Projekt

Proces ETL – Ceneo.pl

Grupa dziekańska: KrZUIs1011

Grupa projektowa:23

|  |  |  |  |
| --- | --- | --- | --- |
| Imię | Nazwisko | Numer albumu | Wkład w prace nad projektem (proszę wymienić konkretne zadania) |
| Dominik | Płatos | 203739 | Transform, Load, Main, Baza Danych |
| Damian | Dzierżak | 203546 | Extract, ProductTable, Main |
| Aleksandra | Stadnik | 203793 | Dokumentacja, OpiniaTable |

|  |
| --- |
| \_\_/50 pkt |

# Dokumentacja Techniczna

## Użyte technologie

**Maven (ver. 3.3.9)** – narzędzie automatyzujące budowę oprogramowania na platformę Java. Poszczególne funkcje Mavena realizowane są poprzez wtyczki, które są automatycznie pobierane przy ich pierwszym wykorzystaniu. Plik określający sposób budowy aplikacji nosi nazwę POM-u (ang. Project Object Model). Maven wywodzi się z projektu Jakarta. Tak jak i inne produkty fundacji Apache, Maven jest rozprowadzany na licencji Apache License. Szczegółową dokumentację można znaleźć pod adresem: <https://maven.apache.org/>

**Java (jdk 1.8)** – obiektowy język programowania tworzenia programów źródłowych kompilowanych do kodu bajtowego, czyli postaci wykonywanej przez maszynę wirtualną. Język cechuje się silnym typowaniem. Jego podstawowe koncepcje zostały przejęte z języka Smalltalk (maszyna wirtualna, zarządzanie pamięcią) oraz z języka C++ (duża część składni i słów kluczowych). Szczegółowe informacje dostępne są pod adresem <https://www.java.com/pl/>

**Jsoup (ver. 1.10.2)** - Jsoup jest najpopularniejszą biblioteką w Javie przeznaczoną do projektów, w których wymagane są operacje takie jak: pobieranie danych ze stron www; parsowanie źródła strony; wyciąganie ze źródła strony informacji o poszczególnych węzłach. Jeżeli więc w swoim programie chcesz zrobić rzeczy w stylu wyciągnięcie wszystkich nagłówków dostępnych na danej stronie, albo pobranie wszystkich odnośników, to Jsoup sprawdzi się tutaj idealnie. Szczegółowe informacje można znaleźć tu: <https://jsoup.org/>

**IntelliJ (ver. 2016.3.2)** - komercyjne zintegrowane środowisko programistyczne (IDE) dla Javy firmy JetBrains. Wszystkie informacje tu: <https://www.jetbrains.com/idea/>

**Xampp** – darmowy, wieloplatformowy, zintegrowany pakiet, składający się głównie z serwera Apache, bazy danych MySQL i interpreterów dla skryptów napisanych w PHP i Perlu. Nazwa XAMPP jest akronimem od X (ang. cross-platform), Apache, MySQL, PHP, Perl.. Wszystkie informacje dostępne na: https://www.apachefriends.org/pl/

**Git** - rozproszony system kontroli wersji. Szczegółowe informacje: <https://git-scm.com/>

**MySQL** - wolnodostępny system zarządzania relacyjnymi bazami danych. <https://www.mysql.com/>

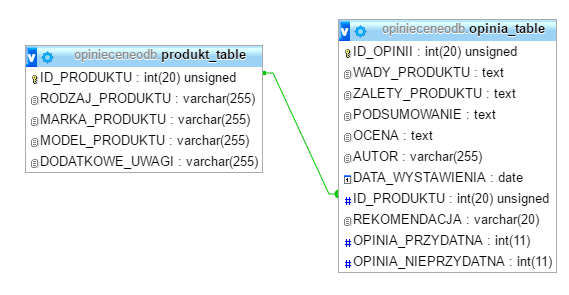
**CSVUtils** – Skorzystaliśmy również z napisanej klasy do eksportu danych do pliku w formacie CSV.

Szczegółowe informacje na temat tej klasy można znaleźć tu: <https://www.mkyong.com/java/how-to-export-data-to-csv-file-java/>

**JDBC** - interfejs programowania, umożliwiający niezależnym od platformy aplikacjom napisanym w języku Java porozumiewać się z bazami danych za pomocą języka SQL. Interfejs ten jest odpowiednikiem standardu ODBC.

