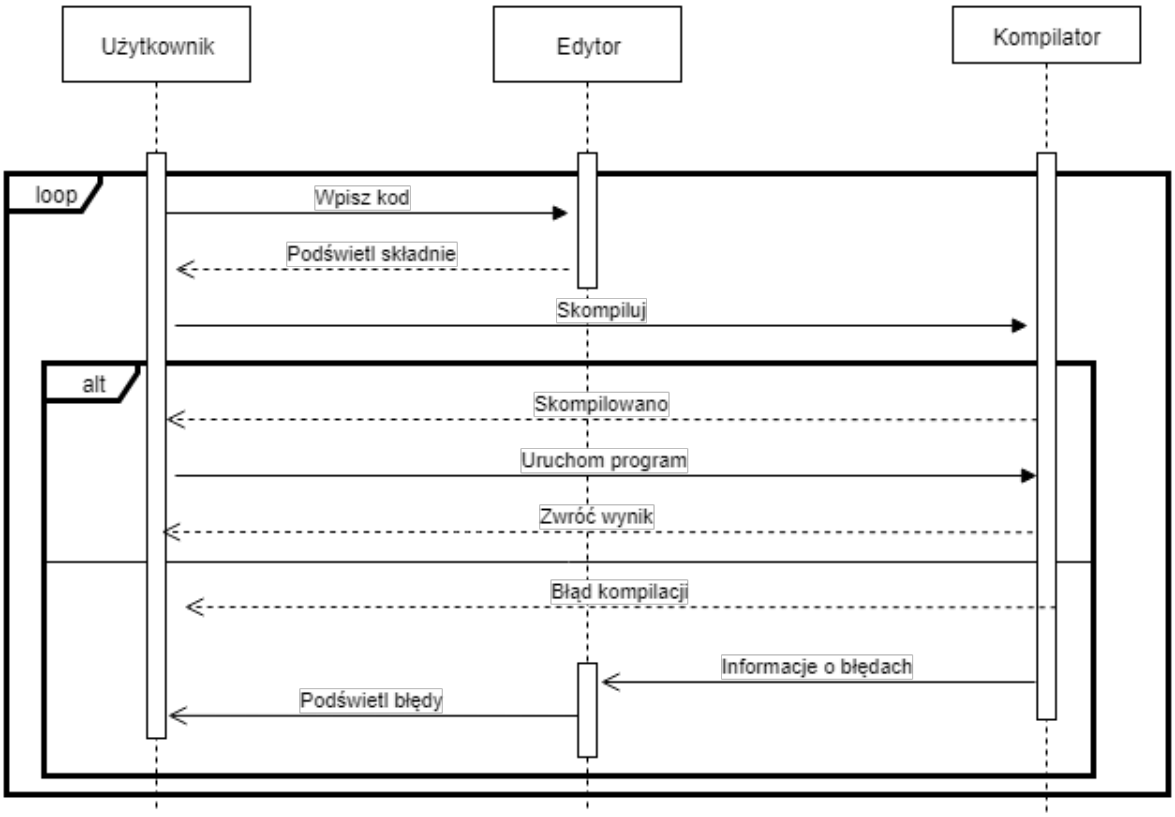
➔ Przypadki użycia

1. Wprowadzenie kodu – ogólnie rozumiane wprowadzenie kodu – od napisania własnego, do pobrania z istniejących szablonów.
2. Wpisanie kodu do edytora – użytkownikowi zostanie udostępniony edytor, dzięki któremu będzie miał możliwość napisania swojego programu w łatwy sposób.
3. Pobranie kodu z pliku – użytkownik będzie miał możliwość pobrać przykładowe programy, tak aby mógł się uczyć na przykładach. Będzie też możliwość pobrania „szablonu” programu, do którego użytkownik będzie musiał coś dopisać, aby program działał.
4. Podświetlenie składni kodu – oraz inne narzędzia wspomagające pisanie kodu, a będące zaimplementowane przez kompilator. Poza

podświetlaniem składni może to być zmiana tematu, auto-wcięcia, podpowiedzi gdzie użytkownik mógłby wprowadzić zmiany, aby kod był czytelniejszy.

5. Kompilacja – po wprowadzeniu kodu użytkownik może go skompilować – zatem wysłać do kompilatora swój kod, aby ten go skompilował.
6. Wyświetlenie błędów kompilacji – po kompilacji użytkownikowi zostanie pokazany status z jakim zakończył się ten proces – jeśli coś poszło nie tak to zostanie podświetlona linijka, która spowodowała błąd.
7. Wykonanie programu – jeśli kompilacja przebiegła zgodnie z planem program można wykonać – użytkownik powinien zobaczyć rezultat uruchomienia swojego kodu, a także móc wejść z nim w interakcję poprzez zapewnienie swojego wejścia.
8. Pobranie wyniku programu – użytkownik może pobrać wynik programu jak i sam program – napisany przez siebie kod.

4. Poglądowy diagram sekwencji programu



5. Diagram przedstawia podstawową funkcjonalność naszego projektu i to w jaki sposób jest ona dostarczana użytkownikowi. Proces zachodzący podczas korzystania z naszej aplikacji jest następujący – początkowo użytkownik wprowadza swój kod lub go pobiera, korzysta przy tym z naszego edytora (podświetlanie składni itp). Po zakończeniu tej części następuje próba kompilacji – kod wysyłany jest do serwera. Następnie mogą się zdarzyć dwie

różne rzeczy: program jest napisany błędnie i proces kompilacji zakończy się błędami – w takim wypadku edytor otrzyma informację w której linijce nastąpił błąd i ją podkreśli lub też wszystko przebiegło poprawnie – wtedy użytkownik otrzyma informację o sukcesie. Po takiej informacji może on uruchomić swój program i zobaczyć jego rezultat.

6. Działanie 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. Użyteczne byłoby też zapewnienie bezpiecznego środowiska dla uruchamianego programu, tak aby użytkownik nie mógł wpłynąć w żaden negatywny sposób na nasz system.

7. 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

8. 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

9. 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