**Правительство Российской Федерации**

**Нижегородский филиал**

**Федерального государственного автономного образовательного учреждения высшего профессионального образования
"Национальный исследовательский университет
"Высшая школа экономики"**

Факультет бизнес-информатики и прикладной математики

**Программа дисциплины**

**«Введение в программную инженерию»**

для направления 231000.62 – Программная инженерия

подготовки бакалавра

Автор программы: доцент Штанюк А.А.

Одобрена на заседании кафедры «Базовая кафедра МЕРА» «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2012г.

Зав. кафедрой Н.И. Кащеев

Рекомендована секцией УМС «Прикладная математика» «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2012г.

 Председатель В.А. Калягин

Утверждена УМС НИУ ВШЭ – Нижний Новгород «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2012 г.

Председатель Н.С. Петрухин

Нижний Новгород, 2012 г.

*Настоящая программа не может быть использована другими подразделениями университета и другими вузами без разрешения кафедры-разработчика программы.*

# Область применения и нормативные ссылки

Настоящая программа учебной дисциплины устанавливает минимальные требования к знаниям и умениям студента и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих данную дисциплину, учебных ассистентов и студентов направлений подготовки 231000.62 «Программная инженерия» подготовки бакалавра, изучающих дисциплину «Введение в программную инженерию».

Программа разработана в соответствии с образовательным стандартом федерального государственного образовательного автономного учреждения высшего профессионального образования Высшей школы экономики.

# Цели освоения дисциплины

Целями освоения данной дисциплины являются как получение теоретических знаний в области современных языков и средств скриптового программирования, таких как Bash, Perl, Python.

# Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:

* Знать языковые средства bash, perl, python и основные принципы программирования на этих языках, а также базовые принципы современной скриптовой обработки информации
* развить «программистское» мышление
* овладеть навыками программирования для применения современных высокоуровневых структур данных, присущих программам на скриптовых языках

В результате освоения дисциплины студент осваивает следующие компетенции:

| Компетенция | Код по НИУ | Дескрипторы – основные признаки освоения (показатели достижения результата) | Формы и методы обучения, способствующие формированию и развитию компетенции |
| --- | --- | --- | --- |
| Способность к саморазвитию, повышению своей квалификации и мастерства | ОК-6 | Студент развивает самооценку в ходе публичных дискуссий, повышает свою квалификацию, участвуя в разработке индивидуальных и коллективных проектов | Подготовка к проектным семинарам, участие в публичных дискуссиях коллектива учебной группы, разработка алгоритмов и программного кода для индивидуальных и коллективных проектов |
| знакомство с архитектурой ЭВМ и систем | ПК-9 | Студент использует современные компьютерные технологии при выполнении заданий учебной дисциплины, применяет современные системы программирования для разработки программного кода, интерпретирует результаты своих разработок | Совершенствование в овладении инструментальными средствами современных систем программирования для разработки программного кода  |
| умение применять основы информатики и программирования к проектированию,конструированию и тестированию программных продуктов | ПК-10 | Студент использует современные программные продукты для разработки программного обеспечения | Разработка алгоритмов и программного кода для практических и домашних заданий, индивидуальных проектов, подготовка к контрольным работам |
| навыки использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных | ПК-15 | Студент демонстрирует умение работать с современными средствами разработки программного обеспечения, знает основы работы с СУБД, умеет работать в локальной сети | Регулярное выполнение практических и домашних заданий, выполнение проектов |

# Место дисциплины в структуре образовательной программы

Настоящая дисциплина относится к циклу дисциплин информационных технологий, обеспечивающих подготовку бакалавра. Курс является общепрофессиональным из цикла курсов подготовки современного специалиста в области разработки ПО. Курс опирается на следующие курсы из цикла «Общие математические и естественнонаучные дисциплины» - фундаментальный курс «Дискретная математика» - федеральный компонент и «Программирование на языке Си» - вузовский компонент. Курс играет важную роль в развитии понимания будущими специалистами низкоуровневого программирования и функционирования компьютерной системы.