## Projekt fizyczny bazy danych

Do stworzenia bazy danych posłużyliśmy się programem XAMPP. Do projektu dołączony jest plik *opinieceneodb.sql* dzięki któremu można wygenerować bazę danych. Projekt fizyczny wygląda następująco:



## Środowisko wymagane do uruchomienia programu

Aby uruchomić program, należy ściągnąć go z publicznego repozytorium:

<https://github.com/dominikQAEngineer/23_Pojekt>

Zainstalować na komputerze program IntelliJ, dostępny pod adresem:

<https://www.jetbrains.com/idea/download/#section=windows>

IntelliJ podczas instalacji zapyta, czy ma przy okazji ściągnąć Javę. Można to również zrobić samemu:

<https://www.java.com/pl/download/win10.jsp>

Dodatkowo należy utworzyć bazę danych, przy pomocy dołączonego do projektu pliku sql. Baza danych powinna mieć następujące parametry:

Nazwa: opinieceneodb

Kodowanie: UTF-8

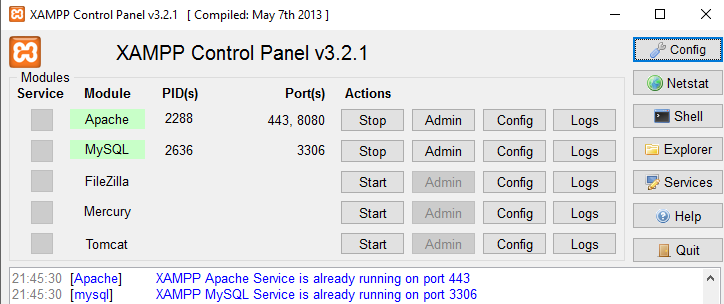
User: root

Hasło:

Można to zrobić za pomocą programu XAMPP:

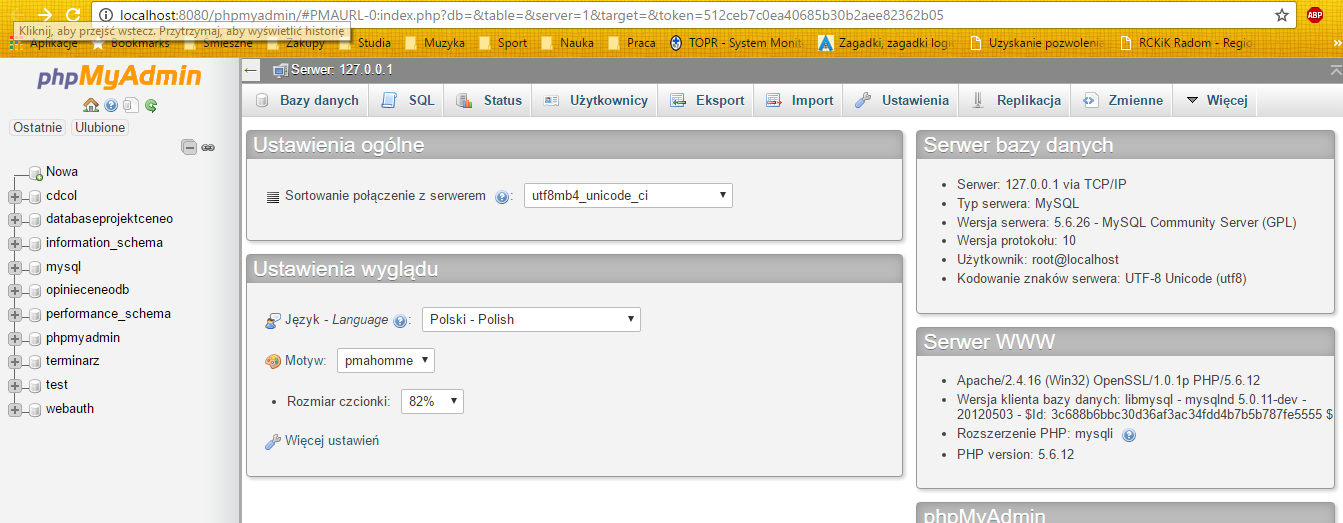
<https://www.apachefriends.org/download.html>

Po uruchomieniu programu otwiera się następujące okno:



Należy kliknąć Start dla modułu Apache oraz MySQL. Port bazy danych powinien wynosić 3306, można to zmienić po kliknięciu przycisku Config. Aby postawić bazę danych wystarczy kliknąć przycisk „Admin” w wierszu modułu MySQL.

Zostaniemy następnie przeniesieni na stronę:



Na tej stronie wystarczy kliknąć Nowa, dalej z górnego menu należy wybrać zakładkę „Import” i załadować plik „opinieceneodb.sql”.

Dzięki temu otrzymujemy w pełni funkcjonalną bazę danych potrzebną do działającego projektu.

