

Informatyka II

EE-DI, WEiI PRz

Instrukcja ta jest dostępna w formie źródłowej pod adresem http://git.dms-serwis.com.pl/henrietta/inf2_eedi.

Pod tym adresem będą pojawiać się również instrukcje do kolejnych laboratoriów.

Laboratorium 2

“Jedyne, co musisz wiedzieć, to gdzie jest biblioteka” - Albert Einstein

Z tego laboratorium przygotowujesz sprawozdanie. Przygotowujesz je na zajęciach, a przy ich zakończeniu wysyłasz na adres podany na końcu tej instrukcji. Instrukcja dla wszystkich jest taka sama.

Rzeczy oznaczone tak, jak poniżej, dotyczą tego, co masz zawrzeć w sprawozdaniu. Na przykład:

Zapisz swoje imię, nazwisko, adres e-mail, kierunek i rok studiów oraz grupę laboratoryjną i numer albumu.

Podaj również numer zajęć laboratoryjnych (nr 2).

Mogą być to też pytania, na które w sprawozdaniu udzielisz odpowiedzi. Możesz pomagać sobie wyszukiwarką internetową, oraz zabrać głos w dyskusji, jeśli się jakaś wywiąże.

Do pewnego miejsca tą instrukcję **należy wykonać**. Niewykonanie jej będzie skutkowało obniżeniem oceny.

Inżynier musi umieć uczyć się nowych rzeczy szybko i pod wpływem stresu z zewnętrznych źródeł. Czasem wymaga to załadowania do głowy podręcznika w rekordowym tempie - niestety, *nikogo to już nie dziwi*.

Ciekawe zasoby do nauki C++:

- Kurs C++0x
- Kurs C++ - Mirosław Zelent
- Ogólna lista kursów

Praca w grupach

‘Sukces ma wielu ojców, porażka jest sierotą’

Tą instrukcję należy wykonać w grupie. Dobierz się z innymi w grupy 3 lub 4-osobowe. Sprawozdanie wysyła każdy za siebie, tak więc wykonuj dokładne notatki!

Kwestia podziału poniższych zadań pozostawiona jest grupie.

Rekurencja w C++

How do I compile

“Dwie podstawowe umiejętności każdego inżyniera to znajomość angielskiego i obsługa Google”

Pracując na systemie GNU/Linux, należy ustalić w jaki sposób kompilować programy napisane w języku C++. Prawie każda instalacja tego systemu dysponuje stosownym kompilatorem.

Rozwiązanie to można lubić lub nie. Z pewnością jest wygodne (jeśli akurat jesteśmy programistą C lub C++). Jest ono również pomocne dla hakerów, którzy włamując się na system linuksowy mają już wszystkie narzędzia pod ręką. Korzystał z tego również słynny robak Morrisa, który unieruchomił 10% ówczesnego Internetu.

Jako dowód działania, skompiluj prosty program:

```
#include <iostream>
#include <unistd.h>

using namespace std;

int main() {
    cout << "The magic number is " << geteuid() << endl;
}
```

Jak przed rozpoczęciem zajęć powie, jak uzyskać wyjście tego programu bez kompilatora, dostaje zwolnienie z zaliczenia i ocenę bardzo dobry. Warto czytać instrukcje z wyprzedzeniem, jakiegokolwiek to wyprzedzenie by nie było.

Ustal jak kompiluje się programy w C++ na systemie Linux. Użyj Google albo zapytaj prowadzącego.
Skompiluj podany program. Co wyświetlił?‘

Algorytm Euklidesa

Algorytm Euklidesa służy do wyznaczania największego wspólnego dzielnika dwóch liczb.

Zapisz wzór na największy wspólny dzielnik liczb x i y .
Matematyczny.
Policz "na piechotę" NWD 36 i 48.

Przypomnij sobie jak wyglądają funkcje rekurencyjne w C++. Na przykład, funkcja licząca n -tą liczbę Fibonacciego:

```
int fib(int n) {
    if (n == 0) {
        return 0;
    }

    if (n == 1) {
        return 1;
    }

    return fib(n-1) + f(n-2);
}
```

Dostrzeż pewne podobieństwo w strukturze. NWD przyjmuje co prawda 2 parametry, ale nie jest to dużym problemem. W razie problemów - Google twoim przyjacielem.

