

Образец оформления титульного листа домашнего задания

**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

---

**Отчет по лабораторным работам  
Разработка WEB приложений**

Студент: Левандовский К.В.

---

Группа: СМ10-61Б

---

Преподаватель: Гапанюк Ю.Е.

## Лабораторная работа № 1

В ходе этой лабораторной работы были изучены основы языка программирования Python, в том числе типы данных, операторы, условные конструкции, циклы, функции. По итогам лабораторной работы было предложено написать программу для нахождения корней биквадратного уравнения. Листинг этой программы представлен далее:

```
1. import sys
2. import math
3.
4. def get_coef(index, prompt):
5.     '''
6.     Читаем коэффициент из командной строки или вводим с клавиатуры
7.     Args:
8.         index (int): Номер параметра в командной строке
9.         prompt (str): Приглашение для ввода коэффициента
10.    Returns:
11.        float: Коэффициент квадратного уравнения
12.    '''
13.    try:
14.        # Пробуем прочитать коэффициент из командной строки
15.        coef_str = sys.argv[index]
16.    except:
17.        # Вводим с клавиатуры
18.        print(prompt)
19.        coef_str = input()
20.        # Переводим строку в действительное число
21.        try:
22.            coef = float(coef_str)
23.        except ValueError:
24.            # Некорректный ввод, вводим коэффициент повторно
25.            print("Некорректный коэффициент. Попробуйте снова.")
26.            coef = get_coef(index, prompt)
27.        return coef
28.
29.
30. def get_roots(a, b, c):
31.     '''
32.     Вычисление корней квадратного уравнения
33.     Args:
34.         a (float): коэффициент A
35.         b (float): коэффициент B
36.         c (float): коэффициент C
37.     Returns:
```

```

38. ...         list[float]: Список корней
39. ...         '''
40. ...     result = []
41. ...     D = b*b - 4*a*c
42. ...     if D == 0.0:
43. ...         root = -b / (2.0*a)
44. ...         result.append(root)
45. ...     elif D > 0.0:
46. ...         sqD = math.sqrt(D)
47. ...         root1 = (-b + sqD) / (2.0*a)
48. ...         root2 = (-b - sqD) / (2.0*a)
49. ...         result.append(root1)
50. ...         result.append(root2)
51. ...     return result
52. ...
53. ... def main():
54. ...     '''
55. ...     Основная функция
56. ...     '''
57. ...     a = get_coef(1, 'Введите коэффициент A:')
58. ...     b = get_coef(2, 'Введите коэффициент B:')
59. ...     c = get_coef(3, 'Введите коэффициент C:')
60. ...     # Вычисление корней
61. ...     roots = get_roots(a,b,c)
62. ...     # Вывод корней
63. ...     len_roots = len(roots)
64. ...     if len_roots == 0:
65. ...         print('Нет корней')
66. ...     elif len_roots == 1:
67. ...         print('Один корень: {}'.format(roots[0]))
68. ...     elif len_roots == 2:
69. ...         print('Два корня: {} и {}'.format(roots[0], roots[1]))
70. ...
71. ...
72. ... # Если сценарий запущен из командной строки
73. ... if __name__ == "__main__":
74. ...     main()

```

## Лабораторная работа № 2

В ходе второй лабораторной работы были изучены основы использования фреймворка Flask в языке программирования Python. Разработано простое веб-приложение, демонстрирующее принцип работы Flask и возможность создания веб-серверов на основе данного фреймворка.

Листинг программного кода, который был написан на лабораторной работе представлен далее:

```
1. from flask import Flask, render_template
2.
3. app = Flask(__name__)
4.
5. class Book:
6.     def __init__(self, title, author, genre):
7.         self.title = title
8.         self.author = author
9.         self.genre = genre
10.
11. class Bookstore:
12.     def __init__(self, name, books):
13.         self.name = name
14.         self.books = books
15.
16. # Создание нескольких книг
17. book1 = Book("The Great Gatsby", "F. Scott Fitzgerald", "Fiction")
18. book2 = Book("To Kill a Mockingbird", "Harper Lee", "Fiction")
19. book3 = Book("Pride and Prejudice", "Jane Austen", "Classic")
20. book4 = Book("1984", "George Orwell", "Dystopian")
21. book5 = Book("The Hobbit", "J.R.R. Tolkien", "Fantasy")
22.
23. # Создание нескольких книжных магазинов
24. bookstore1 = Bookstore("Bookstore A", [book1, book2])
25. bookstore2 = Bookstore("Bookstore B", [book3, book4, book5])
26.
27. @app.route("/")
28. def home():
29.     return render_template('index.html', bookstores=[bookstore1, bookstore2])
30.
31. @app.route("/book/<title>")
```

```
32. def book_details(title):
33.     # Поиск книги по названию
34.     for bookstore in [bookstore1, bookstore2]:
35.         for book in bookstore.books:
36.             if book.title == title:
37.                 return render_template('book.html', book=book)
38.     return "Книга не найдена."
39.
40. if __name__ == "__main__":
41.     app.run()
```

