

ООП. Модуль 1

Задание выполняется на хосте p2, не забудьте создать отдельную папку

Домашнее задание состоит из нескольких небольших отдельных не связанных между собой задач. Каждое задание делается в отдельном файле с уникальным пространством имен.

1. Домашний зоопарк.

- a. Опишите следующие классы
 - Cat - Кот
 - Dog - Собака
 - Fish - Рыбка
- b. Каждый класс должен содержать свойство public \$name - кличка животного
- c. Создайте конструктор класса, в котором это свойство будет проинициализировано (ему должно быть установлено значение)
- d. Создайте трех котиков, пять собачек и одну рыбку, у каждого животного должна быть своя кличка.

2. Голодный кот

- a. Опишите класс HungryCat. Кот должен иметь следующие свойства: кличку, цвет и любимая еда, которые должны быть проинициализированы в конструкторе.

- b. Создайте и реализуйте метод `public function eat($food)` - внутри этого метода должна выводиться строка такого формата: "Голодный кот <кличка кота>, особые приметы: цвет - <цвет кота>, съел <название еды>"
- c. Если кот съел любимую еду, то также нужно вывести строку " " и замурчал 'mrrrrrr' от своей любимой еды"
- d. Создайте двух совершенно разных голодных котов, и накормите каждого пятью разными блюдами. Оба котика обязательно должны помурчать.

3. Фабрика игрушек

- a. Опишите два класса
 - ToyFactory Фабрика Игрушек
 - Toy - игрушка, у каждой игрушки есть название и цена, которые должны быть проинициализированы в конструкторе.
- b. Реализуйте метод создания игрушки на фабрике `public function createToy($name)`, который должен возвращать новый экземпляр игрушки со случайной ценой
- c. Создайте случайное количество игрушек (воспользуйтесь функцией `rand(int min, int max)`) не менее 5 но не более 20 шт. Названия игрушек могут повторяться, но должны встречаться игрушки с разными названиями (но не обязательно проверять, чтобы среди созданных игрушек в одном процессе выполнения программы обязательно присутствовали разные игрушки).
- d. Выведите на экран построчно строки "<Название игрушки> - <Стоимость игрушки>"
- e. Выведите на экран строку "Итого - <Сумма стоимостей всех игрушек>"

4. Уведомления для клиента

- a. Опишите класс User - пользователь, у пользователя обязательно должно быть указано ФИО и email, а также могут быть указаны, а могут и нет пол, возраст и телефон, которые должны быть проинициализированы в конструкторе
- b. создайте и реализуйте методы пользователя
 - notifyOnEmail(\$message) - вызывает функцию send()
 - notifyOnPhone(\$message) - вызывает функцию send()
 - notify(\$message)
 - censor(\$message)
 - send(...) - выводит строку "Уведомление клиенту: <ФИО> на <канал уведомления> (<email или телефон клиента>): <message>", подумайте какие параметры должен принимать этот метод. Например: "Уведомление клиенту: Владимир Николаевич на email (user1@email.com): goodbye"

PS Не забывайте указывать уровни доступа к вашим методам, у каждого метода должен быть уровень доступа (инкапсуляция), о ней мы подробнее расскажем в следующем уроке.

- c. метод notify(\$message), в случае если мы не уверены что клиенту есть 18 лет - должен произвести цензуру сообщения (придумайте любую реализацию метода censor()), затем нужно отправить уведомление клиенту на email и на телефон, если телефон указан
- d. Создайте Разных пользователей и отправьте им уведомления (вызвав метод notify). Пользователи должны быть созданы так, чтобы был случай, когда не был указан телефон и когда бы был, а были случаи когда клиенту есть 18 и нет.

5. Интернет магазин

а. Опишите классы

- Order - Заказ, должен содержать корзину
- Basket - Корзина, должен массив товаров с указанием количества каждого
- Product - Товар, у каждого товара есть название и цена, которые должны быть проинициализированы в конструкторе

б. Класс Order должен содержать два метода

- `getBasket()` - возвращает корзину товара
- `getPrice()` - возвращает общую стоимость товаров в корзине

с. Класс Basket должен следующие методы

- `addProduct(Product $product, $quantity)` - добавляет товар в корзину
- `getPrice()` - возвращает стоимость товаров в корзине
- `describe()` - выводит или возвращает информацию о корзине, в виде строки:
"<Наименование товара> - <Цена одной позиции> - <Количество>"

д. Класс Product должен содержать два метода

- `getName()` - возвращает наименование товара
- `getPrice()` - возвращает стоимость товара

е. Реализуйте все методы

ф. Создайте корзину, заполните ее товарами - создайте заказ на основе этой корзины

- `$order = new Order($basket);`

г. Выведите информацию о корзине этого заказа и выведите общую стоимость заказа

- h. Создайте нового клиента "Николай Николаича" - из предыдущего задания (не копируя описание его класса в этот файл) и отправьте ему уведомление с текстом:
"для вас создан заказ, на сумму: <сумма заказа> Состав: <информация о корзине>"