

МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМ. Р.Е. АЛЕКСЕЕВА»
(НГТУ)

*Институт ядерной энергетики и технической физики
Кафедра «Ядерные реакторы и энергетические установки»*

УТВЕРЖДАЮ:



Директор института

Хробостов А.Е.

«08» сентября 2018 г

Методические рекомендации

*по организации самостоятельной работы
по дисциплине «Компьютерные технологии»*

Направление подготовки

14.04.02 «Ядерные физика и технологии» код
и наименование направления подготовки

Магистерская программа

«Ядерные реакторы и энергетические установки»

Квалификация (степень)
Магистр

Форма обучения

очная

Нижний Новгород,
2018

Методические рекомендации по организации самостоятельной работы по дисциплине «Компьютерные технологии» предназначены для магистров первого и второго курсов, обучающихся по направлению 14.04.02 – «Ядерные физика и технологии».

Цель методических рекомендаций: помочь студентам при самостоятельном изучении учебной программы с использованием лекционных материалов и рекомендуемой учебно-методической литературы.

Самостоятельная работа направлена на развитие профессиональных компетенций дисциплины ПКС-1, ПКС-2, ПКС-3.

Технологии в образовательной деятельности»:

- систематическая проработка лекций, учебной и специальной технической литературы; □
- оформление отчетов по лабораторным работам; □
- подготовка к зачету, экзамену. □

2. Методические рекомендации для студентов по конкретным видам самостоятельной работы

- 1) **Систематическая проработка лекций, учебной и специальной технической литературы.**
- 2) **Оформление отчетов по лабораторным работам.**

Методические рекомендации:

1. Обратитесь к методическим указаниям по проведению лабораторных работ и оформите работу, указав название, цель и краткий порядок проведения работы, выполните необходимые расчеты и построение необходимых графиков, сформулируйте выводы по результатам работы, выполненной на учебном занятии.
2. Подготовьтесь к защите выполненной работы: повторите основные теоретические положения и ответьте на контрольные вопросы, представленные в методических указаниях по проведению лабораторных работ.

Показатели оценки результатов внеаудиторной самостоятельной работы:

- самостоятельная работа студентов по указанным дисциплинам в установленном плане объема контролируется на коллоквиумах, проводимых в ходе лабораторных работ.
- оформление лабораторных работ в соответствии с требованиями, описанными в методических указаниях;
- качественное выполнение всех этапов работы;
- необходимый и достаточный уровень понимания цели и порядка выполнения работы;
- правильное оформление выводов работы;

3) Подготовка к зачету, экзамену. Методические рекомендации:

1. Внимательно прочитайте материал лекций.
2. Постарайтесь разобраться с непонятными, в частности, новыми терминами, используя рекомендованную литературу.
3. Ответьте на контрольные вопросы для самопроверки, имеющиеся в данных методических указаниях.

Показатели оценки результатов внеаудиторной самостоятельной работы:

- качество уровня освоения учебного материала;
- умение использовать теоретические знания при выполнении лабораторных работ или ответе на вопросы; – обоснованность и четкость изложения ответа.

3. Перечень рекомендуемой литературы

№ п/п	Автор(ы)	Заглавие	Издательство, год издания	Назначение, вид издания, гриф	Кол-во экз. в библиотеке
Основная литература					
1	Иванов В.В.	Компьютерные методы обработки экспериментальных данных	Нижний Новгород, [Б.и.], 2015	Курс лекций	Электронный ресурс
2	Иванов А.А.	Автоматизация технологических процессов и производств	Москва, ФОРУМ, 2011.	Учебник. Рекомендовано УМО Вузов по образованию в области автоматизированного машиностроения	22
Дополнительная литература					
1	Тревис Дж.	LabVIEW для всех	Москва ДМК Пресс, 2005.	Учебник.	Электронный ресурс
2	Берлинер Э.М.	САПР в машиностроении	Москва, ФОРУМ, 2008.	Учебник. Рекомендован УМО Вузов РФ по образованию в области транспортных машин и транспортнотехнологических комплексов	1

Методические указания к лабораторным занятиям

Методические указания к лабораторным занятиям по теме «Компьютерные технологии», Н.Новгород, 2013 г. (Доступны в электронном виде на кафедре «Ядерные реакторы и энергетические установки»).