### Лабораторная работа № 3

Третья лабораторная работа по Python была посвящена освоению работы с веб-страницами. Были изучены основные принципы разметки HTML, стилизации элементов страницы с помощью CSS и создания интерактивных элементов с помощью JavaScript. Применены полученные знания при разработке простой веб-страницы, а также написаны и использованы файлы index.html.txt, script.js и style.css для продвинутой работы с web-страницами.

Листинг кода из файла Lab\_3.html:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <meta charset="UTF-8">
5.   <title>История мема Рикроллинг</title>
6.   <style>
7.     body {
8.       font-family: Arial, sans-serif;
9.       font-size: 16px;
10.      line-height: 1.5;
11.      margin: 0;
12.      padding: 0;
13.    }
14.    header {
15.      background-color: #333;
16.      color: #fff;
17.      padding: 20px;
18.      text-align: center;
19.    }
20.    h1 {
21.      margin: 0;
22.      font-size: 36px;
23.    }
24.    main {
25.      max-width: 800px;
26.      margin: 20px auto;
27.      padding: 0 20px;
28.    }
29.    p {
```

```
30.     margin-bottom: 10px;
31. }
32. ul {
33.     margin: 0;
34.     padding-left: 20px;
35. }
36. img {
37.     display: block;
38.     margin: 20px auto;
39.     max-width: 100%;
40.     height: auto;
41. }
42. .meme {
43.     border: 1px solid #ccc;
44.     border-radius: 5px;
45.     margin-bottom: 20px;
46.     overflow: hidden;
47. }
48. .meme img {
49.     display: block;
50.     max-width: 100%;
51.     height: auto;
52. }
53. .meme h2 {
54.     font-size: 24px;
55.     margin: 10px;
56.     text-align: center;
57. }
58. .meme p {
59.     margin: 10px;
60.     text-align: center;
61.     font-style: italic;
62.     color: #888;
63. }
64. .btn {
65.     display: block;
66.     background-color: #333;
67.     color: #fff;
68.     padding: 10px 20px;
69.     border-radius: 5px;
```

