

## Урок 13.

### Задание для самостоятельной работы.

1. Составьте объектно-ориентированную программу, в которой реализуйте класс *fraction*, описывающий обыкновенные дроби вида P/Q (P — целое, Q — натуральное) и операции с ними. В качестве полей данных класса задайте две переменные соответствующих типов. В состав методов класса включите следующие функции:

- конструктор без аргументов;
- конструктор с двумя аргументами;
- функцию, реализующую вывод результата операции с дробями на экран.

Добавьте в состав методов созданного класса методы, осуществляющие следующие операции с дробями:

- сложение;
- вычитание;
- умножение;
- деление;
- сокращение дроби.

Дополнительно реализуйте следующую организацию программы:

- разместите объявление методов класса *fraction* в описании класса, а их определения — вне класса;
- разместите объявление класса *fraction* в отдельном заголовочном файле.

В функции *main()* с помощью директивы *#include* подключите созданный класс и продемонстрируйте использование объектов класса *fraction*.

Результат работы программы может выглядеть следующим образом.

```
Введите операцию в виде a/b + c/d
1/2 + 1/2
1/2 + 1/2 = 1
Продолжаем? (y/n):y
Введите операцию в виде a/b + c/d
1/4 - 1/2
1/4 - 1/2 = -1/4
Продолжаем? (y/n):y
Введите операцию в виде a/b + c/d
1/4 * 2/3
1/4 * 2/3 = 1/6
Продолжаем? (y/n):y
Введите операцию в виде a/b + c/d
3/5 / 5/6
3/5 / 5/6 = 18/25
Продолжаем? (y/n):n
```

2. Составьте объектно-ориентированную программу, позволяющую производить операции с данными в формате времени (11:59:59). Для этого:
- создайте класс с именем *time*, содержащий три поля данных типа *int* (например, *hrs*, *mins* и *secs*), предназначенные для хранения часов, минут и секунд; включите в состав класса два конструктора, один из которых должен инициализировать поля класса нулевыми значениями, а другой — заданным набором значений;
- создайте и включите в состав класса три метода: первый метод (константный) будет выводить значения полей на экран в формате 11:59:59, а два других метода будут осуществлять сложение (например, *add\_time()*) и вычитание (например, *ded\_time()*) двух объектов типа *time*, передаваемых им в качестве аргументов;
- создайте в функции *main()* два инициализированных константных объекта и два неинициализированных простых объекта; затем одному из неинициализированных объектов присвойте результат сложения, а второму неинициализированному объекту — результат вычитания двух инициализированных объектов;
- выведите значения всех четырех объектов на экран.

Возможный результат работы программы представлен ниже.

```
time1 = 05:59:29
time2 = 04:30:30
time3 = 10:29:59
time4 = 01:28:59
```

3. Реализуйте *свой вариант* программы из урока №7 с использованием объектно-ориентированного подхода. Для этого:
- создайте класс *stock*, объектами которого будут позиции по ценным бумагам клиента брокерской фирмы; класс должен содержать четыре поля данных *char\* name*, *char\* type*, *int amount* и *float price* для хранения информации о наименовании, типе, количестве и цене ценных бумаг в портфеле клиента; в состав класса также включите три метода, один из которых будет инициализировать поля класса значениями, другой – пересчитывать позиции с учетом заключенных сделок, а третий – выводить информацию на экран;
  - в качестве методов класса используйте функции, созданные в уроке №7; подумайте, какие методы можно сделать *константными*;
  - расположите определения всех трех методов класса *вне класса*;
  - создайте три массива объектов класса *stock* (например, *sec[3]*, *buy[3]* и *sell[3]*), которые будут хранить информацию о позиции по конкретной ценной бумаге и о сделках по ее покупке и продаже соответственно;
  - в функции *main()* проинициализируйте созданные массивы теми же значениями, что и в программе из урока №7, а затем пересчитайте состояние портфеля с учетом заключенных сделок и выведите результаты на экран.

Результаты работы обеих программ должны совпадать.

Наименование ЦБ	Тип ЦБ	Количество	Цена ЦБ	Стоимость позиции
Газпром	акция	50	141.56	7078.00
Сбербанк	акция	135	101.87	13752.45
Газпром	облигация	214	1001.26	214269.64
Покупки				
Газпром	акция	50	136.54	6827.00
Продажи				
Сбербанк	акция	35	103.14	3609.90
Газпром	облигация	114	1004.63	114527.82
Результат				
Газпром	акция	100	139.05	13905.00
Сбербанк	акция	100	101.87	10187.00
Газпром	облигация	100	1001.26	100126.00

4. Создайте класс *line*, с помощью которого можно будет рисовать на экране линии произвольной длины, произвольной толщины, произвольного типа и произвольного цвета.

В качестве полей данных класса *line* задайте:

- координаты двух концов линии;
- цвет линии;
- тип линии;
- толщину линии.

В состав методов класса включите:

- конструктор с пятью параметрами;
- функцию по прорисовке линии.

Реализуйте следующую организацию программы:

- разместите объявление методов класса *line* в описании класса, а их определения – вне класса;
- разместите объявление класса *line* в отдельном заголовочном файле.

В функции *main()* с помощью директивы *#include* подключите созданный класс и продемонстрируйте использование массива объектов класса *line*, реализовав вывод на экран геометрической фигуры «Пирамида».

Возможный результат работы программы представлен ниже.

