

## TKOM 2020 semestr zimowy - realizacja projektów

1. Do zdobycia jest **max. 30 pkt.** Aby być **dopuszczonym do egzaminu** trzeba uzyskać **co najmniej 15 pkt.** z projektu. Punktacja składa się z następujących elementów:
  - **A – zakres i jakość projektu oraz zrealizowanej funkcjonalności – maksymalnie 10 pkt.,**
  - **B – jakość implementacji - maksymalnie tyle punktów, ile uzyskano za A,**
  - **C – dokumentacja końcowa projektu – maksymalnie 5 pkt. - warunkiem oceny dokumentacji jest uzyskania co najmniej 5 pkt. z A,**
  - **D – punkty za poszczególne etapy realizacji (5 pkt.) – warunkiem ich uzyskania jest nadesłanie w terminie (wg harmonogramu) satysfakcjonujących rezultatów danego etapu,**
  - **E – punkty karne – naliczane za opóźnienia w nadsyłaniu wyników etapów II do V (zgodnie z harmonogramem i punktem 3 i 4 regulaminu),**

**Punkty A÷C przyznawane są za wyniki etapu V (zgodnie z harmonogramem).**

**Punkty karne odliczane będą od ogólnej liczby punktów uzyskanych za projekt, tzn. ocena = A+B+C+D-E**

2. **Harmonogram.** Wszystkie osoby realizujące projekty powinny:
  - I. **Do 10.11.2020 uzgodnić z Prowadzącym ostateczny zakres tematu projektowego.**
  - II. **Do 23.11.2020 (8 rano) nadesłać projekt wstępny – 1 punkt** - (zachęcam jednak do wcześniejszych konsultacji z prowadzącymi zwłaszcza w zakresie funkcjonalności jak i gramatyki):
    - opis zakładanej funkcjonalności i **przykłady obrazujące dopuszczalne konstrukcje językowe oraz ich semantykę.**
    - **formalna specyfikacja i składnia: realizowanego języka (jeśli chodzi np. o interpretery), wszystkich plików/strumieni wejściowych, danych konfiguracyjnych itp.**
    - zwięzłą analizę wymagań funkcjonalnych i niefunkcjonalnych, obsługa błędów.
    - **sposób uruchomienia, wej./wyj.**
    - zwięzły opis sposobu realizacji: modułów/obiektów i interfejsów pomiędzy nimi wraz z opisem co i jak będzie w nich realizowane (m.in. jakie tokeny będą rozróżniane, jak będzie realizowane przetwarzanie w poszczególnych komponentach), wykorzystywane struktury danych (np. do jakich struktur danych trafią określone obiekty - może być opis formalny lub tekstowy), zobrazowane wykorzystanie struktur danych, formy pośrednie itp.
    - zwięzły opis sposobu testowania.
  - III. **Do 7.12.2020 (8 rano) nadesłać:**
    - 2 punkty - implementację analizatora leksykalnego (lub ew. pliki wej. dla wykorzystywanych narzędzi generacyjnych)+przykłady testowe.
  - IV. **Do 11.01.2021 (8 rano) nadesłać:**
    - 2 punkty - implementację analizatora składniowego i generację hierarchii obiektów koniecznych do interpretacji oraz testy; zachęcamy do skonsultowania szkieletu parsera do 23.12.2020.
  - V. **Do 25.01.2021 (8 rano) nadesłać:**
    - całość projektu, tzn. komplet implementacji oraz dokumentacja projektowa (zwięzły opis implementacji i testowania oraz opis użytkowy).
3. **Opóźnienie nadesłania wyników etapów II÷IV lub uznanie przez Prowadzącego rezultatu prac tych etapów za rażąco niesatysfakcjonujące skutkuje punktami karnymi w wysokości 1/dzień roboczy. Istotna jest tu data nadesłania a nie terminu osobistego spotkania.** Zachęcamy do konsultowania się przed wyznaczonymi terminami.

4. Analogicznie do punktu 3, **opóźnienie nadeśłania akceptowalnych dla Prowadzącego wyników etapu V skutkuje naliczeniem punktów karnych w wysokości 2/dzień roboczy. Po 25 stycznia przewidywana jest tylko jednokrotna możliwość wprowadzania poprawek (powiązana z naliczaniem kar za opóźnienie).** Termin definitywnego końca projektu – w tym procesie oceniania - zbiega się z ostatnim dniem semestru (**29 stycznia**).
5. **Warunkiem odbioru końcowej wersji rozwiązania jest omówienie i zatwierdzenie przez Prowadzącego poszczególnych etapów projektu.**
6. **PROWADZĄCY OCENIA PROJEKT JEDYNI W OBECNOŚCI AUTORA.**
7. Wszyscy kończą projekt **25 stycznia (do godziny 8 rano)** - do tego czasu musi być dostarczony Prowadzącemu kod źródłowy rozwiązania, zbiór przypadków testowych oraz dokumentacja końcowa (**wszystko w postaci elektronicznej**) - osoby przedstawiające swoje rozwiązania **przed tym terminem** mogą liczyć na **możliwość poprawy projektu bez utraty punktów**.
8. Zachęcam do przestrzegania zasad implementacji wynikających z inżynierii oprogramowania. Polecam wzorowanie się na przykładach jakie zamieszczone są w materiałach do wykładu na stronach "elki". W szczególności dotyczy to wyraźnego podziału na moduły/obiekty realizujące poszczególne funkcje (np. czytanie z plików wejściowych, analiza leksykalna, analiza składniowa, semantyczna itd.).
9. Języki programowania: **C++, Java, Kotlin, C#, Python** (z ograniczeniem: środowisko docelowe czy też język nie może zastąpić tego, czego Państwo powinniście się nauczyć w ramach przedmiotu TKOM). Istnieje za to możliwość wykorzystywania narzędzi typu **Lex, Yacc, Flex, Bison, ANTLR** czy też biblioteki **boost** (ich nauka sama w sobie jest wyzwaniem i bywa przydatna więc tego typu rozwiązań blokować nie będziemy). Wymagane są jednak wcześniej konsultacje z Prowadzącym, jakie elementy bibliotek/języków chcecie wykorzystać. Środowisko systemowe (Windows/Linux/Android...) nie ma znaczenia. Pozostałe języki programowania również są możliwe, ale **po konsultacji z Prowadzącym**.

**Terminy konsultacji (poszczególni prowadzący mogą przewidzieć dodatkowe terminy spotkań dedykowanych projektowi TKOM) i kontakty:**

**Piotr Gawkowski**

konsultacje: wtorki 18-20, pok. 313a, [P.Gawkowski@ii.pw.edu.pl](mailto:P.Gawkowski@ii.pw.edu.pl), tel. 22 234 1393

**Konrad Grochowski**

konsultacje: piątki 14-16, pok. 313a, [K.Grochowski@ii.pw.edu.pl](mailto:K.Grochowski@ii.pw.edu.pl), tel. 22 234 1393

**Witold Wysota**

pok. 313a, [W.Wysota@ii.pw.edu.pl](mailto:W.Wysota@ii.pw.edu.pl), tel. 22 234 1393

Generalnie, i tak wszystko poprzez MS Teams....