

CERTIFICATE

OF INTERNSHIP



This certificate is presented to:

Камалова Нилуфар Ильхомовна

FOR PARTICIPATION AND PUBLICATION OF THE PAPER ENTITLED

**РОЛЬ СЕМИОТИЧЕСКОГО ПОДХОДА В
ОБУЧЕНИИ ЯЗЫКАМ ПРОГРАММИРОВАНИЯ**

In an International Conference on Educational Discoveries and Humanities, Published online with E- Conference Series, Hosted online from Plano, Texas, USA.

ISSN (E): 2835-3196

SJIF: 5.385



E- Conference Series

Open Access | Peer Reviewed | Conference Proceedings



Date: 27/11/2023



harpermiller
Harper Miller

www.econferenceseries.com

РОЛЬ СЕМИОТИЧЕСКОГО ПОДХОДА В ОБУЧЕНИИ ЯЗЫКАМ ПРОГРАММИРОВАНИЯ.

Камалова Нилуфар Ильхомовна

Бухарский государственный университет преподаватель кафедры”
Прикладная математика и технологии программирования”, доктор
философских наук (PhD)

Самадов Авазбек Шавкатович

Бухарский государственный университет студент 1-2PM-23

Аннотация

В статье представлена информация о подходах, используемых при обучении языкам программирования. Из этих подходов подробно рассматривается семиотический подход. Раскрыта сущность исследований ученых и их научных трудов, в основе которых лежит семиотический подход к обучению языкам программирования. Обоснование семиотического подхода в обучении языкам программирования показано на конкретных примерах.

Ключевые слова: Семиотика, Программирование, подходы к обучению, tkinter, функции, система символов, навыки программирования, язык программирования Python.

Annotation

The article provides information about the approaches used in teaching programming languages. Of these approaches, the semiotic approach is considered in detail. The essence of the research of scientists and their scientific works, which are based on a semiotic approach to teaching programming languages, is revealed. The rationale for the semiotic approach in teaching programming languages is shown by concrete examples.

Keywords: Semiotics, Programming, learning approaches, tkinter, functions, symbol system, programming skills, Python programming language.

Подходы к обучению предоставляют различные способы объяснения и демонстрации концепций программирования. Это помогает студентам



глубоко понять основные принципы и идеи языка программирования, а также позволяет более эффективно использовать язык и разрабатывать более сложные программы. Подходы, включающие практическую работу и проекты, позволяют студентам получить реальный опыт программирования. Они выполняют задачи, решают проблемы и создают программы, которые помогают укреплять и улучшать навыки программирования. Различные подходы к обучению программированию позволяют студентам самостоятельно осваивать язык программирования. Самообучение развивает навыки самообразования и критического мышления. Некоторые подходы к обучению программированию, такие как участие в командах программистов или выполнение проектов в группе, позволяют студентам обмениваться опытом с другими программистами. Общение с опытными программистами, пересмотр кода и участие в обсуждениях кода могут помочь уменьшить количество ошибок и разработать более эффективные подходы к программированию. В целом, подходы к обучению программированию играют важную роль в формировании навыков программирования, обеспечивая глубокое понимание концепций, практическую работу, самообучение, обмен опытом и универсальные навыки, которые можно применять в различных языках программирования.

Существует несколько подходов к обучению языкам программирования. В качестве примеров можно привести системный подход, деятельностный подход, когнитивный подход, проблемный подход, семиотический подход [4]. Семиотический подход в программировании – методология, основанная на принципах семиотики, изучающей знаки и их семантические функции. Используя семиотический подход, разработчики стремятся создать программное обеспечение, которое будет понятным, легко принимаемым и эффективным для пользователей. В семиотическом подходе программирование рассматривается не только как умение писать код, но и как процесс общения программиста с компьютером. Семиотика в программировании рассматривает программу как набор символов с определенной семантикой. Программисты используют такие символы, как переменные, функции, операторы и константы, для создания кода, который имеет определенное значение и выполняет определенные операции. Семиотический подход помогает понять значение каждого символа и понять



их взаимодействие в контексте программирования. Семиотический подход позволяет рассматривать программирование как связь между программистом и компьютером. Это помогает точнее передавать инструкции на компьютер.

В основе семиотического подхода к обучению языкам программирования лежат исследования многих ученых. Семиотический аспект программирования, в том числе Пьеры Биани[1], который исследовал программирование как семиотическое моделирование в Иксель Брюне[2]. Одним из ученых, изучавших программирование как систему символов, был Григорий Райхерт[3]. Он изучает различные аспекты программирования с семиотической точки зрения, включая синтаксис и семантику программ, их взаимодействие с пользователями и другими программами, а также процессы разработки программного обеспечения. Он также разработал множество примеров и анализов различных программ, чтобы проиллюстрировать применение семиотического подхода к программированию. Понимание программирования как процесса использования символов и символов, представленное в его исследовании, позволило лучше понять принципы программирования и улучшить навыки разработки программного обеспечения.