**Тематический план учебной дисциплины**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Наименование тем | Всего  | Лекции | Практические занятия | Самостоятельная работа |
| часов |
|  |   | Раздел 1. Язык BASH |
| 1.1 | Общие сведения о скриптовых языках программирования.  | 12 | 2 | 2 | 8 |
| 1.2 | Основы программирования на языке BASH | 12 | 2 | 2 | 8 |
|  |  | Раздел 2. Основы программирования на языке Perl |
| 2.1 | Введение в язык Perl. Фундаментальные типы данных. Строки и числа. | 11 | 2 | 4 | 5 |
| 2.2 | Выражения языка Perl. Основные алгоритмические структуры. | 12 | 2 | 2 | 8 |
| 2.3 | Основные структуры данных Perl: массивы и хэши. | 12 | 2 | 2 | 8 |
| 2.4 | Регулярные выражения Perl | 12 | 2 | 2 | 8 |
| 2.5 | Разработка процедур на языке Perl. Стандартный и файловый ввод/вывод | 9 | 2 | 2 | 5 |
|  |   |   | Раздел 3. Основы программирования на языке Python |
| 3.1 | Основы языка Python | 12 | 2 | 2 | 8 |
| 3.2 | Основные структуры данных в языке Python | 14 | 2 | 4 | 8 |
| 3.3 | Ввод/вывод на Python | 12 | 2 | 2 | 8 |
| 3.4 | Основы функционального программирования на Python | 12 | 2 | 2 | 8 |
| 3.5 | Модули и объектно-ориентированное программирование на Python | 12 | 4 | 2 | 6 |
| **Итого** | 144 | 28 | 28 | 88 |

# Формы контроля знаний студентов

|  |  |  |  |
| --- | --- | --- | --- |
| Тип контроля | Форма контроля | Модули | Параметры |
| 1 | 2 |  |  |
| Текущий контроль | Домашнее задание |  | 4 |  | Домашняя контрольная работа  |
| Итоговый | Зачет |  | \* |  | Устная форма, 2 вопроса и 1 задача |

## Критерии оценки знаний, навыков

Текущий контроль осуществляется в виде проверки выполнения практических работ. По каждой работе оформляется электронный отчёт с описанием хода выполнения заданий, обоснованием результатов и выводами.

Итоговый контроль: зачет на последней неделе 2 модуля. Учитываются результаты домашней работы (отчёты по практикам). Оценка определяется в соответствии с п. 10.

# Содержание дисциплины

**Раздел 1. ЯЗЫК BASH**

**Тема 1.1. Общие сведения о скриптовых языках программирования**

Классификация языков программирования. Определение скриптового языка. Особенности скриптовых языков. Преимущества и недостатки. Краткий обзор скриптовых языков.

*Основная литература*

*Сценарный язык[Электронный ресурс]:http://ru.wikipedia.org/wiki/Сценарный\_язык*

*Дополнительная литература*

*Богатырев Р. Природа и эволюция сценарных языков. //Мир ПК. – 2001. - №11*

**Тема 1.2. Основы программирования на языке BASH**

Синтаксические основы языка Bash. Символы и переменные. Стандартный ввод/вывод. Операции с файлами. Регулярные выражения. Основные команды unix и их использование в сценариях. Процедуры. Массивы. Управление выводом.

*Основная литература*

*Пик Д., О’Райли. Unix – инструментальные средства*

*Дополнительная литература*

*Тансли Д. Linux и Unix: программирование в Shell. Руководство разработчика, BHV, 2001.*

**Глава 2. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ PERL**

**Тема 2.1. Введение в язык Perl. Фундаментальные типы данных. Строки и числа**.

Классификация типов в Perl. Скалярные переменные. Основные операции. Представление чисел. Строки и операции над ними.

*Основная литература*

*Матросов А.В., Чаунин М.П. PERL:Программирование на языке высокого уровня. – СПб.:Питер, 2003*

*Дополнительная литература*

*Кристиансен Т., Торкингтон Н. Perl: библиотека программиста. – СПб: Питер, 2001*

**Тема 2.2. Выражения языка Perl. Основные алгоритмические структуры**.

Классификация операторов языка. Префиксная и постфиксная формы алгоритмических операторов. Системные переменные.

