

Задание 1. Программный код.

Условия

Вы можете выбрать любой язык программирования на Ваше усмотрение.

Но перед выполнением задания, необходимо согласовать выбор с нами.

В ответ мы отправим ссылку на стандарты оформления кода, которые необходимо соблюдать.

Критерии проверки:

- программа работает без ошибок - перед отправкой решения убедитесь, что мы не сможем найти никаких несоответствий
- программный код оформлен по стандартам, которые мы Вам отправим

Для выполненного задания пришлите:

- файлы с программным кодом в .zip архиве
- скриншот программного кода в формате .png
- скриншот результатов работы программы (вывод)

Текст задания

Далее Вы найдете описание классов, которые нужно реализовать на выбранном Вами языке программирования.

Назначение классов - расчёт среднемесячных затрат на использование платных интернет сервисов. Бывают сервисы с оплатой за час использования (например, сервер Amazon EC2), или же с фиксированной оплатой в месяц (например, платный аккаунт LinkedIn).

Необходимо реализовать абстрактный класс `PaidService` со свойствами `id`, `name` и `costs` и методом `calculateAverageMonthlyCosts`.

Также необходимо реализовать два класса, которые наследуются от абстрактного класса `PaidService`:

- `FixedMonthlyCostPaidService`
- `FixedHourlyCostPaidService`

Эти дочерние классы должны отличаться только реализацией метода `calculateAverageMonthlyCosts`.

Для сервиса с почасовой оплатой, формула для расчета затрат такова:

среднемесячные затраты = $30 * 24 * costs$

Для сервиса с фиксированной оплатой за месяц:

среднемесячные затраты = `costs`

После реализации классов, необходимо написать программный код, который создаст массив со следующими экземплярами классов (атрибуты указаны в таком порядке: id, name, costs):

```
FixedHourlyCostPaidService('service1', 'Google Orkut', 11)
FixedHourlyCostPaidService('service2', 'Google Voice', 9.4)
FixedMonthlyCostPaidService('service5', 'YouTube', 8064)
FixedHourlyCostPaidService('service3', 'Mandrill', 11.2)
FixedHourlyCostPaidService('service4', 'Google Finance', 7.8)
FixedMonthlyCostPaidService('service7', 'Google Building Maker', 5347)
FixedMonthlyCostPaidService('service6', 'LinkedIn', 6863)
```

Теперь необходимо реализовать три обработки этого массива:

1. Вывести id, name и среднемесячные затраты для всех элементов массива. Вывод должен быть упорядочен по убыванию среднемесячных затрат. При совпадении суммы затрат – упорядочивать данные по названию сервиса (в алфавитном порядке от "a" до "z").
2. Вывести первые 5 значений свойства name элементов из упорядоченного списка в пункте (1).
3. Вывести последние 3 значения свойства id элементов из упорядоченного списка в пункте (1).
4. Сравнить среднемесячные затраты на сервисы YouTube и Mandrill. Вывести сравниваемые сервисы в порядке убывания затрат или два сервиса в одну строку через слеш (/), если затраты равны.

Пример вывода для экземпляров классов, указанных в задании выше:

1.

```
service3 / Mandrill / 8064
service5 / YouTube / 8064
service1 / Google Orkut / 7920
service6 / LinkedIn / 6863
service2 / Google Voice / 6768
service4 / Google Finance / 5616
service7 / Google Building Maker / 5347
```

2.

```
Mandrill
YouTube
Google Orkut
LinkedIn
Google Voice
```

3.

```
service2
service4
service7
```

4.

```
Mandrill / YouTube
```