

Praktyka programowania 2012/2013

Instrukcja projektowa cz.0

Wprowadzenie

Prowadzący: Tomasz Goluch

Wersja: 1.2

I. Warunki zaliczenia

Cel: Zapoznanie studentów z warunkami zaliczenia części projektowej przedmiotu.

Najważniejsze informacje dotyczące zaliczenia części projektowej przedmiotu:

- Niniejsza instrukcja jest dostępna pod adresem: http://kaims.pl/~goluch/lab/InstrProjPPcz0_2012-13 – proszę sprawdzić czy nie pojawiła się **zaktualizowana** wersja.
- Obecność na zajęciach projektowych jest **obowiązkowa**.
- Pierwsze zajęcia mają charakter **wprowadzający** (1 godz./grupa).
- Odbędzie się **osiem** zajęć projektowych (**jedno** wprowadzające, **trzy** niepunktowane i **cztery** punktowane), każde (oprócz wprowadzających) po **dwie** godziny zajęciowe.
- Godzina zajęciowa trwa **45 min**.
- Do zrealizowania są dwa projekty.
- Za każdy projekt można uzyskać maksymalnie **15** punktów.
- W sumie można uzyskać **2 x 15 = 30 pkt.** co stanowi **100%** części oceny laboratoryjnej.
- Próg zaliczenia laboratorium to **50 % (15 pkt.)**.
- Do zaliczenia całego przedmiotu należy posiadać ponad **50%** sumy punktów z wszystkich trzech części (wykład, laboratorium, projekt).
- Wyniki projektu **grup: 05, 08 i 10** – (**termin wspólny**) dostępne są na stronie domowej: www.kaims.pl/~goluch w zakładce: Dla studentów → Semestr letni 2012/2013 → Praktyka programowania → projekt → Wyniki projektu grupy: [właściwa podgrupa](#).

II. Kryteria oceny

Cel: Zapoznanie studentów z najważniejszymi elementami mającymi wpływ na ocenę.

Informacje dotyczące oceny z części laboratoryjnej przedmiotu:

- Laboratorium ma na celu zweryfikowanie wiedzy pozyskanej na wykładzie co oznacza, że jest to **praca indywidualna**, aczkolwiek w przypadku wątpliwości można zadawać pytania prowadzącemu.
- Podstawą zaliczenia projektu jest **umieszczenie** kodu na platformie STOS oraz osobista **prezentacja** kodu i działającego programu w godzinach zajęć projektowych.
- Pierwszy projekt będzie prezentowany w systemie **Windows** a drugi w systemie **Linux**.
- Na laboratorium dostępne są następujące środowiska programistyczne (**Windows**):
 - Microsoft Visual Studio 2008 i 2010¹,
 - Dev-C++ 4,
 - Code::Blocks,
 - Ideone – kompilator online i narzędzie do debugowania dostępne pod adresem: www.ideone.com.
- (**Linux**):
 - KDevelop.

¹ Każdy student może otrzymać środowisko programistyczne Microsoft Visual Studio za pomocą programu Microsoft DreamSpark® (dawniej MSDNAA). Więcej szczegółów dostępnych jest na stronie programu: <http://www.eti.pg.gda.pl/pracownicy/oprogramowanie/Aby.htm>