*Основная литература*

*Матросов А.В., Чаунин М.П. PERL:Программирование на языке высокого уровня. – СПб.:Питер, 2003*

*Дополнительная литература*

*Кристиансен Т., Торкингтон Н. Perl: библиотека программиста. – СПб: Питер, 2001*

**Тема 2.3. Основные структуры данных Perl: массивы и хэши.**

Понятие массива в языке Perl. Основные операции над элементами массива. Массивы и Хэши. Создание, заполнение и использование хэшей.

*Основная литература*

*Матросов А.В., Чаунин М.П. PERL:Программирование на языке высокого уровня. – СПб.:Питер, 2003*

*Дополнительная литература*

*Кристиансен Т., Торкингтон Н. Perl: библиотека программиста. – СПб: Питер, 2001*

**Тема 2.4**. **Регулярные выражения Perl**

Понятие регулярного выражения. Строительные блоки регулярных выражений. Примеры использования.

*Основная литература*

*Матросов А.В., Чаунин М.П. PERL:Программирование на языке высокого уровня. – СПб.:Питер, 2003*

*Дополнительная литература*

*Кристиансен Т., Торкингтон Н. Perl: библиотека программиста. – СПб: Питер, 2001*

**Тема 2.5. Разработка процедур на языке Perl. Стандартный и файловый ввод/вывод**

Процедуры с параметрами и без. Оператор shift. Разработка процедур и их использование. Работа с консолью: чтение, запись данных. Работа с файлами.

*Основная литература*

*Матросов А.В., Чаунин М.П. PERL:Программирование на языке высокого уровня. – СПб.:Питер, 2003*

*Дополнительная литература*

*Кристиансен Т., Торкингтон Н. Perl: библиотека программиста. – СПб: Питер, 2001*

**Глава 3. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ PYTHON**

**Тема 3.1** Основы языка Python

История возникновения и развития языка Python. Отличия от других скриптовых языков. Синтаксические особенности.

*Основная литература*

*Лутц. М. Изучаем Python. - М.: Символ плюс, 2011*

*Бизли Д. Python: подробный справочник. – М.: Символ плюс, 2010*

*Дополнительная литература*

*Саммерфилдт М. Программирование на Python, М.: BHV, 2009.*

**Тема 3.2** Основные структуры данных в языке Python

Классификация типов данных в языке Python. Массивы и списки. Наборы и словари. Операции доступа к элементам пользовательских типов. Работа с датой/временем.

*Основная литература*

*Лутц. М. Изучаем Python. - М.: Символ плюс, 2011*

*Бизли Д. Python: подробный справочник. – М.: Символ плюс, 2010*

*Дополнительная литература*

*Саммерфилдт М. Программирование на Python, М.: BHV, 2009.*

**Тема 3.3** Ввод/вывод на Python

Стандартный и файловый ввод/вывод. Форматирование данных. Пользовательские функции.

*Основная литература*

*Лутц. М. Изучаем Python. - М.: Символ плюс, 2011*

*Бизли Д. Python: подробный справочник. – М.: Символ плюс, 2010*

*Дополнительная литература*

*Саммерфилдт М. Программирование на Python, М.: BHV, 2009.*

**Тема 3.4** Основы функционального программирования на Python

Понятие функционального программирования. Лямбда-функции. Замыкания, генераторы и ленивые вычисления.

*Основная литература*

*Лутц. М. Изучаем Python. - М.: Символ плюс, 2011*

*Бизли Д. Python: подробный справочник. – М.: Символ плюс, 2010*

*Дополнительная литература*

*Саммерфилдт М. Программирование на Python, М.: BHV, 2009.*

**Тема 3.5** Модули и объектно-ориентированное программирование на Python

Разработка и подключение модулей. Определение класса. Конструкторы и деструкторы. Наследование.

*Основная литература*

*Лутц. М. Изучаем Python. - М.: Символ плюс, 2011*

*Бизли Д. Python: подробный справочник. – М.: Символ плюс, 2010*

*Дополнительная литература*

*Саммерфилдт М. Программирование на Python, М.: BHV, 2009.*

# Образовательные технологии

Традиционное чтение лекций. Решение задач по тематике лекций. Обсуждение прикладных и алгоритмических аспектов решаемых задач. Разработка учебных компьютерных программ.

## Методические рекомендации преподавателю

Темы индивидуальных заданий для проведения практических занятий должны отличаться для каждого нового учебного года

## Методические указания студентам

Рекомендуется подготовка к каждому занятию по заданиям, озвученным преподавателем на предыдущем занятии.

