

Подзадание 1.

- a) Study the Java code below in detail. Make sure you understand what is going on.

```
public interface PlaneFigure{
    public double getArea();
    public double getPerimeter();
}

public class Rect implements PlaneFigure{
    private double width;
    private double height;
    public Rect(double w, double h){
        width = w;
        height = h;
    }
    public double getArea(){
        return width*height;
    }
    public double getPerimeter(){
        return 2*width + 2*height;
    }
}

public class Driver{
    public static void main(String[] args){
        PlaneFigure pf = new Rect(2,3);
        test(pf);
    }
    private void test(PlaneFigure pf){
        System.out.println(pf.getArea());
        System.out.println(pf.getPerimeter());
    }
}
```

- b) Convert this Java code to C++. Create a separate header file and code file for the Rect class.
- c) Check your C++ implementation.
- d) Have your solution of this exercise checked by your lecturer.

Подзадание 2

Вам дан класс Foo:

```
struct Foo {
    void say() const { std::cout << "Foo says: " << msg << "\n"; }
protected:
    Foo(const char *msg) : msg(msg) { }
private:
    const char *msg;
};
```

Как видно, создатель класса не хотел чтобы его использовали и "спрятал" конструктор класса. Но вам очень нужен объект этого класса, чтобы передать его в функцию foo_says:

Задание по C++ №7, весна 2019 года

```
void foo_says(const Foo& foo) { foo.say(); }
```

В этом задании вам нужно реализовать функцию `get_foo` (сигнатура которой намерено не приводится в задании полностью, вам нужно подумать и вывести ее самостоятельно) так, чтобы следующий код компилировался и работал:

```
foo_says(get_foo(msg));
```

Где `msg` — произвольная C-style строка.

Подзадание 3.

a) Study the implementation of the `main` function given below.

```
int main()
{
    Date d(1,2,2000);
    cout << d << endl;

    Person p("Filip", "Wallaert", d, true);
    cout << p << endl;
    const Person p2("Eva", "Beyens", Date(21,8, 1995), false);
    cout << p2 << endl;

    Person * p3 = new Student("Karen", "Verbinnen",
                             Date(13, 5, 1985), false, 1234);
    cout << *p3 << endl;
    return 0;
}
```

Running this code, should give the following output:

```
1/2/2000
Wallaert Filip: male, born on 1/2/2000
Beyens Eva: female, born on 21/8/1995
Verbinnen Karen: female, born on 13/5/1985, student number: 1234
```

b) Implement all classes which are required to compile and run the above `main` function with the desired output. Your implementation should satisfy the following criteria:

- A separate header file and code file should be created for each class.
- Make use of data encapsulation.
- Provide getter and setter methods for all private member variables. You may assume the provided day, month and year values form a valid date.
- Hint: The C++ standard library has a `string` class. Include the header file `string` to use this class.