

Обектно-ориентирано програмиране

спец. Информатика, 2013/2014

Задачи

за задължителна самоподготовка

№4

[1] **Задача 2.2.10.** Да се намери резултатът от изпълнението на програмата:

```
#include <iostream.h>
class F
{ private:
    int a, b;
public:
    // конструктор по подразбиране
    F();
    // конструктор с един параметър
    F(int);
    // конструктор с два параметъра
    F(int, int);
    // извеждане на член-данните на класа
    void display() const;
};
F::F()
{ a = b = 0;
}
F::F(int x)
{ a = x;
}
F::F(int x, int y)
{ a = x;
  b = y;
}
void F::display() const
{ cout << a << " " << b << endl;
}
int main()
{ F e1, e2(1), e3(1, 2);
  e1.display();
  e2.display();
  e3.display();
  return 0;
}
```

[2] Задача 2.2.16. Като се използва класът Point, определящ точка от равнината и дефиниран по следния начин:

```
class Point
{ public:
    Point(double, double);
    double xcoord() const;
    double ycoord() const;
private:
    double x,           // абсциса
          y;           // ордината
};
Point::Point(double a, double b)
{ x = a;
  y = b;
}
double Point::xcoord() const
{ return x;
}
double Point::ycoord() const
{ return y;
}
```

да се дефинират член-функциите на клас Line, определящ права линия, преминаваща през две точки от равнината и деклариран по следния начин:

```
class Line
{ public:
    // конструктор на линия, минаваща
    // през зададени две точки
    Line(const Point&, const Point&);
    // функция за достъп до едната точка
    Point point1() const;
    // функция за достъп до другата точка
    Point point2() const;
    // разстояние между двете точки, през
    // които минава правата линия
    double len() const;
    // намира ъгловия коефициент на линия
    double slope() const;
    // намира пресечната точка на две линии
    Point intersection(const Line&) const;
private:
    Point p1, p2;           // точки, определящи линия
};
```

Да се напише програма, която намира ъгловите коефициенти и пресечната точка на линиите, които се явяват диагонали на правоъгълника с върхове: (0, 0), (8, 0), (8, 4) и (0, 4), както и дължината на диагонала на правоъгълника.

[3] Задача 2.2.17. Да се дефинира клас BankAccount, определящ банкова сметка на клиент, съдържаща: име на клиент (символен низ с дължина до 23 знака), номер на банкова сметка (символен низ с дължина до 15 знака) и налична сума на клиент (реално число). Класът да притежава методи, чрез които: да създава банкова сметка на клиент, да извежда на екрана банкова сметка, да внася и тегли пари от банкова сметка. Да се дефинира главна функция, която демонстрира класа.

[4] Задача 2.2.21. Да се намери резултатът от изпълнението на програмата:

```
#include <iostream.h>
class K
{ private:
    int a;
    double b;
public:
    K(int, double);
    K(K const &);
    void display() const;
};
K::K(int x, double y)
{ a = x;
  b = y;
}
K::K(K const &r)
{ a = r.a;
  b = r.b;
}
void K::display() const
{ cout << a << " " << b << endl;
}
int main()
{ K e1(4, 1.8), *e2 = &e1;
  e1.display();
  e2->display();
  K e3 = *e2;
  e3.display();
  return 0;
}
```

[5] Задача 2.2.25. Лекар трябва да поддържа статистика за броя на посещенията, които са направили пациентите му. За всеки пациент лекарят записва името, датата на раждане и броя на посещенията, които е направил пациентът през една календарна година.

Да се напише програма, която за лекар създава списък от пациентите му, намира средния брой на посещенията на пациентите, както и пациентите, направили повече от зададен брой посещения.