

**ЗАКЛЮЧИТЕЛЬНЫЙ ЭТАП****ЭКЗАМЕНАЦИОННОЕ ЗАДАНИЕ (ВАРИАНТ 1)****по компетенции****«РАЗРАБОТКА НАСТОЛЬНЫХ ПРИЛОЖЕНИЙ»****(НАПРАВЛЕНИЕ «ПРОГРАММИРОВАНИЕ»)****ЗАДАНИЕ:**

Вам предстоит написать настольное приложение. В данном задании представлены общие требования к программе, интерфейсу и основная логика.

ОБЩИЕ ТРЕБОВАНИЯ:

Написать программу согласно представленному заданию. В программе должны быть реализованы указанные в задании особенности программы, макет интерфейса и требование к коду.

При запуске программы открывается окно авторизации. К заданию прилагается `csv` файл, содержащий список пользователей, их данные и роль (работник, администратор и директор). После авторизации, в зависимости от роли, открывается окно с соответствующим функционалом. Работник может выполнять свои должностные обязанности. Администратор имеет возможность редактировать, удалять и изменять данные других пользователей. Директор имеет доступ к общей статистике по работе пользователей.

В ходе оценки проверяется работоспособность модулей программы, соблюдение стиля кода, его читаемость и наличие комментариев.



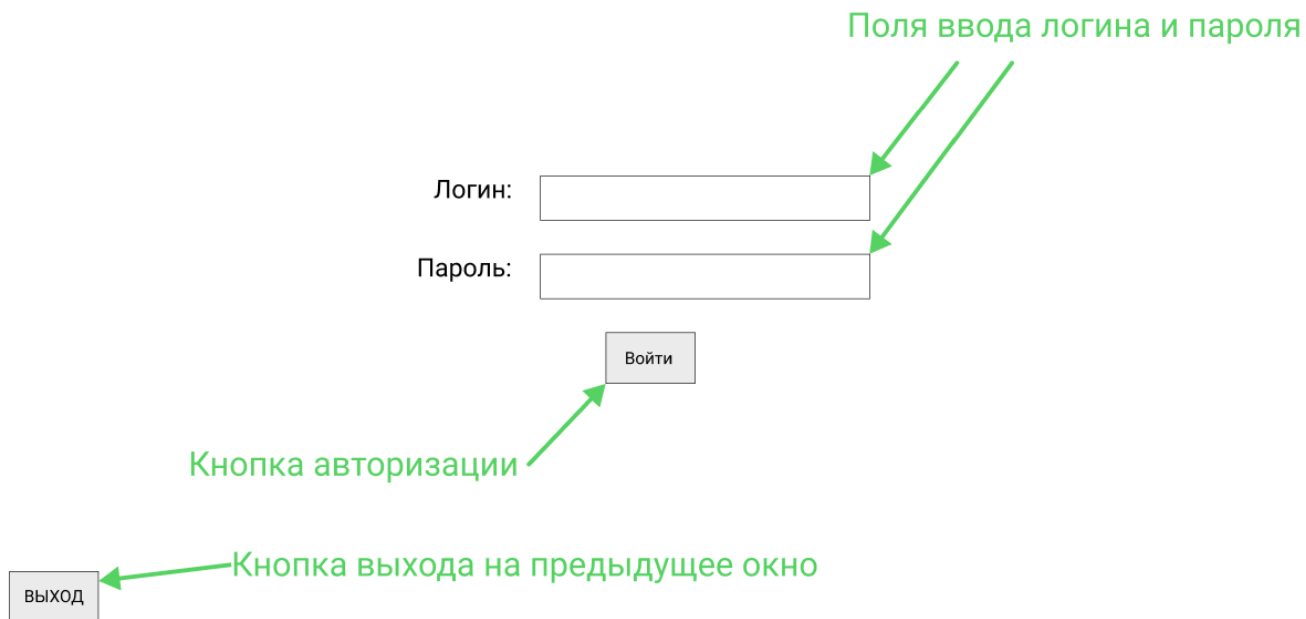
ОПИСАНИЕ ИНТЕРФЕЙСА

- Окно авторизации
 - Окно содержит поля ввода: Логин и Пароль.
 - Окно содержит кнопку для подтверждения ввода.
- Окно пользователя
 - Окно содержит надпись, приветствующее пользователя.
 - Окно содержит рабочую кнопку функционала.
 - Окно содержит надпись, отражающую результативность работника.
- Окно директора
 - Окно содержит таблицу с результативностью работников.
 - Окно содержит диаграмму, наглядно отображающую текущую результативность работников.
- Окно администратора
 - Окно содержит таблицу с данными всех сотрудников.
 - Окно содержит кнопку изменения, добавления и удаления сотрудников.
- Окно добавления/изменения данных пользователя
 - Окно содержит поля ввода данных сотрудника.
 - Окно содержит кнопку подтверждения изменения данных\добавления нового сотрудника.



МАКЕТ:

- Окно авторизации





- Окно работника

Текст с приветствием сотрудника, имя берётся из файла

Добро пожаловать, (Имя сотрудника)!