Для более глубокого усвоения курса предполагается использование студентами дополнительной литературы, работа в библиотеке, поиск информации в сети Интернет

# Оценочные средства для текущего контроля и аттестации студента

## Тематика практических заданий

1. Программирование задач администратора в unix-среде.
2. Обработка данных на языке Perl.
3. Использование регулярных выражений для анализа данных.
4. Знакомство с программированием на языке Python.
5. Функциональное программирование на Python.
6. Объектно-ориентированные возможности Python.

## Примеры заданий промежуточного /итогового контроля

***Практические задания к зачёту:***

* Задача 1. Написать программу, переименовывающую все файлы в каталоге в соответствии со значением счётчика (001.\*, 002.\*, …). Формат счётчика и каталог местоположения файлов задаются в виде параметров скрипта.
* Задача 2. Написать программу, выводящую список подкаталогов для заданного каталога с указанием суммарного размера и количества всех файлов, которые в них находятся.
* Задача 3. Написать программу, которая составляет отчёт о количестве разных типов файлов (по расширению), находящихся в заданном каталоге и всех его подкаталогах.
* Задача 4. Написать программу, которая в заданном каталоге архивирует все подкаталоги, не трогая файлов и создаёт отчёт, в котором перечислены старые имена каталогов и новые имена архивов.

# Порядок формирования оценок по дисциплине

Контроль знаний студентов включает формы текущего и итогового контроля. Текущий контроль осуществляется в течение двух модулей. В рамках учебного курса предусмотрены различные формы текущего контроля знаний и работы студентов на практических занятиях: практические задания (после каждого практического занятия, каждое по 80 минут), 1 домашняя контрольная работа. Каждая форма текущего контроля оценивается по 10-балльной шкале, оценка выставляется в рабочую ведомость преподавателя. По результатам текущего контроля организуются индивидуальные консультации в рамках второй половины рабочего дня преподавателя. Форма итогового контроля – зачет по окончании второго модуля. Каждая форма итогового контроля оценивается так же по 10-балльной шкале.

**Практическое задание:**

оценка в 10 баллов проставляется в исключительных случаях самостоятельно проведенной работы, результаты которой могут в дальнейшем использоваться в учебном процессе или в исследовательской работе студента;

оценка в 8-9 баллов проставляется при самостоятельно разработанном или удачно адаптированном и отлично представленном исследовании по выбранной тематике;

оценка в 6-7 баллов проставляется при своевременно выполненном и самостоятельно представленном исследовании по выбранной тематике;

оценка в 4-5 баллов проставляется при частичном, несамостоятельном участии в выполнении работ над заданием;

оценка в 2-3 балла проставляется, когда студент не может самостоятельно представить работу или когда работа носит явные признаки заимствований (работу предлагается переделать);

оценка в 1 балл проставляется при наличии каких-либо демонстративных проявлений безграмотности и неэтичного отношения к работе.

**Контрольная работа:**

высшая оценка в 9 баллов (10 баллов только в исключительных случаях) проставляется при полностью правильных ответах на вопросы и отличном выполнении заданий (правильном решении задачи, четком и исчерпывающем ее представлении);

почти отличная оценка в 8 баллов проставляется при полностью правильных ответах на вопросы и отличном выполнении заданий, но при отсутствии четкого и исчерпывающего представления решаемой задачи;

оценка в 7 баллов проставляется при правильных ответах на вопросы и правильном решении задачи, но при наличии отдельных неточностей в ответах на вопросы;

оценка в 6 баллов проставляется при наличии отдельных неточностей в ответах на вопросы (включая грамматические ошибки) или неточностях в решении задачи непринципиального характера (описки и случайные ошибки);

оценка в 5 баллов проставляется в случаях, когда в ответах на вопросы и в решении задачи имеются неточности и ошибки, свидетельствующие о недостаточном понимании изучаемой дисциплины и требующие дополнительного обращения к учебным материалам;

оценка в 4 балла проставляется при наличии серьезных ошибок в ответах на вопросы и в решении задачи, что свидетельствует о наличии пробелов в знании изучаемой дисциплины;

оценка в 3 балла проставляется при наличии лишь отдельных положительных моментов в ответах на вопросы и в решении задач, говорящих лишь о потенциальной возможности в последующем более успешного выполнения заданий; оценка в 3 балла, как правило, ведет к повторному решению дополнительной задачи;

оценка в 2 балла проставляется при полном отсутствии положительных моментов в ответах на вопросы и в решении задачи и, как правило, ведет к повторному написанию контрольной работы в целом;

оценка в 1 балл проставляется в тех случаях, когда наряду с неправильными ответами на вопросы и решением задачи имеют место какие-либо демонстративные проявления безграмотности или неэтичное отношение к изучаемой дисциплине.

