Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ.**

Специальность: 09.02.07 Информационные системы и программирование

Квалификация: Программист

ПРАКТИЧЕСКАЯ РАБОТА №3

ПО «ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ПО»

Листов: 4

|  |  |
| --- | --- |
| Выполнил студент  Группы: П50-7-20  М. М. Антонова | Проверил преподаватель  М. Н. Гацкан  «\_\_\_\_» \_\_\_\_\_\_\_2023 года |

Москва 2023

# ПРАКТИЧЕСКАЯ РАБОТА №3

Цель работы: составить алгоритм и создать программу подсчета суммы цифр всех дат для каждого месяца всех дней в году. (Кимчи Рамен, рандомом выпадает паук)

1. Вначале прописываем используемые вспомогательные библиотеки

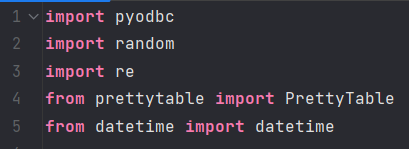


Рисунок 1. Импорт вспомогательных элементов

1. Далее подключаем базу данных (SQL MS Server)

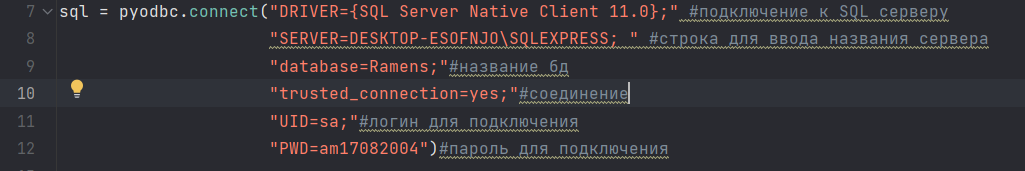


Рисунок 2. Подключение бд

1. Следующим шагом вводим переменные

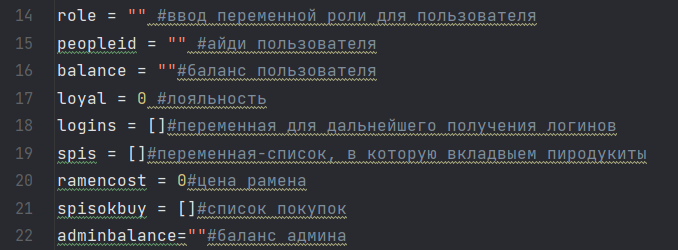


Рисунок 3. Переменные в проекте

1. Для красивого вывода таблиц используем класс и библиотеку



Рисунок 4. Класс с библиотекой

1. Дальше большая часть функций имеет одну и туже схему (обращение к бд, запрос в бд, обновление бд, закрытие обращения), но также присутствует перебор данных)

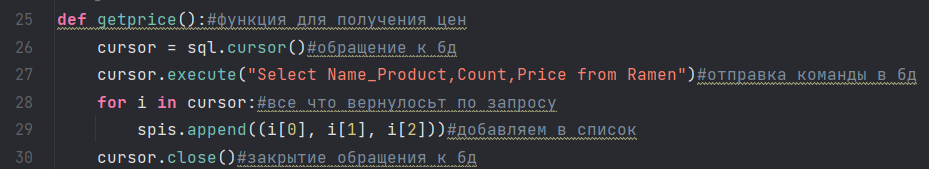


Рисунок 5. В данном случае мы обращаемся в бд для проверки цен

1. Перейдем к авторизации

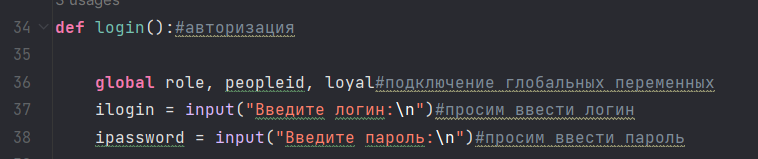


Рисунок 6. Первым делом подключим глобальные переменные, которые позже будут также использоваться в других функциях. Дальше просим пользователя ввести логин и пароль



