МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ «ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ ЕВФРОСИНИИ ПОЛОЦКОЙ»

Факультет информационных технологий

Кафедра технологий программирования

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине «О-ОТПиСП»

Тема: «Создание классов и объектов»

|  |  |
| --- | --- |
| Выполнил: | студент группы 23-ИТ-1  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.Ю. Страпко |
| Проверил: | ассистент кафедры ТП  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.В. Васильева |

|  |
| --- |
|  |

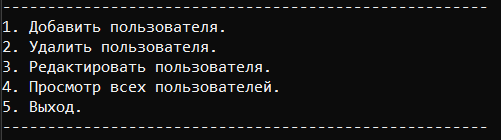
**Цель работы:**Научиться создавать классы и объекты этих классов, вызывать методы и обращаться к данным через методы этих объектов, передавать объекты других классов через методы, понимать влияние модификаторов доступа на видимость данных и методов.

1. **Практическая часть**

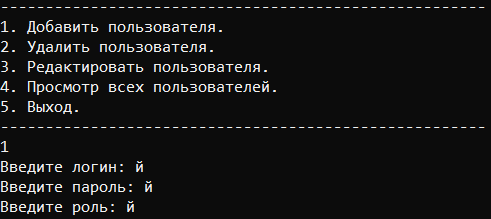
Листинг 1.1 – Файл Program.cs

1. using System.Reflection.Metadata;
2. class User
3. {
4. private string login;
5. private string password;
6. private string role;
7. public User(string login, string password, string role)
8. {
9. this.login = login;
10. this.password = password;
11. this.role = role;
12. }
13. public void ShowInfo()
14. {
15. Console.WriteLine($"Логин: {login} Пароль: {password} Роль: {role}");
16. }
17. public string GetLogin() => login;
18. public string GetRole() => role;
19. public string GetPassword() => password;
20. public void ChangePassword(string newPassword)
21. {
22. password = newPassword;
23. }
24. public void ChangeRole(string newRole)
25. {
26. role = newRole;
27. }
28. }
29. class Application
30. {
31. private string name;
32. private int volume;
33. private List<User> users;
34. public Application(string name, int volume)
35. {
36. this.name = name;
37. this.volume = volume;
38. users = new List<User>();
39. }
40. public void Add(User user)
41. {
42. users.Add(user);
43. }
44. public void Remove(string login)
45. {
46. users.RemoveAll(u => u.GetLogin() == login);
47. }
48. public void RedactUser(string login, string newPassword, string newRole)
49. {
50. foreach(var user in users)
51. {
52. if(user.GetLogin() == login)
53. {
54. user.ChangePassword(newPassword);
55. user.ChangeRole(newRole);
56. return;
57. }
58. }
59. }
60. public void ShowUser()
61. {
62. foreach (var user in users)
63. {
64. user.ShowInfo();
65. }
66. }
67. }
68. class Program
69. {
70. static void Main()
71. {
72. Application application = new Application("Управление пользователями", 2);
73. while (true)
74. {
75. Console.WriteLine("-------------------------------");
76. Console.WriteLine("1. Добавить пользователя.");
77. Console.WriteLine("2. Удалить пользователя.");
78. Console.WriteLine("3. Редактировать пользователя.");
79. Console.WriteLine("4. Просмотр всех пользователей.");
80. Console.WriteLine("5. Выход.");
81. Console.WriteLine("-------------------------------");
82. string choise = Console.ReadLine();
83. switch (choise)
84. {
85. case "1":
86. Console.Write("Введите логин: ");
87. string login = Console.ReadLine();
88. if (string.IsNullOrWhiteSpace(login))
89. {
90. Console.WriteLine("Некорректный ввод.");
91. break;
92. }
93. Console.Write("Введите пароль: ");
94. string password = Console.ReadLine();
95. if (string.IsNullOrWhiteSpace(password))
96. {
97. Console.WriteLine("Некорректный ввод.");
98. break;
99. }
100. Console.Write("Введите роль: ");
101. string role = Console.ReadLine();
102. if (string.IsNullOrWhiteSpace(role))
103. {
104. Console.WriteLine("Некорректный ввод.");
105. break;
106. }
107. application.Add(newUser(login,password, role));
108. break;
109. case "2":
110. Console.Write("Введите логин для удаления: ");
111. login = Console.ReadLine();
112. if (string.IsNullOrWhiteSpace(login))
113. {
114. Console.WriteLine("Некорректный ввод.");
115. break;
116. }
117. application.Remove(login);
118. break;
119. case "3":
120. Console.Write("Введите логин для редактирования: ");
121. login = Console.ReadLine();
122. if (string.IsNullOrWhiteSpace(login))
123. {
124. Console.WriteLine("Некорректный ввод.");
125. break;
126. }
127. Console.Write("Введите новый пароль: ");
128. password = Console.ReadLine();
129. if (string.IsNullOrWhiteSpace(password))
130. {
131. Console.WriteLine("Некорректный ввод.");
132. break;
133. }
134. Console.Write("Введите новую роль: ");
135. role = Console.ReadLine();
136. if (string.IsNullOrWhiteSpace(role))
137. {
138. Console.WriteLine("Некорректный ввод.");
139. break;
140. }
141. application.RedactUser(login, password, role);
142. break;
143. case "4":
144. application.ShowUser();
145. break;
146. case "5":
147. return;
148. default:
149. Console.WriteLine("Некорректный ввод.");
150. break;
151. }}}}

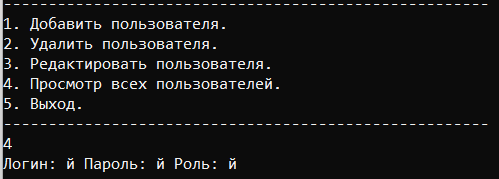
**2.Скриншоты результата**

****

**Рисунок 2.1 – Меню управления**

****

**Рисунок 2.2 – Добавление пользователя**

****

**Рисунок 2.3 – Просмотр всех пользователей**

**3. Пояснение алгоритма**

Данный код представляет собой консольное приложение на C#, которое управляет пользователями. Вот краткое описание его работы:

Класс User:

Содержит три приватные поля: login, password, role. Конструктор принимает эти параметры и инициализирует поля. Методы: ShowInfo(): выводит информацию о пользователе. Геттеры для доступа к полям: GetLogin(), GetRole(), GetPassword(). ChangePassword(string newPassword): изменяет пароль. ChangeRole(string newRole): изменяет роль.

Класс Application:

Хранит название приложения, объем и список пользователей. Методы: Add(User user): добавляет пользователя в список. Remove(string login): удаляет пользователя по логину. RedactUser(string login, string newPassword, string newRole): редактирует информацию о пользователе. ShowUser(): выводит информацию о всех пользователях.

Класс Program:

Содержит метод Main(), который запускает главный цикл приложения. В цикле выводится меню с выбором действий (добавить, удалить, редактировать пользователей, просмотреть всех пользователей или выйти). В зависимости от выбора пользователя выполняются соответствующие действия, такие как ввод данных и вызов методов класса Application.

**Вывод:** в ходе данной лабораторной работы были изучено создание классов и объектов этих классов, вызов методов и обращение к данным через методы этих объектов, передача объектов других классов через методы, понимание влияния модификаторов доступа на видимость данных и методов.