**Зачет:**

На зачете (экзамене), представляющем собой письменные ответы на вопросы и решение задачи с последующим собеседованием, оценка проставляется следующим образом:

высшая оценка в 9 баллов (10 баллов только в исключительных случаях) проставляется при отличном выполнении заданий (полных, с примерами и возможными обобщениями ответах на вопросы, при правильном решении задачи и детальном ее представлении);

почти отличная оценка в 8 баллов проставляется при полностью правильных ответах на вопросы и решении задачи, но при отсутствии примеров и обобщений, а также детального представления решаемой задачи;

оценка в 7 баллов проставляется при правильных ответах на вопросы и правильном решении задачи, но при отсутствии пояснений и обобщений, а также детального представления решаемой задачи;

оценка в 6 баллов проставляется при наличии отдельных неточностей в ответах на вопросы или неточностях в решении задачи непринципиального характера (описки и случайные ошибки);

оценка в 4-5 баллов проставляется в случаях, когда в ответах на вопросы и в решении задачи имеются существенные неточности и ошибки, свидетельствующие о недостаточном понимании изучаемой дисциплины;

оценка в 2-3 балла проставляется при наличии лишь отдельных положительных моментов в ответах на вопросы и в решении задачи;

оценка в 1 балл проставляется в тех случаях, когда наряду с неправильными ответами на вопросы и решением задачи имеют место какие-либо демонстративные проявления безграмотности или неэтичное отношение к изучаемой дисциплине.

По результатам устного собеседования с преподавателем возможны корректировки оценки в ту или иную сторону.

Накопленная оценка за текущий контроль учитывает результаты студента следующим образом:

О*накопленная* = 0,5\* *Од/з* +0,5*\*Оаудитор.*

Способ округления накопленной оценки – арифметический.

Результирующая оценка за дисциплину рассчитывается по формуле:

*Орезульт = 0,6\*Онакопленная* + *0,4\*Озачет*

В диплом выставляет результирующая оценка по учебной дисциплине.

Способ округления результирующей оценки по учебной дисциплине – арифметический.

# Учебно-методическое и информационное обеспечение дисциплины

***Основная литература:***

*Матросов А.В., Чаунин М.П. PERL:Программирование на языке высокого уровня. – СПб.: Питер, 2003*

*Лутц. М. Изучаем Python. - М.: Символ плюс, 2011*

*Бизли Д. Python: подробный справочник. – М.: Символ плюс, 2010*

***Дополнительная литература***

*Богатырев Р. Природа и эволюция сценарных языков. //Мир ПК. – 2001. - №11*

*Кристиансен Т., Торкингтон Н. Perl: библиотека программиста. – СПб: Питер, 2001*

*Саммерфилдт М. Программирование на Python, М.: BHV, 2009.*

***Электронные ресурсы***

*Сценарный язык[Электронный ресурс]:http://ru.wikipedia.org/wiki/Сценарный\_язык*

*Пик Д., О’Райли. Unix – инструментальные средства*

# Материально-техническое обеспечение дисциплины

Мультимедийное оборудование – ноутбук, экран, проектор.

Состав программного обеспечения:

1. Операционная система: Windows или Linux
2. Офисный пакет: MS Office или Open Office или Libre Office
3. Пакет для создания pdf-файлов
4. Среда разработки DRScheme или DrRacket
5. Среда разработки MS Visual Studio Express Edition (C++)

Используется ПО в компьютерном классе НИИТ. В НИУ ВШЭ – Нижний Новгород студентам предоставляется возможность самостоятельной работы с электронными ресурсами информации, периодической литературой.

В компьютерном классе (НИИТ) доступ on-line

Автор программы, доцент Штанюк А.А.