НАЖАТЬ

Ваш результат: XXX

Текущий результат
работника

Рабочая кнопка

НАЗАД

Кнопка выхода на предыдущее окно



- Окно директора

Номер сотрудника	Имя сотрудника	Результативность

Отображение
результатов сотрудников

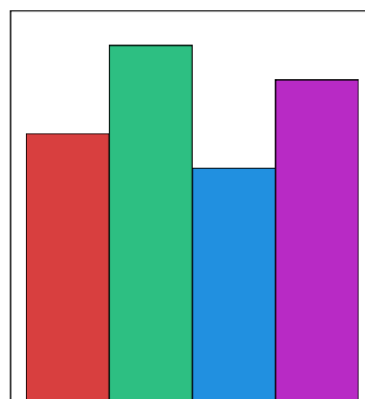


Диаграмма результатов
работников

НАЗАД

← Кнопка выхода на предыдущее окно



- Окно администратора

Номер сотрудника	Логин	Пароль	Роль	Имя сотрудника

Добавить Изменить Удалить

Кнопки управления списком сотрудников

Отображение
списка сотрудников

НАЗАД

Кнопка выхода на предыдущее окно





- Окно добавления/изменения данных пользователя

Логин:

Пароль:

Имя:

Фамилия:

Роль:

Поля ввода

ПРИНЯТЬ

НАЗАД ← Кнопка выхода на предыдущее окно



ЛОГИКА РАБОТЫ ПРОГРАММЫ:

При запуске программы открывается окно авторизации. Пользователь вводит логин и пароль. Если введенные данные совпадают с логином и паролем учетных записей, представленных в *csv* файле, происходит переход в одну из 3-х форм в зависимости от роли пользователя.

Форма работника.

При входе на форму работника отображается надпись с приветствием пользователя, в котором указаны имя и фамилия. На форме расположена функциональная кнопка, нажатие которой отображает должностную инструкцию работника. Справа от кнопки расположена надпись, которая демонстрирует результативность работника (количество нажатий на данный момент). Количество нажатий хранится в *csv* файле для каждого работника, а также обновляется после каждого нажатия, как на форме, так и в файле.

Форма администратора.

При входе на форму администратора отображается таблица с данными всех пользователей, кроме самого администратора. Выбрав одного из пользователей, администратор может нажать кнопку “удалить” для удаления пользователя из списка. Также администратор имеет возможность изменить данные выбранного пользователя с помощью кнопки “изменить”. Администратор может добавить нового пользователя, нажав кнопку “добавить”.

Все изменения отображаются в таблице на форме администратора, а также в *csv* файле.

Форма добавления/изменения пользователя.

При нажатии на кнопку “добавить” в окне администратора откроется окно добавления нового пользователя. При условии, что ВСЕ поля заполнены, администратор имеет возможность нажать кнопку “принять”. В случае если на форме администратора была нажата кнопка “изменить”, то данная форма открывается с уже заполненными значениями полей для выбранного пользователя. При нажатии кнопки “принять” все изменения должны отображаться в таблице на форме администратора и в *csv* файле.

Форма директора.

При входе на форму директора отображается таблица с результативностью работников предприятия. Справа от таблицы отображается диаграмма, наглядно демонстрирующая значения из таблицы. Вы имеете возможность выбрать вид диаграммы (круговая, столбцы, график и т.д.). В диаграмме должны быть указаны имя и фамилия работников, а также достоверно отображаться их показатели из таблицы.



ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ИНТЕРФЕЙСА

Общие требования:

- Каждое окно должно иметь кнопки навигации, возвращающие пользователя программы на предыдущее окно, либо закрывающее программу, в случае окна авторизации.
- Размеры элементов интерфейса не должны быть меньше 50 px в ширину и высоту.
- Шрифт: Arial, размер не меньше 12 и не больше 20, цвет - черный (0, 0, 0).
- Фон всех форм должен быть серым (120, 120, 120).
- Фон всех полей ввода и кнопок должен быть белым (255, 255, 255).

ТРЕБОВАНИЯ К ПРОЕКТУ:

Участник вправе выбрать язык программирования и среду разработки из предлагаемого списка:

Языки программирования: Java, C++, C#, Python, Object Pascal;

Среды разработки: IntelliJ IDEA, Processing3, Visual Studio, PyCharm, Pascal ABC.Net, PythonIDE.

В программе должны быть выполнены все указанные в задании механики и элементы, предусмотрены проверки для предотвращения ошибок работы программы (пример: попытка использовать неинициализированный объект, наличие пустых полей при добавлении/изменении пользователя и др.). В случае возникновения ошибок должна быть реализована обратная связь (сообщение с указанием произошедшей ошибки).

Имена переменных и элементов должны иметь говорящие названия на английском языке (соблюдая стиль кода, пример: camelCase, snake_case). Для написанных подпрограмм и переменных следует оставлять поясняющие комментарии.

Шаблоны комментариев:

Все комментарии – строчные и оформлены по следующему макету:

- Данный метод принимает следующие значения: *имя параметра – его описание* [*; следующий параметр*].
- Выполняет следующие действия: описание функционала.
- В качестве результата возвращает следующее значение: описание результата.

Программа должна быть оптимизирована и рассчитана на длительные сеансы использования.