```
70.     text-decoration: none;
71.     margin: 20px auto;
72.     width: 200px;
73.     text-align: center;
74. }
75. .btn:hover {
76.     background-color: #666;
77. }
78. </style>
79. </head>
80. <body>
81.     <header>
82.         <h1>История мема Рикроллинг</h1>
83.     </header>
84.     <main>
85.         <section>
86.             <h2>Что такое Рикроллинг?</h2>
87.             <p>Рикроллинг - это мем, который начал распространяться в 2007 году. Он заключается в том,
            что пользователи отправляют друг другу ссылку на видео "Never Gonna Give You Up" от Рика
            Эстли.</p>
88.         </section>
89.         <section>
90.             <h2>Как начался мем Рикроллинг?</h2>
91.             <p>Мем начал распространяться после того, как пользователи сайта 4chan начали отправлять
            друг другу ссылки на видео "Never Gonna Give You Up" вместо обещанных ссылок на другие
            контенты. Это стало известно как "ретроспективная шутка", когда кто-то делает что-то
            неожиданное или дурацкое, чтобы вызвать смех.</p>
92.         <section>
93.             <h2>Как Рикроллинг стал популярным мемом?</h2>
94.             <p>Мем Рикроллинг стал очень популярным в интернете в 2008 году, когда его использовали в
            шутках и приколах на YouTube, Reddit, Twitter и других сайтах. В 2009 году "Never Gonna
            Give You Up" стал одним из самых популярных видео на YouTube и был номинирован на премию
            MTV в категории "Лучшее видео на YouTube".</p>
95.         </section>
96.         <section>
97.             <h2>Как использовать Рикроллинг в своих шутках?</h2>
98.             <p>Чтобы использовать Рикроллинг в своих шутках, нужно подготовить ссылку на видео "Never
            Gonna Give You Up" и отправить ее вместо обещанной ссылки на другой контент. Можно также
            использовать фразу "Вы никогда не угадаете, что здесь" или что-то подобное, чтобы
            привлечь внимание.</p>
```



99. <p>Например, вы можете использовать фейковую ссылку, которая выглядит как настоящая, но ведет на "Never Gonna Give You Up". Вот пример:</p>

100. <p><a href="https://www.youtube.com/watch?v=dQw4w9WgXcQ&ab\_channel=RickAstley" target="\_blank">https://bmstu.ru/</a></p>

101. </section>

102. <section class="meme">

103. 

104. <h2>Рикролл</h2>

105. <div>

106. <p>We're no strangers to love</p>

107. <p>You know the rules and so do I</p>

108. <p>A full commitment's what I'm thinking of</p>

109. <p>You wouldn't get this from any other guy</p>

110. <br>

111. <p>I just wanna tell you how I'm feeling</p>

112. <p>Gotta make you understand</p>

113. <br>

114. <p>Never gonna give you up</p>

115. <p>Never gonna let you down</p>

116. <p>Never gonna run around and desert you</p>

117. <p>Never gonna make you cry</p>

118. <p>Never gonna say goodbye</p>

119. <p>Never gonna tell a lie and hurt you</p>

120. <br>

121. <p>We've known each other for so long</p>

122. <p>Your heart's been aching, but</p>

123. <p>You're too shy to say it</p>

124. <p>Inside, we both know what's been going on</p>

125. <p>We know the game and we're gonna play it</p>

126. <br>

127. <p>And if you ask me how I'm feeling</p>

128. <p>Don't tell me you're too blind to see</p>

129. <br>

130. <p>Never gonna give you up</p>

131. <p>Never gonna let you down</p>

132. <p>Never gonna run around and desert you</p>

133. <p>Never gonna make you cry</p>

134. <p>Never gonna say goodbye</p>

135. <p>Never gonna tell a lie and hurt you</p>

136. <br>

```
137. <p>(Ooh, give you up)</p>
138. <p>(Ooh, give you up)</p>
139. <p>Never gonna give, never gonna give</p>
140. <p>(Give you up)</p>
141. <p>Never gonna give, never gonna give</p>
142. <p>(Give you up)</p>
143. <br>
144. <p>We've known each other for so long</p>
145. <p>Your heart's been aching, but</p>
146. <p>You're too shy to say it</p>
147. <p>Inside, we both know what's been going on</p>
148. <p>We know the game and we're gonna play it</p>
149. <br>
150. <p>I just wanna tell you how I'm feeling</p>
151. <p>Gotta make you understand</p>
152. <br>
153. <p>Never gonna give you up</p>
154. <p>Never gonna let you down</p>
155. <p>Never gonna run around and desert you</p>
156. <p>Never gonna make you cry</p>
157. <p>Never gonna say goodbye</p>
158. <p>Never gonna tell a lie and hurt you</p>
159.</div>
160.
161. <a href="https://www.youtube.com/watch?v=d0w4w9WgXcQ" class="btn">Посмотреть видео</a>
162. </section>
163.
164. <h1>Тест на Рикролл</h1>
165. <p>Ответьте на вопросы ниже и узнайте, насколько хорошо вы знакомы с Рикроллом!</p>
166. <form>
167. <p>1. Какое видео ассоциируется с понятием "Рикролл"?</p>
168. <label><input type="radio" name="q1" value="a">"Never Gonna Give You Up" Рика
    Эстли</label><br>
169. <label><input type="radio" name="q1" value="b">"Take On Me" A-ha</label><br>
170. <label><input type="radio" name="q1" value="c">"Sweet Child O' Mine" Guns N' Roses</label><br>
171. <p>2. Какой год был выпущен видеоклип "Never Gonna Give You Up"?</p>
172. <label><input type="radio" name="q2" value="a">1987</label><br>
173. <label><input type="radio" name="q2" value="b">1989</label><br>
174. <label><input type="radio" name="q2" value="c">1991</label><br>
175. <p>3. Какую награду получил Рик Эстли за "Never Gonna Give You Up"?</p>
```

