

# Тестовое задание для программиста C# (BackEnd)

## Введение

Вам нужно выполнить три задания на разные темы: разработка на SQL, разработка интерфейсов и разработка на .Net.

Вы можете выбрать любые варианты в заданиях 1 и 2. Задание 3 имеет один вариант.

**Все три задания выложить на [GitHub](#)**

Удачи!

## Задание 1 «Разработка на SQL»

Выберите и реализуйте один из вариантов задания.

### Вариант 1. Проектирование баз данных

#### Проверяемые навыки:

- навык проектирования;
- способность к анализу;
- логическое мышление;
- предметное мышление.

#### Задание:

Разработать структуру базы данных (перечень таблиц с описанием полей, ключей, связей между таблицами) для учета проводимых курсов обучения.

#### Исходные данные:

Требования к базе данных:

1. Один и тот же курс может проводиться несколько раз.
2. Один и тот же обучаемый может обучаться на разных курсах обучения.
3. Один и тот же обучаемый не может проходить один и тот же курс дважды.
4. Количество мест на курсах обучения определяется при добавлении курса в расписание.

### Вариант 2. Написание запроса к базе данных

#### Проверяемые навыки:

- навык написание запросов;
- логическое мышление – способность составлять алгоритмы.

#### Задание:

1. Написать SQL-запрос, который возвращает объем продаж в количественном выражении в разрезе сотрудников за период с 01.10.2013 по 07.10.2013:
  - Фамилия и имя сотрудника;
  - Объем продаж сотрудника.

Данные отсортировать по фамилии и имени сотрудника.

2. На основании созданного в первом задании запроса написать SQL-запрос, который возвращает процент объема продаж в разрезе сотрудников и продукции за период с 01.10.2013 по 07.10.2013:

- Наименование продукции;
- Фамилия и имя сотрудника;
- Процент продаж сотрудником данного вида продукции (продажи сотрудника данной продукции/общее число продаж данной продукции).

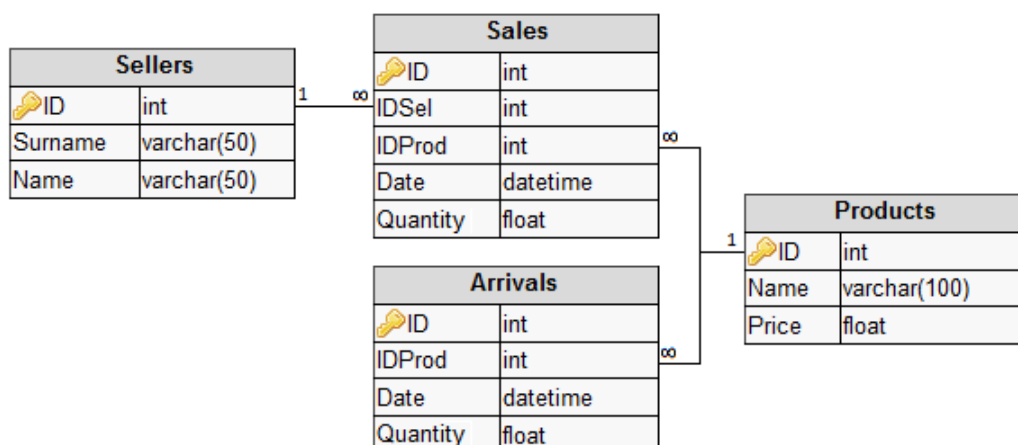
В выборку должна попадать продукция, поступившая за период с 07.09.2013 по 07.10.2013.

Данные отсортировать по наименованию продукции, фамилии и имени сотрудника.

#### Исходные данные:

Даны таблицы с данными:

1. Sellers – продавцы продукции:
  - ID – идентификатор продавца;
  - Surname – фамилия продавца;
  - Name – имя продавца.
2. Products – продукция:
  - ID – идентификатор продукта;
  - Name – наименование продукта;
  - Price – цена за единицу продукта.
3. Sales – продажи продукции:
  - ID – идентификатор;
  - IDSel – поле ID из таблицы Sellers;
  - IDProd – поле ID из таблицы Products;
  - Date – дата продажи продукции;
  - Quantity – объем проданной продукции.
4. Arrivals – поступление продукции:
  - ID – идентификатор;
  - IDProd – поле ID из таблицы Products;
  - Date – дата поступления продукции;
  - Quantity – объем поступившей продукции.



