# Вариант 15 конкурсных заданий *практического* этапа Конкурса

Вы встали перед выбором будущей профессии. Ваш учитель любезно предоставил вам данные по вакансиям в ведущих компаниях в формате csv файла. Вы поняли, что проанализировать данный файл вручную невозможно, после чего вам приходит идея написать функционал для более удобного понимания и представления информации.

Вы готовы приступать к задачам, но сначала необходимо **создать репозиторий** для проекта и задач, которые Вы будете выполнять. Обязательно сделайте его **PUBLIC**, а то мы не сможем проверить решение вашей последней задачи. Каждую задачу вам необходимо будет правильно оформить и залить в репозиторий. *Код, который Вы напишете необходимо задокументировать, чтобы другие программисты могли понять, что делает код и за что отвечает. Примеры документирования приведены ниже.*

**Пример документирования кода на языке С++**

/\*\*

\* Это описание функции foo

\*

\* @param str это описание аргумента str

\* @param pattern это описание аргумента pattern

\* @return это описание того, что вернет функция

\*/

int foo(std::string str, std::string& pattern)

{

...

};

**Пример документирования на языке Python**

def complex(real=0.0, imag=0.0):

"""Описание функции complex.

Описание аргументов:

real – описание аргумента

imag – описание аргумента

"""

if imag == 0.0 and real == 0.0: return complex\_zero

...

Теперь можете приступать к решению задач!

К задачам прикреплен файл *vacancy.txt,* который хранит в себе информацию о должностях, заработной плате, типе трудоустройства, компаниях, количестве сотрудников в компании.

Столбцы: *Salary(целое число), Work\_Type, Company\_Size, Role, Company.*

Разделитель *«;».*

***Задача 1.***

Вы поняли, что хотите сначала проанализировать какие максимальные зарплаты можно получить при разных должностях в разных профессиях. Для этого создайте таблицу vacancy\_new.csv в которую запишите три столбца Work\_Type, Role, Salary. При этом сама вакансия(Work\_Type) должна иметь максимальный размер зарплаты в своей категории, если в категории несколько типов работы с одинковой максимальной ЗП, добавьте в файл Role идущую первой в алфавитном порядке и с самым большим кол-вом работников в компании. После этого выведите сумму и вакансию на которую может претендовать стажер(с маленькой буквы): <Role> - <зарплата>

*В задаче запрещено использование сторонних библиотек(Pandas и др)*

*Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка (для языка Python – PEP 257). После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория.*

***Задача 2***

Для себя вы отметили, что хотели бы работать в компании с самой большой зарплатой, так как у вас много амбиций и желаний. Поэтому Вам необходимо отсортировать данные с помощью сортировки слиянием(в задаче нельзя использовать встроенные функции сортировок) по столбцу зарплата в компании в порядке убывания. После этого выведите в консоль информацию о компании с наивысшей зарплатой.

Формат вывода: “В компании <Company> есть профессия: <Role>, з/п в такой компании составит: <Salary>”

*Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка. После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория*

***Задача 3***

**Ввод: стандартный ввод**

**Вывод: стандартный вывод**

Пришло время сделать наработку для интерфейса, который будет взаимодействовать с базой данных. Для этого Вам необходимо написать консольную программу, которая будет запрашивать у Вас название профессии(Role), а на выход будет выдавать список вакансий, если ничего не найдено будет выводить: “К сожалению, ничего не удалось найти”. Программа должна всегда запрашивать название. Прекратить свою работу она сможет только после ввода “устал”.

Поиск необходимо осуществить с помощью двоичного алгоритма поиска**.**

Формат ответа на запрос пользователя: “В <компании> найдена искомая вакансия: <вакансия>. З/п составит: <Salary>”

Поиск необходимо осуществлять в файле vacancy.txt

*Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка. После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория*

***Задача 4***

Программа должна считывать данные из файла vacancy.txt и хранить их в виде многомерного списка. Каждый словарь будет содержать данные о должности, заработной плате, типе трудоустройства, компании.

Реализуйте методы/функции, которые будут определять среднюю заработную плату (после вычета налога) для определенной компании. Процентная ставка налога зависит от размера заработной платы и составляет:

Заработная плата до 45000 – налог 8%

Заработная плата 45001 - 50000 – налог 10%

Заработная плата свыше 50001 – налог 13%

*Формат списка:*

Название компании: средняя заработная плата (после вычета налога)

*Полученный список запишите в новый файл – vacancy\_average.txt. Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка. После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория.*

***Задача 5***

Программа должна считывать данные из файла vacancy.txt и хранить их в хеш-таблице. Ключом будет должность, а значением - данные о заработной плате, типе трудоустройства, компании и количестве сотрудников в компании. В случае, если ключ встречается несколько раз, необходимо разрешить коллизии посредством метода цепочек.

Выведите информацию о 3 должностях с наибольшей и 3 должностях с наименьшей зарплатами. Если встречается несколько должностей с одинаковой заработной платой, то необходимо вывести все должности.

*Не забудьте сделать комментарии к коду согласно стандартам документирования кода выбранного языка. После выполнения необходимо сделать локальные и удаленные изменения Вашего репозитория.*

***Задача 6***

Ваш код будет использоваться программистом, которого возьмут на работу, поэтому он должен быть правильно оформлен и выложен на GitHub. Весь написанный код должен быть задокументирован согласно стандартам документирования кода выбранного языка.

Также необходимо оформить README.md для Вашего репозитория. Пункты, которые должны быть описаны:

1. Название проекта

2. Описание проекта

3. Оглавление (необязательно)

4. Как установить и запустить проект

5. Как использовать проект