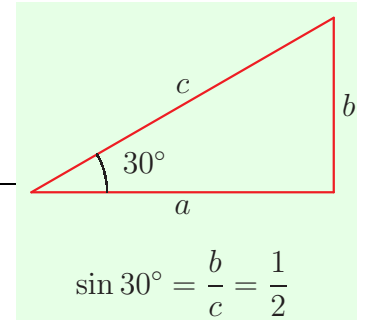


Programowanie obiektowe II

Laboratorium nr 1

Stefan Sokołowski



Zadanie 1 (co jest grane?):

Jak wiadomo, $\sin 30^\circ = 0.5$. Wyjaśnić niezgodne z tą równością zachowanie następującego programiku:

```
class Bład1 {
    public static void main(String[] argumenty) {
        System.out.println("\n sin(30) == " + Math.sin(30) + "\n");
    }
}
```

Uwaga:

Program proszę przetestować na serwerze linuksowym **unixlab**. Ściąga z najważniejszych komend linuksowych, potrzebnych do zdalnego uruchomienia programu w Jawie, znajduje się pod

iis.ans-elblog.pl/~stefan/Dydaktyka/ProgObiII/Slajdy/praca_z_linuksem.html.

Zadanie 2 (komendy poprawne i błędne):

Załóżmy, że zdefiniowano klasę **Samochod** z **publicznymi** własnościami i metodami:

własności
int kola;
int siedzenia;
int kierownica;

metody
Boolean start();
void hamuj(int jak)
void przyspiesz();

Niech a będzie obiektem klasy **Samochod**. Które z poniższych komend są składniowo poprawne¹ i dlaczego?

1. $a.kola = 4;$
2. $a.siedzenia(5);$
3. $start();$
4. $a.start(2);$
5. $a.start();$
6. $System.out.println(a.start());$
7. $System.out.println(a.hamuj(2));$
8. $a = a.start();$
9. $int wynik = a.start();$
10. $Boolean wynik = a.przyspiesz();$

Zadanie 3 (projektowanie klasy):

Dla jak zdefiniowanych klas następujący kod jest poprawny?

¹Składniowo (syntaktycznie) poprawne — nie spowodują błędów traktacji kompilacji, chociaż mogą wygenerować sygnał błędów w trakcie działania lub doprowadzić do błędnych wyników.

```
Regal r = new Regal();
r.dodajPolke(new Polka(20));
Polka p = new Polka(20);
p.dodajKsiazke(new Ksiazka(1111));
r.dodajPolke(p);
System.out.println(r.ilePolek());
System.out.println(p.ileKsiazek);
```

Zadanie 4 (projektowanie klasy):

Napisałem program w Jawie, obsługujący przychody ze sprzedaży:

```
class Sklep {
    public static void main(String[] args) {
        Sprzedawcy
            Jacek = new Sprzedawcy(),
            Iwona = new Sprzedawcy(),
            Roman= new Sprzedawcy();
            . . . . .
        Jacek.sprzedaz(5.50);
        Roman.sprzedaz(10.80);
        Roman.sprzedaz(7.00);
        Iwona.sprzedaz(11.20);
        Jacek.sprzedaz(2.30);
            . . . . .
        System.out.println(
            "\n Sumaryczny przychod: " + Sprzedawcy.uzysk() + "\n"
        );
    }
}
```

Program ma wyświetlać sumaryczny przychód pochodzący z pojedynczych transakcji przeprowadzonych przez sprzedawców.

Żeby ten program działał, potrzebna jest jeszcze klasa **Sprzedawcy**. Proszę ją tak napisać,

1. żeby nie zawierała żadnych własności publicznych,
2. żeby nie wymagała żadnych zmian, gdybym w programie głównym chciał dopisać nowych sprzedawców (pierwszy wiersz **czzerwonych kropek**),
3. żeby nie wymagała żadnych zmian, gdybym w programie głównym chciał dopisać nowe transakcje sprzedaży (drugi wiersz **czzerwonych kropek**).

W samym programie głównym proszę niczego nie zmieniać i niczego nie dopisywać.

Zadanie 5:

Proszę zaprojektować klasę `Macierze` obsługującą podstawy rachunku macierzowego. Powinna ona dostarczać środki dla

- tworzenia macierzy i wypełniania jej liczbami,
- drukowania macierzy,
- transponowania macierzy,
- dodawania macierzy o zgodnych wymiarach,
- mnożenia macierzy o odpowiednich wymiarach,
- wyliczania wyznacznika macierzy kwadratowej.

Zadanie 6 (domowe specjalne):

Działania na trójkątach można zrealizować na różne sposoby:

- albo zdefiniować klasę `Trojkat` tak, żeby można było konstruować trójkąty:

```
Trojkat jakis3kat = new Trojkat(...parametry...);
```

a następnie pytać o ich pola i obwody:

```
double pole = jakis3kat.pole();  
double obwod = jakis3kat.obwod();
```

- albo zdefiniować klasę `Trojkat` bez takich publicznych metod, a za to dodatkowo zdefiniować klasę obsługującą `Trojkaty` tak, żeby można było konstruować trójkąty:

```
Trojkat jakis3kat = new Trojkat(...parametry...);
```

a następnie pytać o ich pola i obwody:

```
double pole = Trojkaty.pole(jakis3kat);  
double obwod = Trojkaty.obwod(jakis3kat);
```

Proszę wypróbować oba rozwiązania. Oprócz wymienionych klas potrzebne będą programy testujące te rozwiązania.

Wskazówka:

Pole trójkąta o podanych wierzchołkach najłatwiej jest liczyć ze wzoru Herona — por. np.

https://pl.wikipedia.org/wiki/Wzór_Herona.

Uwaga:

Pierwszej osobie, która przyśle mi (na s.sokolowski@ans-elblag.pl) poprawne rozwiązanie tego zadania, przysługują punkty do zaliczenia laboratorium. Liczba tych punktów zależy od oryginalności i elegancji rozwiązania.