Перечень контрольных вопросов по дисциплине " Компьютерные технологии в "

Перечень вопросов, выносимых на зачет во втором семестре:

1. Интегрированные прикладные системы
2. Таймеры

3. История развития вычислительной техники
4. Аналого-цифровые преобразователи (АЦП)
5. История развития операционных систем (ОС)
6. Характеристики АЦП
7. Формы представления данных в вычислительных системах
8. Платы ввода/вывода данных ПК
9. Выходной каскад логического элемента
10. Методы передачи сигналов и защита от помех
11. Современная интерпретация принципа Фон-Неймана
12. Стандарт NAMUR
13. Процесс записи в ВУ
14. Виды помех
15. Процесс чтения в ВУ
16. Гальваническая развязка
17. Процессор
18. Интерфейсы
19. Архитектура процессора
20. Параллельный интерфейс 1. Архитектура Фон-Неймана
21. Промышленные интерфейсы
22. Конвейерная архитектура
23. Протоколы обмена данными в последовательных интерфейсах
24. Гарвардская архитектура
25. Протокол MODBUS
26. RISC-архитектура
27. HART-протокол
28. Компьютерная память
29. Режимы работы вычислительных систем
30. Прямой доступ к памяти DMA (ПДП)
31. Виртуальные приборы
32. Прерывание
33. Методы программирования вычислительных устройств
34. Выходной каскад логического элемента
35. Микроконтроллеры

Перечень вопросов, выносимых на экзамен в третьем семестре:

1. Основное назначение, классификация сетей
2. Принципы работы в локальных и глобальных сетях.
3. Организация совместной работы в локальных и глобальных сетях
4. Программное и аппаратное обеспечение, назначение классификация
5. Прикладной уровень ПО.
6. Служебный уровень ПО
7. Системный уровень ПО.
8. Базовый уровень ПО

9. Классификация операционных систем
10. Операционные оболочки, основные функции, классификация.
11. Сетевые операционные системы.
12. Системное программное обеспечение
13. Системные управляющие и системные обрабатывающие программы.
14. Операционная система как совокупность программных средств
15. Драйверы.
16. Назначение, основные требования к служебному ПО.
17. Защита компьютера, обнаружение и восстановление зараженных файлов.
18. Антивирусное программное обеспечение

4. Задания для самостоятельного выполнения по дисциплине

4.1. Сгенерировать массив данных.

4.2. Создать приложение, в котором будут применены функции генерации сигналов, массивы, строки а так же операции визуализации данных.

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

Нижегородский государственный технический университет им. Р.Е. Алексеева

Кафедра «Ядерные реакторы и энергетические установки»

УТВЕРЖДАЮ:

Директор института



Хробостов А.Е.
«08» сентября 2018 г.

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО
ПОДГОТОВКЕ ЛАБОРАТОРНЫХ РАБОТ**

по дисциплине «Компьютерные технологии»

(наименование дисциплины)

14.04.02 «Ядерная физика и технологии»

(код и наименование направления подготовки)

Ядерные реакторы и энергетические установки

(наименование программы магистратуры)

Квалификация (степень) _____ магистр

Форма обучения _____ очная

(очная, очно-заочная, заочная)

Нижний Новгород,
2018

Лабораторная работа № 1. Установка Python и настройка программной среды. Базовые конструкции.

1. Перед установкой необходимо определить разрядность системы – 32битная или 64-битная: Этот компьютер->Свойства. Информация будет доступна в поле Тип системы.
2. Заходим на официальный сайт python <https://www.python.org/downloads/windows/> и переходим по ссылке «Latest Python 3 Release - Python x.x.x». В случае 64-битной версии Windows скачиваем Windows x86-64 executable installer, в противном случае Windows x86 executable installer и запускаем установщик.
3. На втором экране мастера установки необходимо выбрать опцию «Add Python 3.6 to the PATH».
4. Проверяем корректность установки. Запускаем командную строку (Windows + кнопка «R»), вводим в поле open: cmd и нажимаем Enter. В открывшемся окне вводим:

```
C:\Users\olga>python --version
```

Результат выполнения команды должен сообщить установленную версию Python, например: Python 3.6.3.

Если этого не произошло, проверьте, добавился ли путь к Python в PATH: выполните команду set в командной строке C:\Users\olga>set

В результатах выполнения найдите переменную PATH и убедитесь, что в ней присутствует путь до Python и Python\Scripts. Если в переменной их нет, то можно добавить их или переустановить Python.

2. Установка Visual Studio Code.

Писать программы на Python можно в любом текстовом редакторе, добавив в название файла расширение .py. Например, можно использовать Notepad++ для редактирования и запускать скрипты Python из командной строки.