## Opis klas

#### Klasa OpiniaTable

**Atrybuty:**

idProduktu – nr identyfikacyjny produktu pobierany ze strony Ceneo (podany przez użytkownika).

idOpinii – numer identyfikacyjny opinii pobrany z portalu ceneo.

zaletyProduktu – zalety produktu jakie wymienił autor opinii.

wadyProduktu – wady produktu jakie wymienił autor opinii.

podsumowanie - treść opinii jaką wystawił autor.

iloscGwiazdek – Liczba gwiazdek na jaką autor opinii ocenił produkt.

autor – autor opinii pobierany z portalu ceneo.

dataWystawienia – data wystawienia opinii.

rekomendacja – przyjmuje jedną z dwóch wartości: POLECAM/ NIE POLECAM.

opiniaPrzydatna – ilość głosów określających opinię jako przydatną.

opiniaNieprzydatna - ilość głosów określających opinię jako nieprzydatną.

**Metody:**

Konstruktor **OpiniaTable**(Element element, String productId);

Argumenty:

element – dana strona pobrana z portalu ceneo

productId – id produktu jaki chce wyszukać użytkownik

Konstruktor jest odpowiedzialny za pobranie z otrzymanej strony interesujących go atrybutów przy pomocy selektorów css(funkcja select z biblioteki Element).

**toString**()

Zwracany typ: String

Metoda nadpisana (override) odpowiedzialna za wyświetlenie obiektu klasy OpiniaTable w postaci ciągu znaków.

Oprócz w/w metod, klasa zawiera również metody get oraz set pozwalające dla danego obiektu pobrać bądź ustawić atrybuty metody. Zostały one wygenerowane automatycznie przy pomocy środowiska IntelliJ

#### Klasa ProduktTable

**Atrybuty:**

idProduktu – nr identyfikacyjny produktu podany przez użytkownika.

rodzajProduktu – rodzaj produktu pobrany z portalu ceneo (kategoria).

markaProduktu – marka produktu, czyli firma.

modelProduktu – nazwa produktu nadana przez producenta, pobrana z portalu.

dodatkoweUwagi – dodatkowe uwagi na temat przedmiotu – zaraz pod nazwą produktu

**Metody:**

Konstruktor **ProductTable**(Element element, String productId);

Argumenty:

element – dana strona pobrana z portalu ceneo

productId – id produktu jaki chce wyszukać użytkownik

Konstruktor jest odpowiedzialny za pobranie z otrzymanej strony interesujących go atrybutów przy pomocy selektorów css(funkcja select z biblioteki Element).

**toString**()

Zwracany typ: String

Metoda nadpisana (override) odpowiedzialna za wyświetlenie obiektu klasy ProduktTable w postaci ciągu znaków.

Oprócz w/w metod, klasa zawiera również metody get oraz set pozwalające dla danego obiektu pobrać bądź ustawić atrybuty metody. Zostały one wygenerowane automatycznie przy pomocy środowiska IntelliJ

#### Klasa Extract

**Atrybuty:**

GLOBALURL – atrybut definiujący adres www portalu, z którego korzystamy w aplikacji (ceneo)

**Metody:**

ArrayList<Element> **extractReviewsFromPage**(Document document)

Argumenty: document – dokument HTML pobrany z portalu Ceneo

Zwracana wartość: Lista elementów(opinii).

Funkcja ta na wejściu otrzymuje dokument html przedstawiający jedną stronę opinii dla danego przedmiotu i wybiera z niego przy pomocy selektora css wszystkie opinie, zwraca je w postaci listy obiektów klasy Element która reprezentuje znacznik HTML

**boolean hasNextPage**(Document document)

Argumenty: document – pobrana strona z portalu ceneo.

Zwracana wartość – true/false

Funkcja sprawdzająca czy na pobranej stronie znajduje sie przycisk przekierowujący na natępną stronę.

String **getUrlToNextPage**(Document document)

Argumenty: document – pobrana strona przy pomocy danego url

Zwracana wartość – String ze ścieżką URL do następnej strony

Funkcja przy pomocy selektora css pobiera atrybut href z kodu źródłowego strony, z elementu który przekierowuje na następną stronę.

Element **extractProduct**(String productId)

Argumenty: Id produktu wybrane przez użytkownika program

Zwracane wartości: znacznik html który zawiera informacje potrzebne do zasilenia klasy ProductTable.

Funkcja pobiera znaczniki html z części strony odpowiedzialnej za opis danego produktu.

ArrayList<Element> **extractReviews**(String productId)

Argumenty: Id produktu wybrane przez użytkownika program.

Zwracane wartości: Lista znaczników HTML odpowiedzialnych za wszystkie opinie dla danego produktu dostępne na portalu ceneo.