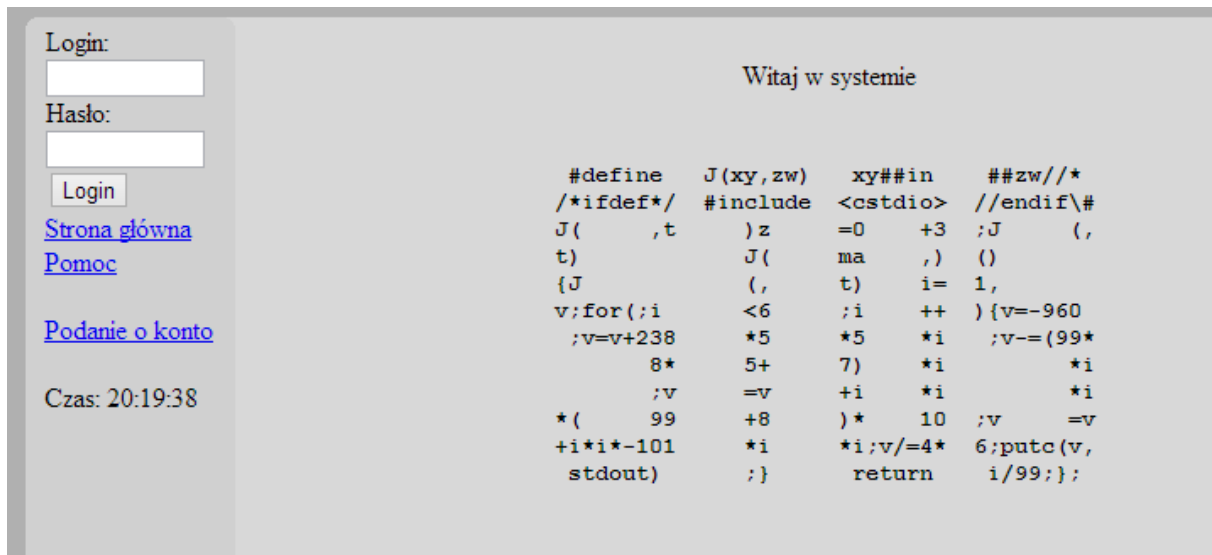
- Na ocenę końcową mają wpływ następujące czynniki:
 - **terminowość** – po terminie będą przyjmowane tylko projekty z ważnym zwolnieniem lekarskim.
 - **kompletność** – występowanie wszystkich funkcjonalności opisanych w zadaniu.
 - **poprawność** – zgodność funkcjonalności z opisem ich działania.
 - **czytelność kodu** – kod powinien być zrozumiały nie tylko dla autora ale i dla prowadzącego projekt.
 - **brak powtórzeń kodu** – stosowanie pętli oraz funkcji.
 - **stosowanie przejrzystego stylu programowania** – formatowanie kodu, odstępy, konwencja nazewnictwa. Nie wymaga się konkretnego stylu jednak ważne aby był on jednolity i najlepiej zgodny z któryś ze standardów (przykładowo środowisko programistyczne Microsoft Visual Studio zapewnia automatyczne formatowanie kodu).
 - **stosowanie komentarzy** – ale tylko w uzasadnionych przypadkach, najlepiej gdyby kod ich nie wymagał (można to osiągnąć wykorzystując odpowiednie nazwy zmiennych i stałych, nazwy funkcji, typy numeryczne, dyrektywy preprocesora – `#define` oraz słowa kluczowe – `typedef`).
 - **samodzielność** – **UWAGA !!!** – każdy z dwóch projektów będzie poddany weryfikacji **antyplagiatowej**. W przypadku wykrycia plagiatu w pierwszym projekcie autor otrzymuje (**-5 pkt.**). W przypadku wykrycia plagiatu w drugim projekcie osoba nie zalicza części projektowej co automatycznie powoduje niezaliczenie przedmiotu.
 - **Dodatkowa funkcjonalność** – do każdego projektu prowadzący zamieści na stronie **bonusowo** punktowane funkcjonalności (dopuszcza się implementację własnych propozycji po uzgodnieniu ich z prowadzącym).

III. Platforma internetowa STOS

Cel: Zapoznanie studentów z procedurą zakładania konta oraz obsługi platformy STOS.

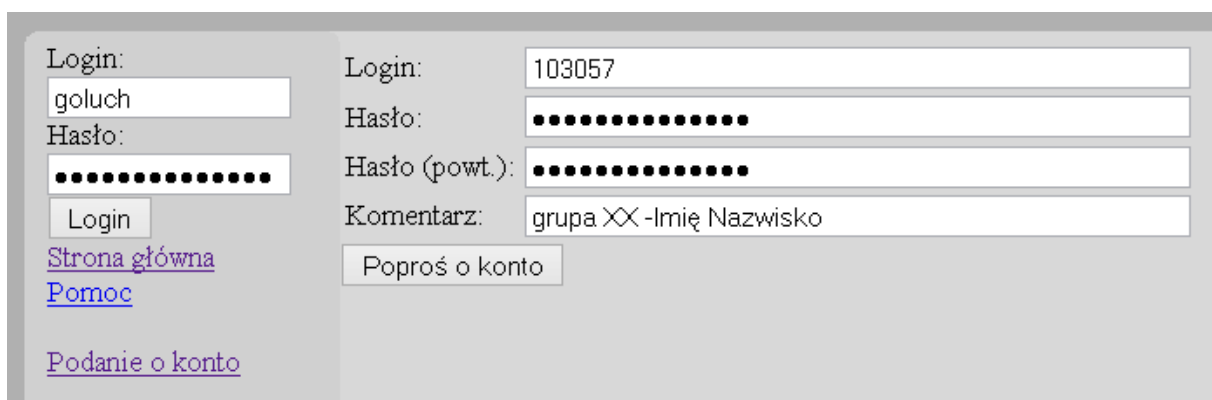
Zasady zakładania konta i obsługi internetowej platformy STOS:

- Platforma jest dostępna pod adresem: <https://kaims.pl/~kmocet/stos>
- Strona startowa została przedstawiona na rys.1, w pierwszym kroku należy wybrać opcję → Podanie o konto.