Napisz w C++ funkcję liczącą NWD.
Funkcja ta powinna wypisywać swoje parametry przy każdym wywołaniu.
Policz przy użyciu tej funkcji NWD 36 i 48.

Podstawy programowania obiektowego

Przydatne materiały:

- Podstawowe wiadomości o klasach
- Metody w klasach
- Pierwiastek w C++

Programowanie obiektowe pozwala nam podzielić problem na mniejsze kawałki, z których każdy zostanie rozwiązany oddzielnie. *Obiekt* to ideowo pewna grupa danych, która dysponuje operacjami rozwiązującymi jej problemy.

W tym zadaniu należało będzie stworzyć klasę odpowiadającą pewnej istniejącej specyfikacji. Specyfikacją tą jest istniejący program, który po “doklejeniu” do kodu z klasą i nagłówka programu powinien wykonywać się poprawnie. Im więcej linijek tego programu wykona się poprawnie - tym lepiej.

Porada: spróbuj zacząć od pierwszej części, a potem w miarę rozwoju klasy dodawaj kolejne. Są one w dużej mierze niezależne. Pracujecie w grupie - każda osoba może opracowywać inną część, aby było szybciej!

```
int main() {
    // powinno wypisac: "X: 3, Y: 4"
    Wektor w1(3, 4);
    w1.wypisz();

    // powinno wypisac: "Dlugosc wektora to 5", "Dlugosc wektora to 13"
    Wektor w2(3, 4);
    cout << "Dlugosc wektora to " << w2.dlugosc() << endl;
    Wektor w3(5, 12);
    cout << "Dlugosc wektora to " << w3.dlugosc() << endl;

    // powinno wypisac "X: 4, Y: 6", "X: 3, Y: 4);
    Wektor w4(3, 4);
    w4.dodaj(1, 2);
    w4.wypisz();
    w4.dodaj(-1, -2);
    w4.wypisz();

    // powinno wypisac "X: 0, Y: 0"
    Wektor w5;
    w5.wypisz();

    // powinno wypisac "X: 2, Y: 2", oraz "X: 4, Y: 4"
    Wektor w6(1, 1);
    w6.pomnoz(2);
    w6.wypisz();
    w6.pomnoz(2);
    w6.wypisz();

    // powinno wypisac "X: 1, Y: 1", "X: 0.5, Y: 0.5"
    Wektor w7(2, 2);
    w7.podziel(2);
    w7.wypisz();
    w7.podziel(2);
    w7.wypisz();

    // Powinno wypisac "X: 0, Y: 0"
```

```

Wektor w8(4, 3);
w8.zeruj();
w8.wypisz();

// Powinno wypisac "X: 10, Y: 10"
Wektor w9(5, 5);
w9.dodaj(w9);
w9.wypisz();
}

```

Przy `w9` będzie niezbędne przeciążenie metody `dodaj()`, zaś przy `w5` - przeciążenie konstruktora.

Funkcję `wypisz()` już napisano, ale i tak musisz wymyślić w jaki sposób umieścić ją w deklaracji klasy:

```

void Wektor::wypisz() {
    cout << "X: " << x << ", Y:" << y << endl;
}

```

W sprawozdaniu zamieść kompletny kod rozwiązania. Zamieść w nim również imiona i nazwiska członków twojej grupy.

Wyślij sprawozdanie

To już prawie koniec zajęć. Zapisz i wyślij swoje sprawozdanie.

Wyślij je na adres `sprawozdania@henrietta.com.pl`.
W tytule umieść imię, nazwisko i numer zadania.
Załącz sprawozdanie, lub wklej je w treść maila.

Copyright (c) 2017 Piotr Maślanka. Niektóre prawa zastrzeżone.

Kod źródłowy dostępny na http://git.dms-serwis.com.pl/henrietta/inf2_eedi.