Funkcja wykorzystuje wyżej opisane metody. Łączy się w pętli ze wszystkimi stronami opinii dostępnych na portalu ceneo.pl. Połączenie ze stroną następuje przy pomocy biblioteki jsoup oraz funkcji connect. Następnie pobierany jest cały dokument html danej strony z opiniami, używana jest funkcja **extractReviewsFromPage.** Aby sprawdzić czy jest następna strona wykonuje się funkcja **hasNextPage,** jeżeli strona występuje to następuje przekierowanie na następną stronę **getUrlToNextPage,** i znów zebranie wszystkich opinii. Tak aż do momentu kiedy nie będzie następnej strony.

#### Klasa Transform

**Metody**:

ArrayList<OpiniaTable> **transformOpinie**(ArrayList<Element> elements, String productId)

Argumenty:

Elements – znaczniki html przechowujące opinie

productId – nr produktu wybrany przez użytkownika

Zwracana wartość:

Lista elementów klasy OpiniaTable

Metoda transformOpinie zamienia znaczniki HTML na obiekty typu OpiniaTable przy pomocy odpowiednich selektorów css.

ProduktTable **transformProduct**(Element element, String productId)

Argumenty:

Element – znaczniki html przechowujące dane o produkcie

productId – nr produktu wybrany przez użytkownika

Zwracana wartość:

Obiekt klasy ProduktTable.

Funkcja zamienia znaczniki HTML pobrane wcześniej z portalu Ceneo na obiekt reprezentujący dany produkt przy pomocy selektorów css.

OpiniaTable **parseReviewValues**(OpiniaTable review)

Argumenty, Zwracana wartość:

Obiekt klasy OpiniaTable.

Funkcja odpowiedzialna jest za usunięcie z opinii wszelkich polskich znaków oraz apostrofów, które mogą zepsuć import danych do bazy.

ProduktTable **parseProductValues**(ProduktTable product)

Argumenty, Zwracana wartość:

Obiekt klasy ProduktTable.

Funkcja odpowiedzialna jest za usunięcie z produktu wszelkich polskich znaków oraz apostrofów, które mogą zepsuć import danych do bazy.

#### Klasa Load

**Atrybuty**:

url – zmienna tekstowa przechowująca adres bazy danych.

user – zmienna tekstowa przechowująca użytkownika bazy danych.

pass – zmienna tekstowa przechowująca hasło bazy danych.

statement - obiekt klasy Statement przechowujący dane o połączeniu z Bazą.

transform – obiekt klasy Transform służący do odpowiedniego przygotowania danych przed wgraniem do bazy.

**Metody**:

void **wstawProduktDoBazy**(ProduktTable product)

Argumenty: obiekt klasy Produkt.

Funkcja odpowiedzialna jest za wstawienie danego produktu do bazy danych, czyli tabeli ProduktTable. Na początek tworzone jest połączenie z bazą danych przy pomocy klasy Connection oraz sterownika JDBC. Jeżeli połączenie zostało uzyskane, to przy pomocy komendy INSERT IGNORE wstawiany jest do bazy rekord, zawierający wszystkie atrybuty danego argumentu funkcji. Komenda IGNORE zapewnia brak dublowania się rekordów w bazie danych. Na koniec połączenie z bazą jest zamykane

void **wstawOpinieDoBazy**(ArrayList<OpiniaTable> reviews )

Argumenty: Tablica obiektów klasy OpiniaTable.

Funkcja odpowiedzialna jest za wstawienie danych opinii do bazy danych, czyli tabeli OpiniaTable. Na początek tworzone jest połączenie z bazą danych przy pomocy klasy Connection oraz sterownika JDBC. Jeżeli połączenie zostało uzyskane, to przy pomocy komendy INSERT IGNORE wstawiane są do bazy rekordy, zawierające atrybuty danej listy argumentów funkcji. Komenda IGNORE zapewnia brak dublowania się rekordów w bazie danych. Na koniec połączenie z bazą jest zamykane.

void **pokazWszystkieOpinieDlaID**(String productID)

Argumenty: id produktu który wybrał użytkownik.

Podobnie jak dwie poprzednie, funkcja tworzy połączenie z bazą danych, a następnie przy pomocy komendy SELECT wybiera wszystkie opinie dla podanego id produktu, żeby następnie wyświetlić je na konsoli w postaci ciągu znaków. Każdy nowy rekord zaczyna się od nowej linii, atrybuty poszczególnych kolumn zawarte są w nawiasach kwadratowych.

void **zapiszWszystkieOpinieDoPlikuDlaID**(String productID)

Argumenty: id produktu który wybrał użytkownik.

