**Task-6-3**

**Программа учета департаментов.**

**Продолжение Task-6-2, но с использованием Enum-ов.**

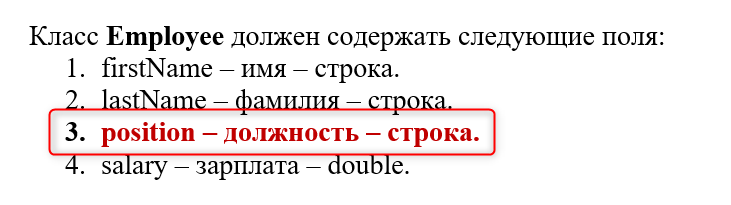
**Цели задания:**

1. Научиться пользоваться Enum-ами.

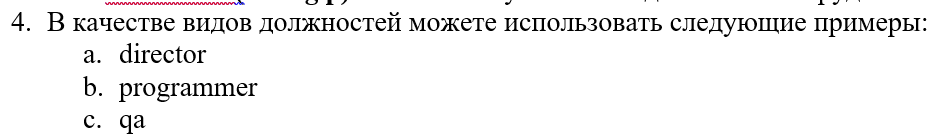
**Описание:**

Это заключительно задание по нашему виртуальному миру сотрудников и департаментов. Вообще говоря, будет еще одно, следующее, но там скорее короткая задача-головоломка, чем программирование сотрудников.

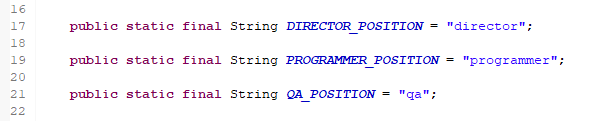
Глобальная цель текущего задания – познакомить вас с новым элементов языка – **enum**. А тактическая цель – исправить серьезную ошибку в нащей программе. Ошибку, которая вызвана вот этим требованием:



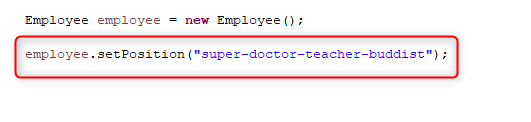
И далее по тексту идет:



Но это некорректное требование, **потому что его нельзя гарантировать**! Даже если вы введете в свою программу три строковых константы, каждая из которых будет описывать одну позицию, вот так:



Все равно, это не гарантирует вам, что какой-нибудь безответственный программист не станет использовать ваш класс Employee вот так:



И тогда все алгоритмы в вашей программе, скорее всего пойдут наперекосяк. А главное, что виноваты в этом будете вы – создатель класса **Employee**, а не он – безответственный пользователь класса **Employee**, потому что никто не может заставлять человека правильно использовать неправильные инструменты!

Ладно, все это лирика. Суть задачи – заменить поле

**private String position;**

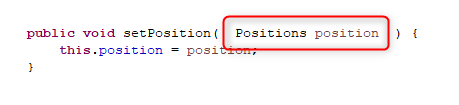
в классе Employee, полем:

**private Positions position;**

Где **Positions** – это enum с описанием трех констант, которые были указаны ранее.

Чем нам это поможет? Да именно тем, что теперь наш метод **setPosition()** будет принимать в качестве параметра не строку (любую строку, как вы могли понять ранее!), а переменную типа **enum**. А что такое enum?

**Это структура данных, которая объединяет под одним типом данных несколько строковых констант!**



Все, наша задача решена! Теперь, чтобы задать позицию сотрудника в виде **"super-doctor-teacher-buddist"**, безответственный программист должен расширить ваш enum, и добавить в него соответствующую константу. А за то, что он гадит в ваши классы, вы сможете подойти к нему и надавать ему между носа! ☺

Вот, собственно, и вся суть задачи – просто изучите enum-ы и используйте их. Но, вам ведь надоело писать один и тот же код в третий раз. Поэтому мы добавим в код еще одно требование.

Ваша программа ведь хранит список сотрудников и их директора, так? Все это объединяется классом **Deparnment**. И у каждого сотрудника есть поле **salary**.

А еще у enum-ов, кроме основной функции – описания констант, есть три приятные «плюшки»:

1. Вы можете объявить в enum-е **поля**, которые будут применяться к каждой константе отдельно.
2. Вы можете объявить в enum-е **конструкторы**, которые будут работать с каждой константой отдельно.
3. Вы можете объявить в enum-е **методы**, которые будут работать с каждой константой отдельно.
4. Вы можете объявить в enum-е **статические методы**, которые будут работать с целым enum-ом.

Поэтому в нашем enum-е мы:

1. Объявим поле **bonus**, которое будет содержать значение премии для каждой позиции.
2. Объявим конструктор, который установит размер бонуса для каждой позиции.
3. Объявим метод getBonus(), который сможет вернуть размер бонуса для каждой позиции.
4. И реализуем статический метод calculateTotalSalary(Employee e), который будет принимать на вход сотрудника, получать его зарплату, и рассчитывать полную зарплату по формуле:

totalSalary = salary + (salary \* bonus) / 100

То есть, как вы поняли, bonus – это некий процент от зарплаты сотрудника, который добавляется к его заработной плате. Тут важно, что у каждой позиции одинаковый размер бонуса, но у каждого сотрудника – разные зарплаты. Поэтому и totalSalary – у всех будет разная.

Ну а в общую бизнес логику мы добавим такой пункт:

Пользователь может ввести информацию о команде сотрудников (см. задачи 6-1 и 6-2), и указать, какой бюджет у него есть на данный момент. А программа должна рассчитать:

**Сколько месяцев эта команда сможет работать на заказчика при его бюджете.**

И получится у нас что-то вроде **«Планировщика стартапов»** - программа, которая помогает юному бизнесмену ответить на вопрос – «сколько месяцев у вас есть, чтобы выстрелить на рынке до того, как вы обанкротитесь?».

Ну, или как часто бывает, чуть проще – «Через сколько месяцев вы обанкротитесь?» ☺

Вот и готова наша задача 6-3.