

## Домашнее задание №4

### Задание #1. Практика с ООП

Представьте что вы создаете сайт для компании сдающих автомобили поминутно (каршеринг). У компании есть ряд тарифов. Вам необходимо реализовать каждый тариф в своем классе. У каждого тарифа две основные характеристики - цена за 1 км, цена за 1 минуту. Каждый тариф обязан иметь метод для подсчета цены поездки. В некоторых тарифах возможно использование дополнительных услуг, которые могут быть объявлены с помощью механизма trait. Минимальный возраст водителя - 18 лет, максимальный - 65 лет. В случае возраста от 18 до 21 года, тариф повышается на 10%. Ваша задача - посчитать цену, которую получит пользователь, если проедет X км и Y минут по тарифу Z.

#### Тариф базовый

- цена за 1 км = 10 рублей
- цена за 1 минуту = 3 рубля

#### Тариф почасовой

- Цена за 1 км = 0
- Цена за 60 минут = 200 рублей
- Округление до 60 минут в большую сторону

#### Тариф суточный

- цена за 1 км = 1 рубль
- цена за 24 часа = 1000 рублей
- округление до 24 часов в большую сторону, но не менее 30 минут. Например 24 часа 29 минут = 1 сутки. 23 часа 59 минут = 1 сутки. 24 часа 31 минута = 2 суток.

#### Тариф студенческий

- цена за 1 км = 4 рубля
- цена за 1 минуту = 1 рубль
- Возраст водителя не может быть более 25 лет

#### Дополнительные услуги (трейты):

- грс в салон - 15 рублей в час, минимум 1 час. Округление в большую сторону. Доступно на всех тарифах
- Дополнительный водитель - 100 рублей единоразово, доступен на всех тарифах кроме базового и студенческого

#### Ожидаемая реализация:

1. Создать интерфейс, который будет содержать описание метода подсчета цены и другие необходимые функции

2. Реализовать абстрактный класс, который будет описывать основные методы и заниматься определением возраста и имплементировать описанный в п. 1 интерфейс
3. Все тарифы должны наследоваться от абстрактного тарифа из п. 2
4. Дополнительные услуги реализовываются с помощью механизма трейтов. Они доступны по умолчанию, но не активированы. Активация происходит путем передачи аргументов в конструктор класса.

Пример вызова:

Тариф базовый(10 км, 1 час, 20 лет, без доп. услуг) =  $(10 * 1 + 60 * 3) * 1.1 = 209$  руб;

### Альтернативный вариант:

Опишите несложный, но вполне конкретный автомобиль с помощью ООП.

1. Автомобиль должен иметь функцию движения с заданным расстоянием и скоростью и направлением. Например “Автомобиль->Двигаться(200 метров, 10 м/с, вперед)”. При начале движения автомобиля:
  - a. вы включаете двигатель
  - b. Включаете нужную передачу (вперед\назад)
  - c. двигаетесь в соответствии с параметрами двигателя, при необходимости включая охлаждения.
  - d. выключаете двигатель и коробку передач
2. Ваш автомобиль должен иметь двигатель. Двигатель должен иметь функцию включения, выключения и охлаждения. Считается что двигатель мгновенно разгоняется до указанной скорости и двигается равномерно все это время. Охлаждение может быть включено ТОЛЬКО двигателем. Параметры двигателя:
  - a. Кол-во лошадиных сил. 1 лошадиная сила позволяет развивать 2м/с.
  - b. температура. Каждые пройденные 10 метров повышают температуру на 5 градусов. При достижении 90 градусов, двигатель нужно охладить с помощью включения вентилятора. Каждое включение вентилятора охлаждает на 10 градусов.
3. Ваш автомобиль может поддерживать одну из двух коробок передач - автоматическую или ручную. **Необязательно:** добавить нейтральную передачу
  - a. Автоматическая имеет два состояния:
    - i. Режим езды вперед
    - ii. Режим езды назад
  - b. Ручная коробка передач имеет:
    - i. передачи от 1 до 2. При скорости от 0 до 20 используется передача №1, в противном случае передача №2.
      1. Т.к. характеристика разгона в данной задаче отсутствует - как только передана скорость больше 20 - всегда используется передача №2. Переключение между передачами внутри программы не требуется
    - ii. задняя

4. Остальные опции автомобиля остаются на ваши усмотрение, например, вы можете установить сиденье для водителя или руль.

Пример структуры:

- NIVA наследуется от класса
  - Car использует механизм trait и содержит
    - Engine
    - TransmissionAuto, TransmissionManual
      - используют общий трейт который позволяет ехать назад

Какая именно коробка передач будет использована в конечном классе выбираются в конструкторе конечного класса (в данном случае NIVA). Каждое действие выводится на экран с указанием расстояния, на котором находится автомобиль в данный момент.



