



Zadanie projektowe z języka Haskell

Semestr: zima 2012

Opis zadania

Zadanie realizowane jest w **dwuosobowych** zespołach. Skład zespołu należy podać przy oddawaniu rozwiązania (patrz niżej). W ramach zadania należy napisać prostą aplikację do **układania tygodniowego planu zajęć** na uczelni. Każde zajęcia mają określony przedmiot, przedział czasowy, grupę studencką i salę w której się odbywają. Przedmiot ma określony czas zajęć w tygodniu.

Operacje oferowane przez taki program powinny obejmować:

- utworzenie/zmiana/przeglądanie przedmiotów;
- utworzenie/zmiana/przeglądanie grup studenckich;
- utworzenie/zmiana/przeglądanie sal;
- układanie/przeglądanie planu zajęć.

Układanie powinno być zarówno manualne jak i automatyczne.

W przypadku manualnego układania aplikacja ma uniemożliwiać złamanie ograniczeń planu np.: dwa zajęcia nie mogą jednocześnie odbywać się w jednej sali, dwa przedmioty nie mogą odbywać się jednocześnie, jedna grupa nie może mieć więcej niż x godzin dziennie, zajęcia odbywają się od godziny y do godziny z). W przypadku automatycznego układania planu można zastosować najprostszy algorytmu przeglądania wszystkich możliwych kombinacji.¹

Interfejs aplikacji można zrealizować w trybie tekstowym na zasadzie: wyświetlamy dostępne opcje, użytkownik wybiera opcję, użytkownik wpisuje dane/aplikacja prezentuje dane, ponownie wyświetlamy dostępne opcje. Szczegółowa organizacja dostępnych opcji, prezentowanych danych i sposobu interakcji z użytkownikiem jest częścią zadania i podlega ocenie. Aplikacja ma być w pełni **użyteczna** (choć z uproszczonym interfejsem) i odporna na błędy we wprowadzanych danych (np. podanie liter zamiast cyfr lub wartości spoza zadanego zakresu). Wprowadzane dane muszą być przechowywane w plikach, tak żeby można było powrócić do poprzedniego stanu po ponownym uruchomieniu aplikacji.

Termin i forma oddania zadania

Źródła programu należy wysłać w formie załącznika na adres a.ratkowski@elka.pw.edu.pl do dnia **10 stycznia 2013 r.** Mail powinien być zatytułowany „[SPOP Haskell] *Nazwisko1 Nazwisko2*”. Dodatkowo należy również dołączyć dokument (*.pdf*, *.rtf* lub *.doc*) zawierający jedną (i tylko jedną) stronę opisu do źródeł. Opis ten może mieć charakter półformalny (odpowiednik osobistego prezentowania źródeł prowadzącemu). Należy w nim założyć, że czytelnik wie do czego program służy, natomiast chce zorientować się, co gdzie jest w kodzie i dlaczego zrobione zostało tak, a nie inaczej (jeśli zastosowano niekoniecznie oczywiste rozwiązania). Poziom szczegółowości opisu wyznacza jego maksymalny rozmiar równy jednej stronie.

Kryteria oceny

Projekt jest oceniany w skali 0-20 punktów. Przy ocenie będą brane pod uwagę:

- zakres funkcjonalny programu (czy program udostępnia wymienione w opisie zadania funkcje?),
- zachowanie programu (czy jest zgodne z oczekiwaniami?),
- czytelność i intuicyjność interfejsu,
- zwięzłość i czytelność kodu, w tym jakość komentarzy,
- przydatność (w trakcie dokonywania przeglądu kodu) dołączonej „dokumentacji”.

ⁱ Ambitnych zachęcam do zastosowania bardziej wyszukanego algorytmu.