

1. Реализуйте класс Student (Студент), который будет наследовать от класса User, подобно тому, как это сделано в теоретической части урока. Этот класс должен иметь следующие свойства: name(имя, наследуется от User), surname (фамилия, наследуется от User), year (год поступления в вуз). Класс должен иметь метод getFullName() (наследуется от User), с помощью которого можно вывести одновременно имя и фамилию студента. Также класс должен иметь метод getCourse(), который будет выводить текущий курс студента (от 1 до 5). Курс вычисляется так: нужно от текущего года отнять год поступления в вуз. Текущий год получите самостоятельно.

2. Реализуйте класс Rectangle, о котором я рассказываю в видео в теоретической части урока. У него должны быть следующие свойства: ширина width, высота height. Также у него должны быть следующие методы: получить ширину getWidth, установить ширину setWidth, получить высоту getHeight, установить высоту setHeight.

3. Разработать три класса, которые следует связать между собой, используя наследование: 1. класс Product, который имеет три элемент-данных — имя, цена и вес товара (базовый класс для всех классов); 2. класс Buy, содержащий данные о количестве покупаемого товара в штуках, о цене за весь купленный товар и о весе товара (производный класс для класса Product и базовый класс для класса Check); 3. класс Check, не содержащий никаких элемент-данных. Данный класс должен выводить на экран информацию о товаре и о покупке (производный класс для класса Buy);

4. 1) Создать класс Man (человек), с полями: имя, возраст, пол и вес. Определить методы задания имени, возраста и веса. Создать производный класс Student, имеющий поле года обучения. Определить методы задания и увеличения года обучения.

5. Создать базовый класс Car, характеризуемый торговой маркой (строка), весом, мощностью. Определить методы переназначения и изменения мощности. Создать производный класс Lorry (грузовик), характеризуемый

также грузоподъемностью кузова. Определить для него функции переназначения марки и изменения грузоподъемности.

6. Создайте класс Родителя и унаследуйте от него класс Ребёнка. Сделайте так, чтобы родителю при создании можно было задать имя, а с помощью метода `name` у экземпляра класса можно было это имя посмотреть. Создайте метод послушности так, чтобы все родители были послушные, а дети — нет. Например: Василий Игоревич послушный: `true` Маша послушная: `false` Коля послушный: `false` Создайте класс родитель (`Parent`) с тремя методами. Конструктор просто присваивает родителю имя. Метод `name` возвращает имя родителя. И, наконец, метод `obedient`, который просто возвращает `true`. А после этого создайте класс ребёнка (`Child`), унаследовав его от родителя. Переопределите у ребёнка метод `obedient` так, чтобы он возвращал `false`.

7. Напишите заготовку для небольшого магазинчика, который торгует фильмами, музыкой и книгами. Создайте класс продукта, у экземпляров которого есть два поля: цена и количество на складе. При создании нового продукта можно передать значения цены и остатка. Сделайте также у продукта возможность посмотреть цену. А потом унаследуйте от этого класса три других: для музыкального альбома, для книги и для фильма соответственно. Ну 9 и создайте в основной программе какой-нибудь продукт, например, «фильм Леон». Выведите его стоимость в консоль. Фильм Леон стоит 290 руб. Создайте классы `Product`, `Movie`, `Disk` и `Book`, каждый в своём файле соответственно. 2-й, 3-й и 4-й классы отнаследуйте от класса продукта. Напишите у класса продукта конструктор `initialize(price, amount)`, который записывает в поля значения своих аргументов. Потом подключите эти файлы в основную программу с помощью `require_relative` и создайте фильм.

```
movie = Movie.new(290, 4)
```