

Regulamin przedmiotu: Języki Programowania

Zajęcia trwają 15 tygodni (2 godziny wykładu, 2 godziny laboratorium tygodniowo)

Zaliczenie zajęć jest uwarunkowane zaliczeniem zajęć laboratoryjnych

Prowadzący zajęcia laboratoryjne:

mgr inż. Łukasz Graczykowski

mgr inż. Małgorzata Janik

mgr inż. Tomasz Pietrzak

mgr inż. Maciej Szymański

mgr inż. Barbara Trzeciak

dr inż. Hanna Zbroszczyk

Organizacja zajęć laboratoryjnych:

- przewidzianych jest 15 zajęć laboratoryjnych (w tym 12 punktowanych, 2 kolokwia, 1 dodatkowe)
- obecność jest obowiązkowa (możliwe są maksymalnie 2 nieobecności);
- spóźnienie na zajęcia powyżej 15 minut automatycznie jest odnotowane jako nieobecność;

Zasady oceniania na zajęciach punktowanych:

- zajęcia punktowane obejmują wykonanie 12 (dwunastu) zadań o zróżnicowanym stopniu trudności (pierwsze zajęcia są także punktowane)
- dopuszczenie do wykonania zadania może być uwarunkowane zaliczeniem kolokwium wstępnego
- w trakcie pisania programu wolno korzystać z napisanych przez siebie programów oraz zasobów Internetu*
- napisany w trakcie trwania laboratorium program należy oddać na tych samych zajęciach
- za każde zadanie można otrzymać **0-5 pkt** (zrozumienie zadania: 1pkt, wykorzystanie formalnych środków języka C++: 3 pkt, aspekty użytkowe oraz strona estetyczna: 1 pkt)
- w przypadku nie skończenia programu na zajęciach oceniony zostanie napisany, skompilowany oraz działający jego fragment (w przypadku programu, który nie kompiluje, ani nie wykonuje się poprawnie możliwe jest zdobycie maksymalnie 2 pkt), **program należy skończyć we własnym zakresie (nie dokończenie programu może skutkować niedopuszczeniem do kolejnych zajęć)**;
- w nadzwyczajnych sytuacjach (**niezależnych od studenta**) prowadzący może wyrazić zgodę na skończenie programu w domu, ustalając nową (niższą) ocenę w zależności od stopnia zaawansowania pracy wykonanej na zajęciach
- w przypadku nieobecności studenci są zobowiązani do zrealizowania materiału we własnym zakresie i przedstawienia rozwiązania **najdalej 2 tygodnie** po nieobecności (na zajęciach lub konsultacjach) - w przypadku usprawiedliwionej nieobecności możliwe jest zaliczenie zaległego programu na mniejszą (4 pkt) ilość punktów; **nie nadrobienie zaległości (zarówno w przypadku nieobecności usprawiedliwionej i nieusprawiedliwionej) może skutkować niedopuszczeniem do kolejnych zajęć**
- w przypadku nieobecności nieusprawiedliwionej liczba zdobytych punktów wynosi **0 (zero)**

*) nie wolno korzystać z programów pocztowych (chyba, że prowadzący wyrazi zgodę), komunikatorów internetowych, ani z programów kolegów z grupy swojej, jak i żadnej innej; korzystanie z telefonów komórkowych jest także zabronione.

Zasady oceniania kolokwiów:

- w trakcie semestru będą 2 (dwa) kolokwia: jedno w połowie semestru, drugie na końcu;
- kolokwium będzie polegało na napisaniu 1 (jednego) programu z materiału zrealizowanego na zajęciach (możliwe jest jednak korzystanie z:
 - własnych programów z zajęć,
 - materiałów wykładu dostępnych w trybie “offline” lub w wersji papierowej,
 - podręczników do programowania w C oraz C++,
 - własnych notatek);
- napisany program należy przesłać przed końcem trwania kolokwium na adres e-mailowy prowadzącego;
- program będzie oceniany w skali 0-20 pkt (oceniane będą:
 - zakres merytoryczny zrealizowanego zadania,
 - wykorzystane środki formalne języka C++,
 - aspekty użytkowe interfejsu,
 - strona estetyczna);

Jest możliwość poprawy kolokwium na ostatnich zajęciach (w grupie swojej lub innej), przy pierwszej poprawie kolokwium możliwe będzie zdobycie maksymalnie 15 pkt, przy drugiej poprawie – 10 pkt.