Podobnie jak w powyższej metodzie, funkcja ta pobiera wszystkie opinie dla podanego nr identyfikacyjnego. Następnie zapisuje je do pliku CSV przy pomocy klasy CSVUtils. Do pliku na początku zapisywany jest opis produktu, czyli rekord z tabeli ProduktTable, a następnie wszystkie opinie znalezione dla danego produktu. Nazwa pliku tworzona jest w formacie:

produktNr[productID]WszystkieOpinie.csv

Znaleźć go można w folderze projektu PlikiTekstowe.

void **zapiszJednaOpinieDoPlikuDlaID**(String productID, String opiniaID)

Argumenty: id produktu który wybrał użytkownik, a także wybrany nr opinii.

Funkcja ta działa podobnie jak zapisywanie do pliku wszystkich opinii, tylko że tym razem mamy podane konkretne ID opinii, którą chcemy zapisać do pliku, więc zapisywana jest tylko jedna opinia. Nazwa pliku tworzona jest w formacie:

produktNr[productID]OpiniaNr[opiniaID].csv

Znaleźć go można w folderze projektu PlikiTekstowe.

void **wyczyscOpinie**()

Metoda usuwa wszystkie rekordy z tabeli OpiniaTable przy pomocy polecenia:

TRUNCATE opinia\_table

void **wyczyscOpinieDlaID**(String idProduktu)

Argumenty: id produktu który wybrał użytkownik.

Metoda usuwa wszystkie rekordy z tabeli OpiniaTable dla podanego produktu.

#### Klasa Main

**Metody**

void **wyczyscEkran**()

Funkcja odpowiedzialna za czyszczenie konsoli przy pomocy polecenia:

System.out.print("\f");

int **menu**(String productID)

Funkcja odpowiedzialna za wyświetlanie całego menu użytkownika. Wygląda ono w następujący sposób:

Wybrane ID:

Wybierz, co chcesz zrobic z danym produktem:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1) Pozyskaj dane ze zrodla (E)

2) Przeksztalc dane do postaci modelu danych (T)

3) Zaladuj dane do hurtowni (L)

4) Pelny proces ETL

5) Wyswietl opinie z bazy danych

6) Eksportuj opinie do pliku csv

7) Eksportuj wybrana opinie do pliku csv

8) Usun wszystkie opinie z bazy danych

9) Wybierz inne ID

10) Zakoncz program

Podaj numer z menu:

Funkcja pobiera ID produktu, które na samym początku podaje użytkownik, następnie zwraca nr pozycji z menu, który wybrał użytkownik i przekazuje ją do funkcji main.

void **main**()

Funkcja ta jest odpowiedzialna za działanie programu. Na początku przyjmuje od użytkownika nr produktu, jaki chce on wyszukać na portalu ceneo.pl. Następnie wyświetla wyżej opisane menu i czeka na nr funkcjonalności jaki poda użytkownik.

Wybrany numer jest przetwarzany przez funkcję sterującą switch.

Dalej wykorzystane są metody oraz klasy opisane wcześniej, odpowiedzialne za proces ETL jak i zapisywanie danych do pliku. Za każdym razem sprawdzane jest, czy nr jaki podał użytkownik występuje na portalu ceneo oraz walidowana jest poprawność podanego znaku (czy przypadkiem użytkownik nie podał litery zamiast cyfry).