```

176. <label><input type="radio" name="q3" value="a">Престижную музыкальную премию
      Грэмми</label><br>
177. <label><input type="radio" name="q3" value="b">Награду MTV за лучшее музыкальное
      видео</label><br>
178. <label><input type="radio" name="q3" value="c">Никакой награды он за это видео не
      получал</label><br>
179. <br>
180. <input type="button" value="Проверить ответы" onclick="checkAnswers()">
181. </form>
182. <p id="result"></p>
183. <script>
184.   function checkAnswers() {
185.     var q1 = document.querySelector('input[name="q1"]:checked').value;
186.     var q2 = document.querySelector('input[name="q2"]:checked').value;
187.     var q3 = document.querySelector('input[name="q3"]:checked').value;
188.     var correctAnswers = 0;
189.     if (q1 == "a") { correctAnswers++; }
190.     if (q2 == "b") { correctAnswers++; }
191.     if (q3 == "c") { correctAnswers++; }
192.     var result = document.getElementById("result");
193.     if (correctAnswers == 3) {
194.       result.innerHTML = "Поздравляем! Вы настоящий эксперт по Рикролла!";
195.     } else if (correctAnswers == 2) {
196.       result.innerHTML = "Вы знаете неплохо Рикролла, но есть некоторые проблемы в знаниях.
          Попробуйте еще раз!";
197.     } else {
198.       result.innerHTML = "К сожалению, вы недостаточно хорошо знаете Рикролла. Попробуйте еще раз!";
199.     }
200.   }
201. </script>
202. <div>
203.   <h2>Проверка ссылки на Рикролла</h2>
204.   <label for="url">Введите URL-адрес:</label>
205.   <input type="text" id="url">
206.   <button onclick="checkUrl()">Проверить ссылку</button>
207.   <p id="result"></p>
208. </div>
209.
210. <script>
211.   function checkUrl() {

```

```
212.   var urlInput = document.getElementById("url");
213.   var url = urlInput.value;
214.
215.   if (url.includes("dQw4w9WgXcQ")) {
216.       document.getElementById("result").innerHTML = "Это Рикролл!";
217.   } else {
218.       document.getElementById("result").innerHTML = "Это не Рикролл.";
219.   }
220. }
221.</script>
222.
223.</main>
224.</body>
```

## Лабораторная работа № 4

Четвёртая лабораторная работа по Python была посвящена применению ранее полученных знаний и опыта работы с фреймворком Flask, HTML, CSS и JavaScript. Было изучено использование классов и списков в Python, а также применение их при разработке web-приложения с использованием Flask. В рамках лабораторной работы было разработано приложение для создания списка задач с возможностью добавления, редактирования и удаления задач. Получен опыт в создании и использовании классов и списков в Python, а также в интеграции данной функциональности с веб-приложением на Flask.

Листинг программного кода этой лабораторной работы представлен далее:

```
1.  from flask import Flask, render_template, request
2.  app = Flask(__name__)
3.
4.  class Книга(object):
5.      def __init__(self, название, автор, год_издания):
6.          self.название = название
7.          self.автор = автор
8.          self.год_издания = год_издания
9.
10.     def __str__(self):
11.         return f"Книга: {self.название}, Автор: {self.автор}, Год издания: {self.год_издания}"
12.
13.     class Библиотека(object):
14.         def __init__(self, название, книги):
15.             self.название = название
16.             self.книги = книги
17.
18.         def __str__(self):
19.             return f"Библиотека: {self.название}"
20.
21.     # Создаем экземпляры класса Книга
22.     книга1 = Книга("Преступление и наказание", "Федор Достоевский", 1866)
23.     книга2 = Книга("1984", "Джордж Оруэлл", 1949)
24.     книга3 = Книга("Улисс", "Джеймс Джойс", 1922)
25.     книга4 = Книга("Гарри Поттер и философский камень", "Джоан Роулинг", 1997)
26.     книга5 = Книга("Великий Гэтсби", "Фрэнсис Скотт Фицджеральд", 1925)
27.
```

```
28. # Создаем экземпляры класса Библиотека
29. библиотека1=Библиотека("Центральная библиотека", [книга1, книга2, книга3])
30. библиотека2=Библиотека("Детская библиотека", [книга4, книга5])
31.
32. @app.route("/")
33. def hello_world():
34.     return render_template('index.html', библиотека1=библиотека1, библиотека2=библиотека2)
35.
36. if __name__ == "__main__":
37.     app.run()
```