## Задание 2 «Разработка интерфейса»

Выберите и реализуйте один из вариантов задания.

#### Проверяемые навыки:

- навык разработки интерфейсов;
- понимание нужд заказчика;
- логическое мышление;

- предметное мышление.

### **Задание:**

Для desktop-приложения разработать визуальную форму справочника для просмотра и ввода данных пользователем. Возможные варианты построения прототипов интерфейсов:

- выполнить задание от руки и выслать изображения;
- on-line редактор [Mockup Builder](#);
- редактор [Balsamiq Mockups](#).

Итоговый прототип интерфейса сохранить в одном из форматов: JPEG, PNG или PDF.

## **Вариант 1. Разработка интерфейса карточки сотрудника**

### **Исходные данные:**

В информационной системе предприятия для каждого сотрудника определен набор характеристик, обязательные для заполнения характеристики помечены символом «\*».

Общие сведения:

- \*Фамилия;
- \*Имя;
- Отчество;
- \*Пол;
- \*Дата рождения;
- \*Серия удостоверения личности;
- \*Номер удостоверения личности;
- \*Кем выдано;
- \*Когда выдано;
- ИНН;
- \*Индекс по месту регистрации;
- \*Регион (адрес регистрации);
- \*Населенный пункт (адрес регистрации);
- \*Улица (адрес регистрации);
- \*Дом (адрес регистрации);
- Корпус (адрес регистрации);
- Квартира (адрес регистрации);
- Индекс по месту проживания;
- Регион (адрес проживания);
- Населенный пункт (адрес проживания);
- Улица (адрес проживания);
- Дом (адрес проживания);
- Корпус (адрес проживания);
- Квартира (адрес проживания).

Сведения о предыдущих местах работы:

- \*Место работы;
- \*Должность;
- \*Стаж.

Состав семьи:

- \*ФИО;
- \*Родственник;
- \*Дата рождения.

Необходимо разработать визуальную форму для просмотра и редактирования характеристик сотрудников.

Дополнительное требование: предусмотреть возможность указывать любое количество членов семьи и предыдущих мест работы.

## Вариант 2. Разработка интерфейса работы с совещаниями

### Исходные данные:

В информационной системе необходимо разработать карточку совещания.

Совещание – объект системы, который позволяет фиксировать информацию о времени, месте проведения, участниках совещания и его повестке. Участниками совещания могут быть как сотрудники компании, организующей совещание, так и внешние заинтересованные лица.

Совещания организуются ответственными за их проведение заранее и согласуются с предполагаемыми участниками. Если участники не могут принять участие в совещании в предложенное организатором время, то совещание может быть перенесено.

После проведения совещания ответственный создает протокол совещания, в котором расписаны решения, принятые в ходе совещания. Решения содержат информацию о том, кто и что должен сделать по итогам совещания и к какому сроку.

Далее ответственный согласует протокол с теми, кто принимал участие в совещании. После согласования протокола ответственный за совещание организует начало работ по протоколу.

## Задание 3 «Разработка на .Net»

### Разработка приложения

#### Проверяемые навыки:

- навыки проектирований объектной модели;
- оптимальность кода;
- простота и понятность кода;
- понимание того, что такое хороший код.

#### Задание:

1. Разработать консольное приложение на C# .Net по описанию.
2. Реализовывать долгосрочное хранение информации о встречах необязательно, достаточно хранить встречи в памяти.
3. Для приложения написать план тестирования.

Нужно присылать исходные файлы и проект (solution) в Visual Studio!

#### Исходные данные:

Необходимо разработать приложение для управления личными встречами.

Встреча – событие, которое планируется заранее, для которой всегда известно время начала и примерное время окончания.

Данные о встречах могут быть добавлены, изменены и удалены. Встречи всегда планируются только на будущее время. При этом встречи не должны пересекаться друг с другом.

Также для встречи может быть настроено время, за которое нужно уведомить пользователя о предстоящей встрече. Время напоминания также может быть изменено после создания встречи. При наступлении времени напоминания приложение информирует пользователя о предстоящей встрече и времени ее начала.

Пользователь может посмотреть расписание своих встреч на любой день, в том числе и прошедший.

Помимо просмотра он может с помощью приложения экспортировать расписание встреч за выбранный день в текстовый файл.