# Dokumentacja użytkownika

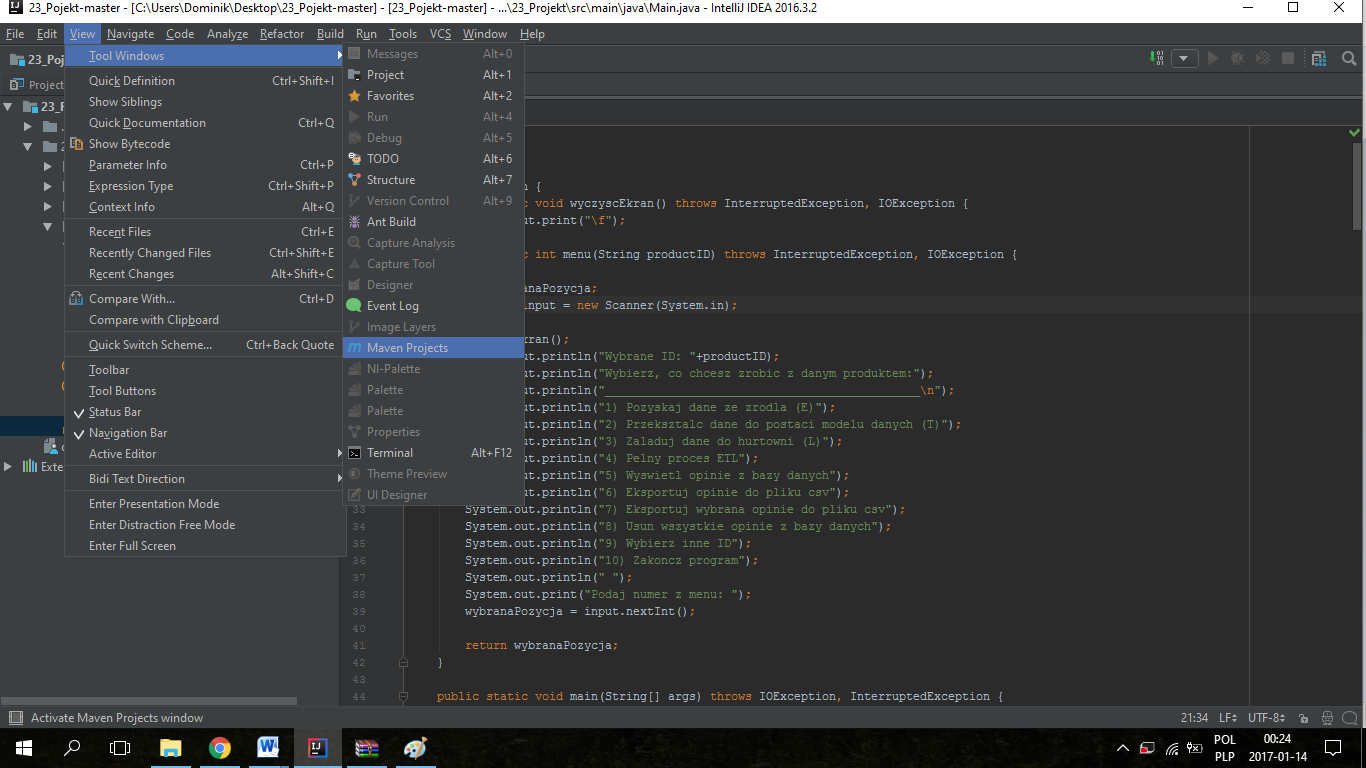
## Instalacja

Jeżeli wszystko z rozdziału „Środowisko wymagane do uruchomienia programu” Zostało zainstalowane to możemy przejść do uruchomienia naszego programu:

Uruchamiamy w tym celu program IntelliJ

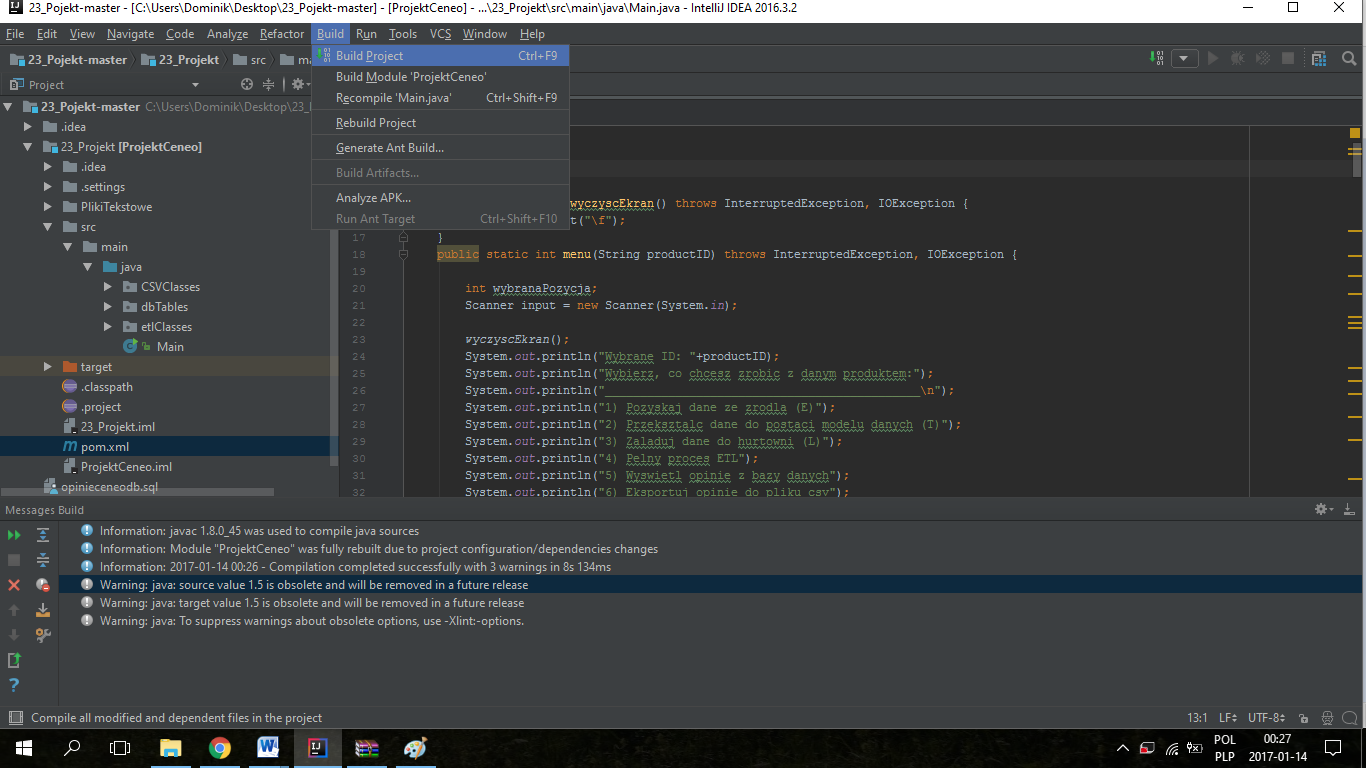
Otwieramy projekt, który pobraliśmy z repozytorium