Рисунок 7. Обращаемся к бд и отправляем запрос



Рисунок 8. Запускаем цикл переборки данных



Рисунок 9. Прописываем варианты событий

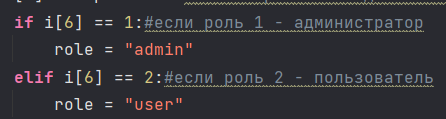


Рисунок 10. Во-втором варианте происходит инициализация пользователя как логина или клиента



Рисунок 11. Получаем карту лояльности



Рисунок 12. Выводим надпись под какой ролью авторизован пользователь

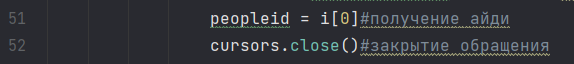


Рисунок 13. Получаем айди пользователя и закрываем ранее открытое обращение к бд

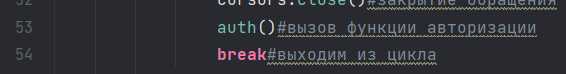


Рисунок 14. Вызываем функцию авторизации и выходим из цикла

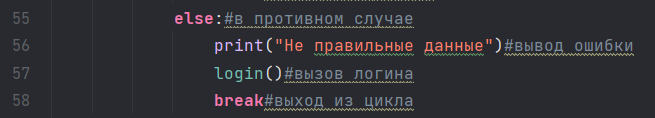


Рисунок 15. Последним в этой функции будет предложено альтернативные варианты (первым мы выведем ошибку о неправильно введенных данных и вызовем авторизацию заново)

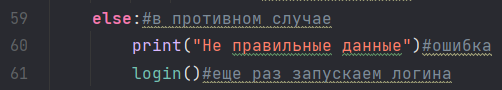


Рисунок 16. Вторым вариантом будет также вывод ошибки и повторное открытие авторизации

1. Следующей стартует функция регистрации



Рисунок 17. Регистрация



Рисунок 18. Снова обращаемся к глобальной переменной и получаем список всех логинов

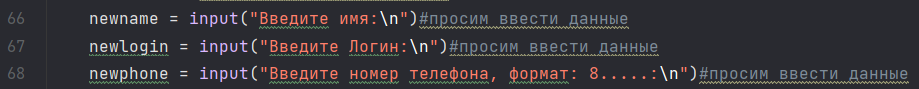


Рисунок 19. Просим заполнить такие поля, как: логин, пароль и номер телефона

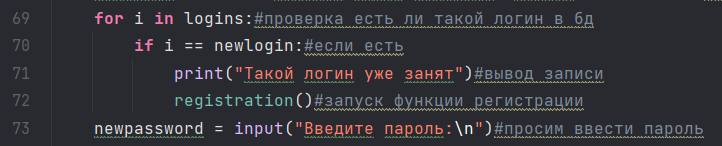


Рисунок 20. Проверяем через цикл есть ли такие данные в базе, и если данные есть, то заново запускаем регистрацию и просим ввести пароль

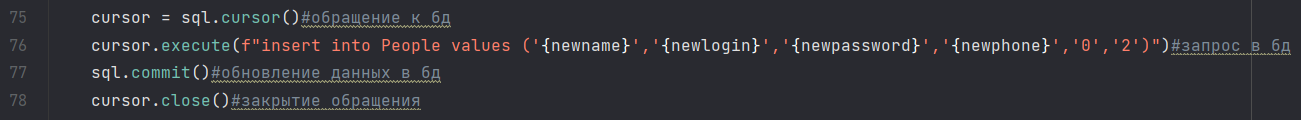


Рисунок 21. Отправляем обращение к бд, отправляем запрос в бд, обновляем данные и закрываем обращение