Семиотический подход помогает программистам глубже анализировать и понимать код, который они пишут, и взаимодействие между его символами. Это помогает выявлять возможные ошибки или несоответствия в коде и улучшает его качество и разборчивость. В модели семиотики знания передаются не непосредственно обучающемуся, а через текст, формулу, графическую модель.

Мы подробно рассмотрим использование семиотического подхода в обучении языкам программирования на следующем примере. Объясняя понятие функции, возьмем в качестве примера мясорубку. Функция измельчителя в том, чтобы разбить продукт что в него кладут он разделяет продукт на кусочки. Функции также выполняют аналогичную функцию. Если в фарш положить мясо, то в результате получится мясной фарш, а если в него положить картофель, то получится фарш из картофеля. Как мы видим на примере функции:

```
def fakt(n):  
    p=1
```



```
for i in range(1,n+1):
    p=p*i
return p
```

любое число, введенное в этой функции, возвращает ее факториал. Итак, задача нашей функции состоит в вычислении факториала.

Чтобы получить четкое представление о функции, будет представлено несколько готовых структурированных функций, и учащемуся нужно будет произнести их, проанализировав задачу, а также то, какой результат она возвращает при определенном входящем значении. В качестве примера можно взять следующие функции:

1-функция	2-функция	3-функция
<pre>def b(n,i): if n % i ==0: t=True else: t=False return t</pre>	<pre>def y(s): n=0 for i in range (len(s)): n=n+int(s[i]) return n</pre>	<pre>def max5(a,b,c,d,e): if a<b: a=b if a<c: a=c if a<d: a=d if a<e: a=e return a</pre>

Книжный шкаф можно использовать для создания представления о переменных. Книжный шкаф используется для хранения книг. В программировании значения также сохраняются в пользовательских разделах, как на полке. На книжной полке книги и журналы можно разложить по отдельным разделам. В программировании также требуется специальное место для хранения значения. То есть он использует int для целых чисел, float для действительных чисел, str для строк, bool для логических типов. Мы можем взять книги с книжной полки и заменить ее другими книгами. Точно так же переменная может принимать значение и изменять это значение во



время программы. Чтобы составить представление об этом, к требованиям предъявляются следующие программы:

1- программа	2- программа	3- программа
a=5 b=3 a=a+b b=a-b print(a+b)	a=5 b=3 a=a*b b=a/b print(a+b)	a=5 b='3' print(a*b)

Семиотика может быть использована для передачи оператора while repeat следующим образом. Чтобы подняться на скалу, необходимо построить лестницу. Каждый раз перед тем, как поставить новую ступеньку лестницы, проверяется, достигла ли она конца скалы. Если он еще не достигнут, добавляется новая лестница, и добавление лестницы прекращается только по достижении вершины. Мы можем увидеть это в приложении ниже:

```
a=int(input())
k=0
while a>0:
    k=k+a%10
    a=a//10
print(k)
```

в этом случае число делится на целое 10, пока он не станет равным 0, и каждый раз добавляются остатки, становясь равными 10. После этого студентам предлагаются следующие программы:

1- программы	2- программы	3- программы
n=int(input()) k=0 while n>0: k=k+n%10 n=n//10 print(k)	n=int(input()) k=0 while n>0: k=k+2 n=n-1 print(k)	while True: a=int(input()) b=int(input()) print(a+b)



Важные элементы семиотического подхода:

- этот подход уделяет большое внимание символической системе языка. Студенты учатся распознавать и понимать ключевые слова, операторы и символы, используемые в языке программирования.

- семиотический подход фокусируется на понимании значения каждого символа. Студенты учатся интерпретировать значения ключевых слов и символов и понимать, как они взаимодействуют друг с другом.

- этот подход также направлен на обучение синтаксическим правилам языка программирования. Студентов обучают правильному использованию синтаксических конструкций и созданию правильных фраз и инструкций.

- важным элементом семиотического подхода является практическая работа с языком программирования. Студенты активно используют язык программирования, чтобы лучше понять его символическую систему и применить свои знания на практике.

В целом, семиотический подход к обучению программированию помогает программистам лучше понимать и описывать природу программирования, улучшать взаимодействие с компьютером и другими программистами и создавать более понятный код. Это позволяет программистам не только программировать компьютер, но и взаимодействовать с помощью языка символов и семантики, создавая более эффективное и выразительное программное обеспечение.

Литература:

1. Пьери Биани "Семиотический аспект программирования", 1988 г. Издательство: Springer, 188 с.
2. Иксель Брюнее "Понимание программирования как процесса семиотического моделирования", 2004 г. Издательство: Morgan & Claypool Publishers 212 с.
3. Григорий Е. Райхерц "Семиотический подход к изучению программирования", 1990 г. Издательство: Лобукс, 248 с.
4. Ilhomovna K. N. Dasturlash tilini andragogika prinsiplari va dasturlash tillarini o'qitish yondashuvlari integratsiyasi asosida o'qitish //Proceedings of International Conference on Scientific Research in Natural and Social Sciences. – 2023. – Т. 2. – №. 5. – С. 91-98.

