Zadanie 1

Stwórz klasę **Point**, obiekty której będą reprezentować punkty na płaszczyźnie kartezjańskiej (o współrzędnych typu **double**). Klasa powinna definiować:

- konstruktor domyślny tworzący punkt $(0, 0)$;
- konstruktor jednoparametrowy tworzący punkt o obu współrzędnych równych przekazanej do konstruktora wartości;
- konstruktor dwuparametrowy (dwie współrzędne);
- metody **getX()** i **getY()** zwracające wartości odpowiednich współrzędych;
- metody **setX(double)** i **setY(double)** zmieniające wartości odpowiednich współrzędnych;
- metody **transX(double)** i **transY(double)** dodające przekazaną wartość do odpowiednich współrzędnych punktu (translacja);
- metodę **transXY(double,double)** dokonującą translacji puntu o podane wartości;
- statyczną funkcję składową **dist** zwracającą odległość pomiędzy dwoma punktami przekazanymi przez referencje.

Uwaga: metody **setX**, **setY**, **transX**, **transY** i **transXY** powinny zwracać przez referencję obiekt (punkt) na rzecz którego nastąpiło wywołanie (czyli ***this**).

Po dostarczeniu odpowiednich definicji następujący program:

download *Point.cpp*

```
#include <iostream>
#include <cmath>

class Point {
    double x, y;
public:
    Point();
    Point(double s);
    Point(double x, double y);

    double getX() const;
    double getY() const;
    Point& setX(double xx);
    Point& setY(double yy);
    Point& transX(double dx);
    Point& transY(double dy);
    Point& transXY(double dx, double dy);
    static double dist(const Point& p, const Point& q);
};
```

```
int main() {
    Point p;
    Point q(1);
    Point r(1,2);
    p.transXY(5,6);
    q.transX(-1).transY(-1).transXY(2,2);
    r.setX(10).transY(-8);
    std::cout << Point::dist(p,q) << std::endl;
    std::cout << Point::dist(p,r) << std::endl;
}
```

powinien wypisac liczby 5 i 13.

Zadanie 2

Zdefiniować klasę **Rect** o prywatnych polach opisujących długości boków (**double**). Zdefiniuj

- konstruktor domyślny, tworzący kwadrat o boku 1;
- konstruktor jednoparametrowy, tworzący kwadrat o podanym boku;
- konstruktor dwuparametrowy (dwa boki);
- metody **getA()** i **getB** zwracające odpowiednie boki prostokąta;
- metodę **getDiagonal()** zwracającą długość przekątnej prostokąta;
- metodę **getArea()** zwracającą pole powierzchni prostokąta;
- metodę **isLargerThan(const Rect&)**, która zwraca **true** gdy *ten* prostokąt ma większe pole od tego przekazanego w argumencie, a **false** w przeciwnym przypadku;
- metodę **info()**, która wypisuje informację o prostokącie, na przykład w formie `Rect[2,3]` (słowo 'Rect' i w nawiasach kwadratowych długości boków).

Przetestuj wszystkie konstruktory i metody w funkcji **main**.

Zadanie 3

Napisz klasę **Letter**, która będzie służyć do analizowania tekstów. Obiekty klasy zawierają tablicę 26 liczb typu **int** odpowiadających literom alfabetu angielskiego (element o indeksie 0 — literze 'A', ... , o indeksie 25 — literze 'Z'). Konstruktor klasy pobiera tekst (jako wskaźnik typu **const char***) i wypełnia tablicę liczbami wystąpień poszczególnych liter.

Uwaga: litery duże i odpowiadające im małe są uważane za takie same.

Klasa udostępnia metody

- **getMostFrequent** zwracającą znak, który występuje najczęściej;
- **numOfDifferent** zwracającą liczbę różnych liter;
- **printFrequencies** drukującą „histogram" opisujący liczby wystąpień poszczególnych liter: wyrównany od dołu, z kolumnami złożonymi z tylu gwiazdek, ile wynosiła liczba wystąpień odpowiedniej litery (patrz przykład poniżej).

Na przykład następujący program

download *LetterHist.cpp*

```
#include <iostream>

class Letters {
    int letters[26]{0};
    // ...
};

int main() {
    const char* text =
            "To be, or not to be- that is the questiona\n:"
            "Whether 'tis nobler in the mind to suffer\n"
            "The slings and arrows of outrageous fortune\n"
            "Or to take arms against a sea of troubles,\n"
            "And by opposing end them.";
    Letters lett(text);
    std::cout << "Most frequent letter:        "
              << lett.getMostFrequent() << std::endl;
    std::cout << "Number of different letters: "
              << lett.numOfDifferent()  << std::endl;
    std::cout << "Frequency table of all letters:\n";
    std::cout << std::endl;
    lett.printFrequencies();
}
```

powinien wydrukować

```
Most frequent letter:        T
Number of different letters: 21
Frequency table of all letters:

                     *
                *    *
    *           *    *
    *           *    *
    *           *    *
    *           *    *
    *           *   **
*   *          **   **
*   *          **  ***
*   *          **  ***
*   *          **  ***
*   *   *      **  ***
*   *  **      **  ***
*   *  **      **  ****
**  ** **      **  ****
```

```
** ******    **  ****
** ******  ****  ****
** ******  ***** **** *
** ****** ********** * *
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```