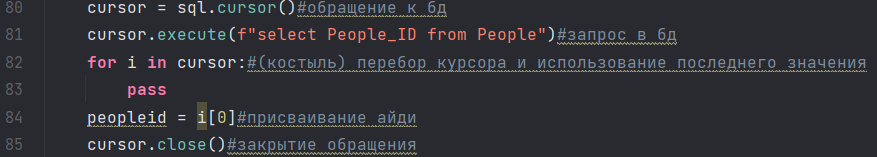


Рисунок 22. Отправляем обращение к бд, отправляем запрос в бд, обновляем данные, получаем айди нового пользователя и закрываем обращение



Рисунок 23. Снова обращаемся в бд



Рисунок 24. Генерируем баланс пользователя от начальной стоимости рамена и пользователь получаем свой баланс

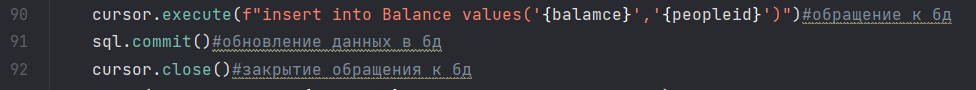


Рисунок 25. Отправляем запрос в бд, обновляем данные о балансе и закрываем обращение



Рисунок 26. Поздравление пользователя с регистрацией

1. Следующей задачей будет реализация получения стоимости рамена



Рисунок 27. Получение стоимости



Рисунок 28. Используем глобальную переменную и обращаемся в бд



Рисунок 29. Отправляем запрос в бд



Рисунок 30. Снова через цикл делаем перебор и складываем стоимость товаров



Рисунок 31. Закрываем обращение и возвращаем общую стоимость

1. Получение логина пользователя



Рисунок 32. Получение логина

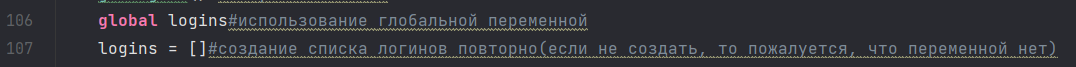


Рисунок 33. Используем глобальную переменную и создаем список всех логинов



Рисунок 34. Обращаемся в бд и отправляем запрос



Рисунок 35. Получаем ответ через цикл



Рисунок 36. Если логина нет, добавляем его в список



Рисунок 37. Закрываем обращение

1. Теперь получаем баланс именно авторизированного пользователя



Рисунок 38. Баланс авторизированного пользователя



Рисунок 39. Обращаемся к глобальной переменной и к бд



Рисунок 40. Оформляем запрос

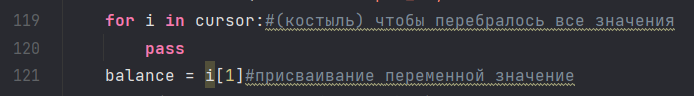


Рисунок 41. Через цикл с перебором присваиваем значение переменной



Рисунок 42. Выводим баланс и закрываем обращение к бд

1. Получаем баланс админа



Рисунок 43. Баланс админа

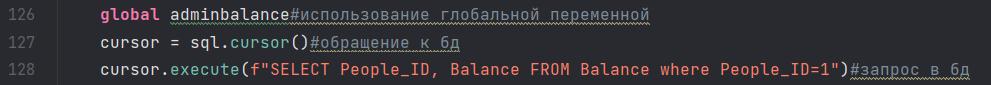


Рисунок 44. Снова глобальная переменная, обращение к бд и запрос

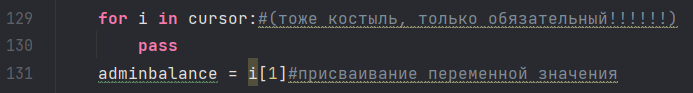


Рисунок 45. Опять цикл с перебором и присвоение значений



Рисунок 46. Закрываем обращение

1. Создаем админку



Рисунок 47. Админка



Рисунок 48. И вперед по новой, глобальная переменная и обращение к бд



Рисунок 49. Выводим баланс и запрашиваем пополнение склада



Рисунок 50. Создаем переменную



Рисунок 51. Если ответ да