Na ocenę końcowa przedmiotu wpływają:

wyniki z kolokwium z laboratorium: $2 * 20 \text{ pkt} = 40 \text{ pkt}$;

wyniki z programów napisanych na zajęciach $12 * 5 \text{ pkt} = 60 \text{ pkt}$.

Ocena końcowa wystawiana jest na podstawie procentowego udziału sumy Uzyskanych punktów do sumy punktów możliwej do uzyskania (100 pkt) wg. następującej zależności:

(50%, 60%> - 3.0

(60%, 70%> - 3.5

(70%, 80%> - 4.0

(80%, 90%> - 4.5

(90%, 100%> - 5.0

Na ostatnich (15) zajęciach osoby, którym brakuje do oceny o 0.5 (pół) stopnia wyższej nie więcej niż 5 pkt mogą poprawić się poprzez napisanie dodatkowego programu.

Zaliczenie eksternistyczne

Dla osób programujących w C++ możliwe jest zaliczenie przedmiotu projektem eksternistycznym. Osoby chcące zaliczyć przedmiot w tej formie powinny zgłosić się do prowadzącego najdalej na **drugich** zajęciach laboratoryjnych, na trzecich zajęciach napiszą kolokwium kwalifikujące do pracy w tym trybie.

Wymagania do projektów eksternistycznych:

- nietrywialny problem, do którego rozwiązania najlepiej nadaje się podejście obiektowe,
- dokładna specyfikacja projektu,
- stworzony projekt z dokumentacją w kodzie źródłowym,
- dokumentacja użytkownika.

I) **Zaliczenie projektu eksternistycznego** polega na zaliczeniu 3 (trzech) etapów kontrolnych w terminach zajęć podanych w nawiasach: *beta* (5), *release candidate* (10), *final* (15).

II) Po etapie *beta* prowadzący może projekt zdyskwalifikować, dlatego do tego czasu zalecane jest uczestniczenie w zajęciach programowych.

III) Po etapie *release candidate*, w przypadku braku możliwości skończenia projektu zawierającego wszystkie elementy języka omawiane na wykładzie prowadzący może projekt zamknąć. Od tego momentu należy uczestniczyć w zajęciach.

IV) Przy ustalaniu oceny ostatecznej brane pod uwagę są oceny z etapów pośrednich.

Literatura

- 1) B. Stroustrup – Język C++ (The C++ Programming Language), WNT 2002
- 2) J. Grębosz – Symfonia C++ standard, Pasja C++, Edition 2005
- 3) B. Eckel - Thinking in C++. Edycja polska, Helion 2002
- 4) S.B. Lippman – Podstawy języka C++ (C++ Primer), WNT 1997
- 5) J. Liberty – Poznaj C++ w 10 minut, Intersoftland 1999
- 6) Nicolai M. Josuttis - C++ Biblioteka standardowa. Podręcznik programisty, Helion 2003

Program przedmiotu:

- 1) Wprowadzenie (zasady zaliczenia przedmiotu), literatura. Język C, a C++. Typy referencyjne.
- 2) Przeładowanie nazw funkcji, wprowadzenie do klas.
- 3) Konstruktory, destruktory, funkcje zaprzyjaźnione.
- 4) Przeładowanie operatorów.
- 5) Dziedziczenie.
- 6) Funkcje wirtualne.
- 7) Operacje wejścia / wyjścia. Operacje na plikach.
- 8) Szablony funkcji.
- 9) Szablony klas.
- 10) Elementy biblioteki STL
- 11) Obsługa sytuacji wyjątkowych.
- 12) Algorytmy, struktury danych I.
- 13) Algorytmy, struktury danych II.
- 14) Konwertery oraz konwersje.
- 15) Wystawienie ocen.