Однако, удобнее это делать в редакторах, предназначенных для управления кодом (Sublime Text, PyCharm, PyDev, WingWare, и т.д.). Далее представлены инструкции по работе с Visual Studio Code (который позволяет редактировать и запускать скрипты Python).

1. Заходим на официальный сайт <https://code.visualstudio.com/> загружаем подходящий дистрибутив и выполняем установку.
2. Для ознакомления с Visual Studio Code можно обратиться к этому источнику <https://habrahabr.ru/company/microsoft/blog/262523/>
3. Во вкладке расширения в строке поиска вбиваем Python и устанавливаем Python.

Далее переходим в терминал (Вид->Интегрированный терминал) и выполняем:

pip install pylint

4. Устанавливаем Code Runner Extension (вкладке расширения в строке поиска вбиваем Code Runner)

3.Создание первой программы.

1. В проводнике создаем новую папку, далее создаем файл с именем HellowWorld.py
2. Добавляем в файл строчку `print("Hellow, World!")`
1. Выполните программу Run Code из контекстного меню файла.
- 3.Создание переменных. Измените программу следующим образом:

```
name = input("What is your name? ")  
print("Hallow,", name, "!" )
```

4.Задания для выполнения лабораторной работы.

1. Создайте программу, которая просит пользователя ввести свое имя и возраст. Распечатайте сообщение, которое сообщает, через какое время ему исполнится 100 лет. Дополните программу:

- а. Запросите еще одно число и выведете на экран соответствующее количество копий предыдущего сообщения.
- б. Распечатайте каждое новое сообщение с новой строки (см. лаб.2 - Экранированные последовательности).

2. Спросите у пользователя число. В зависимости от того, является ли число четным или нечетным, распечатайте соответствующее сообщение пользователю.

Дополните программу:

- а. Если число кратно 4, сообщите об этом.
- б. Спросите у пользователя два числа: делимое (назовите его num) и делитель (check). Если делимое делится без остатка на делитель, сообщите об этом пользователю. Если нет, напечатайте другое соответствующее сообщение.

2. Лабораторная работа № 2. Работа со строками в Python.

Строки в Python — упорядоченные последовательности символов, используемые для хранения и представления текстовой информации, поэтому с помощью строк можно работать со всем, что может быть представлено в текстовой форме.

Задания для выполнения лабораторной работы.

1. Дана текстовая строка, содержащая буквы латинского алфавита, пробелы, запятые и точки. Отформатируйте этот текст по следующим правилам:

- 1) В начале и конце строки не должно быть пробелов.
- 2) Все слова разделяются ровно одним пробелом.
- 3) Точки и запятые пишутся слитно с предыдущим словом, после знака препинания ставится пробел.

Выведите полученную строку.

Пример:

входные данные:

Hello , world.

выходные данные:

Hello, world.

2. Входные данные: Дана текстовая строка. Выходные данные:

1. Сначала выведите третий символ этой строки.
2. Во второй строке выведите предпоследний символ этой строки.
3. В третьей строке выведите первые пять символов этой строки.
4. В четвертой строке выведите всю строку, кроме последних двух символов.
5. В пятой строке выведите все символы с четными индексами (считая, что индексация начинается с 0, поэтому символы выводятся начиная с первого).
6. В шестой строке выведите все символы с нечетными индексами, то есть начиная со второго символа строки.
7. В седьмой строке выведите все символы в обратном порядке.
8. В восьмой строке выведите все символы строки через один в обратном порядке, начиная с последнего.
9. В девятой строке выведите длину данной строки.

Примеры входные данные:

AbraKadabra выходные

данные:

r r

AbraK

Abrakadab
Arkdba
baaar
arbadakarba
abdkrA
11

Лабораторная работа № 3. Списки.

1. Списки в Python

Списки в Python – упорядоченные изменяемые коллекции объектов произвольных типов (почти как массив, но типы могут отличаться).

Задания для выполнения лабораторной работы.

1. Создайте список, например, такой:

```
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
```

и напишите программу, которая выводит все элементы этого списка, которые меньше 5. Дополните программу:

а. Вместо того, чтобы печатать элементы один за другим, создайте новый список, в котором есть все элементы менее 5 из этого списка и распечатайте этот новый список.

б. Запросите у пользователя число и верните список, содержащий только элементы из исходного списка a, которые меньше, чем число, указанное пользователем.