Otwieramy panel sterowania do Maven’a jak na poniższym zrzucie ekranu:

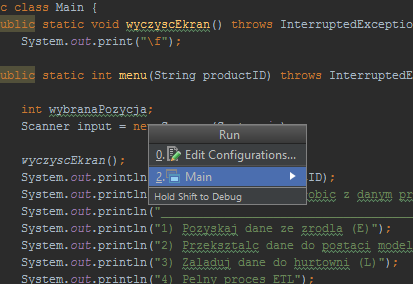


Wybieramy plik POM – w narzędziu do obsługi Maven’a klikamy na „+” a następnie szukamy w naszym projekcie pliku pom.xml.

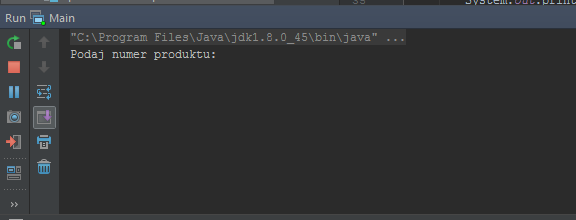
Budujemy program



Następnie możemy kliknąć Run->Run i wybrać klasę Main jako plik do uruchomienia:



Na konsoli pojawi się następująca komenda:



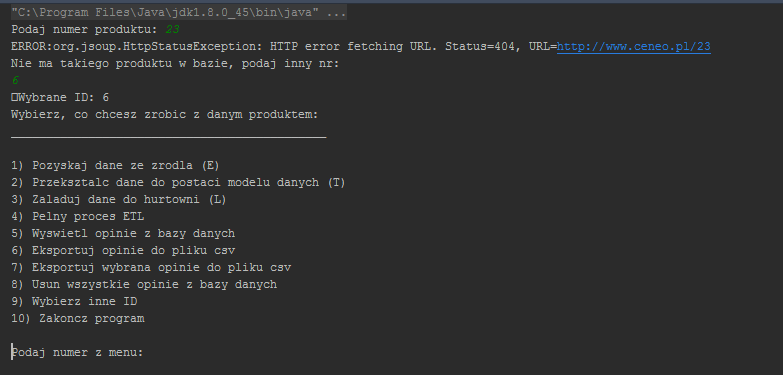
Oznacza to, że można zacząć korzystać z oprogramowania co kończy proces instalacji.

## Instrukcja obsługi

Od razu po uruchomieniu programu pojawia się komunikat:

„Podaj numer produktu:”

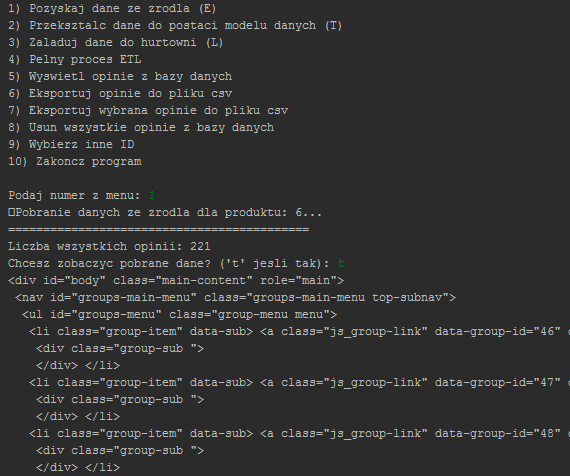
W tym momencie należy podać nr ID produktu, dla którego chcemy wyszukać wszystkie interesujące nas informacje. Program jest odporny na wszelkie pomyłki ze strony użytkownika, co oznacza że będzie prosił o ponowne podanie numeru produktu jeżeli nie ma takiego na portalu internetowym lub podany numer ma niepoprawny format.