Rysunek 1 - Strona startowa platformy STOS.

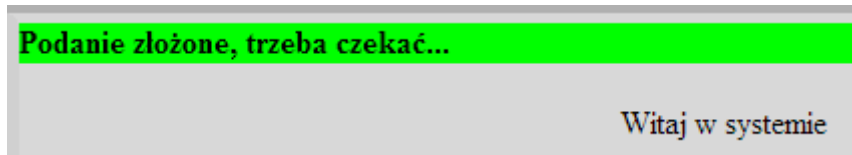
- Okno rejestracji nowego użytkownika przedstawiono na rys. 2



Rysunek 2 - Okno rejestracji nowego użytkownika.

- **UWAGA !!!** – jako nawę użytkownika proszę wpisać **TYLKO NUMER INDEKSU**, inne nazwy będą ignorowane i usuwane!

- Jako **Komentarz** proszę podać frazę: **grupa XX – Imię Nazwisko** gdzie:
 - **XX** – numer grupy projektowej,
 - **Imię** – imię studenta,
 - **Nazwisko** – nazwisko studenta.
- Po poprawnym złożeniu zgłoszenia pojawi się adekwatny komunikat (rys. 3) po którego wyświetleniu należy czekać na akceptację konta przez prowadzącego.



Rysunek 3 – Komunikat poprawnego złożenia podania o założenie konta.

- Po zaakceptowaniu nowego użytkownika przez prowadzącego można już się zalogować do platformy.
- Po zalogowaniu pojawi się lista aktywnych etapów zadań projektowych – rys. 4.

Lp.	Zadanie	Zgłaszanie	Wynik	Punkty	Termin
1	Zadanie pierwsze etap 1	Zgłoszenie		0	
2	Zadanie pierwsze etap 2	Zgłoszenie		0	

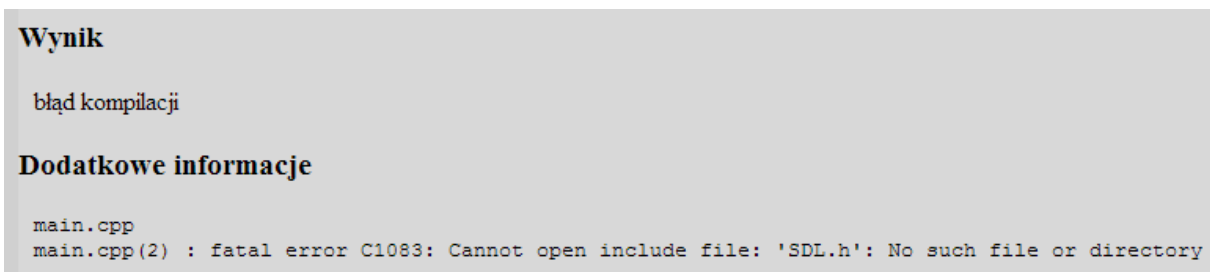
Rysunek 4 - Okno z zawartością listy aktywnych etapów zadań projektowych.

- W celu zapoznania się z treścią zadania należy wybrać odpowiedni wiersz z kolumny **Zadanie**.
- W celu zgłoszenia zadania należy wybrać odpowiedni wiersz z kolumny **Zgłaszanie** co spowoduje wyświetlenie się okna przedstawionego na rys. 5.



Rysunek 5 - Okno pozwalające na zgłoszenie zadania.

- W celu automatycznej weryfikacji kodu proszę wgrać pliki na serwer wykorzystując opcję → **Choose File** (możliwość przesyłania wielu plików projektu) i wybrać opcję → **Wyślij**.
- W przypadku braku błędów powinna pojawić informacja o poprawnej kompilacji.
- W przeciwnym przypadku powinien pojawić się opis błędu (przykładowy komunikat o błędzie wraz z dodatkowymi informacjami – rys. 6).



• Rysunek 6 - Okno pozwalające na zgłoszenie zadania.

- Zadania projektowe nie podlegają automatycznej weryfikacji i są ręcznie sprawdzane przez prowadzącego. Należy zatem skompilować i zaprezentować prowadzącemu działanie umieszczonego wcześniej na platformie SPOX kodu.
- Po zakończeniu zajęć proszę **WYŁĄCZYĆ KOMPUTER !!!**