Powyższy screen pokazuje, że po wpisaniu odpowiedniego numeru produktu pojawia się Menu, dzięki któremu możemy sterować dalszym działaniem programu.

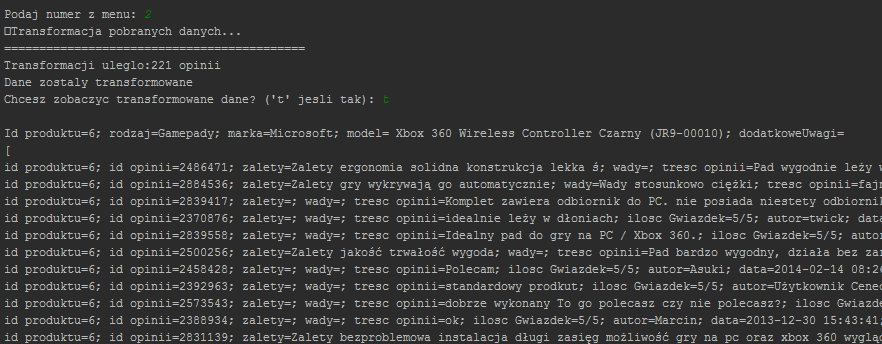
1. Pozyskaj dane ze zrodla (E)

Funkcja odpowiedzialna jest za pobranie znaczników html z portalu ceneo przy pomocy klasy Extract, pobrane dane wyglądają następująco:



1. Przeksztalc dane do postaci modelu danych (T)

Dane pobrane przy pomocy funkcji Extract posiadają bardzo dużo niepotrzebnych informacji, dlatego niezbędna jest ich transformacja:



Po transformowaniu dane są gotowe to transportu do bazy danych.

1. Zaladuj dane do hurtowni (L)

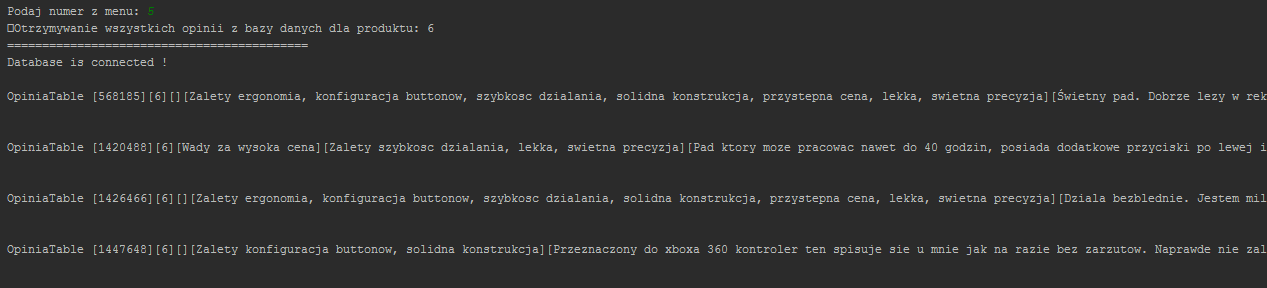
Jeżeli mamy pobrane dane, oraz są już przetransformowane, to możemy je załadować do bazy danych wywołując tą opcję z menu.

1. Pelny proces ETL

Jeżeli nie interesują nas pojedyncze elementy procesu ETL, możemy go wykonać od razu w całości przy pomocy tej komendy pozycji z menu

1. Wyswietl opinie z bazy danych

Ta opcja wyświetla wszystkie opinie z bazy danych dla podanego ID produktu.



1. Eksportuj opinie do pliku csv
2. Eksportuj wybrana opinie do pliku csv
3. Usun wszystkie opinie z bazy danych
4. Wybierz inne ID
5. Zakoncz program

Pozostałe pozycje MENU nie wymagają omówienia. Program zmusza użytkownika do wykonania procesu ETL po kolei, co oznacza że nie można wykonać funkcji Load bez wykonania procesu E oraz T, tak samo jak nie można transformować danych, jeżeli nie wywołaliśmy wcześniej funkcji Extract.