

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«Приднестровский государственный университет
имени Т.Г. Шевченко»

Рыбницкий филиал

Кафедра информатики и программной инженерии

УТВЕРЖДАЮ

Директор Рыбницкого филиала ПГУ им. Т.Г. Шевченко

профессор  И.А. Павлинов



«09» _____ 10 _____ 2020г.

РАБОЧАЯ ПРОГРАММА

на 2020/2021 учебный год

Учебной ДИСЦИПЛИНЫ

«ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ»

Направление подготовки:

09.03.04 «Программная инженерия»

Профиль подготовки:

«Разработка программно-информационных систем»

квалификация (степень) выпускника

Бакалавр

Форма обучения:

очная

Год набора **2017**

Рыбница 2020

Рабочая программа дисциплины *«Программирование на языке высокого уровня»*/сост. А.Б. Глазов. – Рыбница: Рыбницкий филиал ПГУ им. Т.Г. Шевченко, 2020. – 8 с.

РАБОЧАЯ ПРОГРАММА ПРЕДНАЗНАЧЕНА ДЛЯ ПРЕПОДАВАНИЯ БАЗОВОЙ ЧАСТИ БЛОКА ДИСЦИПЛИН (МОДУЛЕЙ) СТУДЕНТАМ ОЧНОЙ ФОРМЫ ОБУЧЕНИЯ ПО НАПРАВЛЕНИЮ 09.03.04 – *«ПРОГРАММНАЯ ИНЖЕНЕРИЯ»*, ПРОФИЛЬ ПОДГОТОВКИ *«РАЗРАБОТКА ПРОГРАММНО ИНФОРМАЦИОННЫХ СИСТЕМ»*.

Рабочая программа составлена с учетом Федерального Государственного образовательного стандарта высшего образования по направлению подготовки 09.03.04 – *«Программная инженерия»*, утвержденного приказом Министерства образования и науки Российской Федерации № 229 от 12 марта 2015 года.

Составитель  А.Б. Глазов, ст. преподаватель



1. Цели и задачи освоения дисциплины

Целью освоения дисциплины «Программирование на языке высокого уровня» является освоение теоретических и практических основ программирования на языке высокого уровня.

Задачами дисциплины являются:

- приобретение знаний о принципах организации, структурах технических и программных средств, используемых в программировании;
- приобретение навыков практического использования инструментальных средств программирования в информационных системах.

2. Место дисциплины в структуре ООП ВО

Дисциплина «Программирование на языке высокого уровня» относится к базовой части блока дисциплин (модулей) (Б1.Б.20) для студентов очной формы обучения направления подготовки 09.03.04 «Программная инженерия».

Изучаемая дисциплина связана со следующими дисциплинами, изучаемыми студентами «Исследование операций», «Логическое и функциональное программирование», «Разработка прикладных программных решений».

3. Требования к результатам освоения дисциплины

Изучение дисциплины направлено на формирование следующих компетенций:

Код компетенции	Формулировка компетенции
ПК-1	готовность применять основные методы и инструменты разработки программного обеспечения
ПК-3	владение навыками использования различных технологий разработки программного обеспечения

В результате освоения дисциплины студент должен:

Знать:

- формальные методы, технологии и инструменты разработки программного продукта;
- концепции эволюционного развития программного обеспечения;
- концепции и реализации программных процессов.

Уметь:

- устанавливать, тестировать, испытывать и использовать программные средства;
- конструировать программное обеспечение;
- разрабатывать основные программные документы;
- работать с современными системами программирования.

Владеть:

- навыками разработки и отладки программ на языках высокого уровня.

4. Структура и содержание дисциплины

4.1. Распределение трудоемкости в з.е./часах по видам аудиторной и самостоятельной работы студентов по семестрам

Семестр	Трудоемк ость, з.е./часы	Количество часов					Форма итогового контроля
		В том числе					
		Аудиторных				Самост. работа	
		Всего	Лекций	Лаб. раб.	Практич. зан.		
7	5/180	72	18	36	18	72	Экзамен (36)
Итого:	5/180	72	18	36	18	72	Экзамен (36)

4.2. Распределение видов учебной работы и их трудоемкости по разделам дисциплины

№ раз-дела	Наименование разделов	Количество часов				
		Всего	Аудиторная работа			Внеауд. работа (СР)
			Л	ПЗ	ЛР	
1	Обзор языков программирования	20	2	-	2	16
2	Базовые типы данных	48	4	4	26	14
3	Сложные типы данных	24	6	2	2	14
4	Работа с файлами	34	4	6	4	20
5	Обработка классов	18	2	6	2	8
Итого:		144	18	18	36	72

4.3. Тематический план по видам учебной деятельности

Лекции

№ п/п	Номер раздела дисциплины	Объем часов	Тема лекции	Учебно-наглядные пособия
1	1	2	Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования	лекционная аудитория, оборудованная мультимедийными средствами
2	2	2	Python как базовый язык для изучения основных свойств языков высокого уровня	
3		2	Классы в Python, создание, наследование, свойства и методы	
4	3	2	Специализированные библиотеки как основа расширения функциональности Python	
5		2	Модульное программирование – основа построения больших проектов	
6		2	Области видимости переменных в зависимости от их создания	
7	4	2	Регулярные выражения в Python, примеры практического использования	
8		2	Парсинг сайтов. Библиотеки request, BeautifulSoup	
9	5	2	Управление внешними приложениями средствами Python, Обмен данными и отправка команд	
Итого:		18		

Практические работы

№ п/п	Номер раздела дисциплины	Объем часов	Тема практического занятия	Учебно-наглядные пособия
1	2	2	Среда проектирования. Компиляторы и интерпретаторы. Жизненный цикл программы.	аудитория, оборудованная мультимедийными средствами
2		2	Операторы, операции, структура программы, примеры программ, запуск программы.	
3	3	2	Среда IDLE для отладки отдельных выражений и написания программ на Python	
4	4	2	Программа просмотра исходников Python с возможным использованием иерархической системы комментариев	
5		2	Работа с различными кодировками текстовых данных	
6		2	Использование переменных типа datetime для построения программ измерения времени. Таймер	

№ п/п	Номер раздела дисциплины	Объем часов	Тема практического занятия	Учебно-наглядные пособия
			учета времени	
7	5	2	Управление внешними приложениями средствами Python	
8		2	Многопоточные приложения и их использование в асинхронных программах	
9		2	Элементы работы с базами данных на примере DBSN	
Итого:		18		

Лабораторные работы

№ п/п	Номер раздела дисци- плины	Объем часов	Тема лабораторного занятия	Наимено- вание лабора- тории	Учебно-наглядные пособия
1	1	2	Изучение языка Python: основные типы данных, операторы, программные конструкции, подключение внешних библиотек различного назначения, решение типовых задач	аудитория № 30	Мультимедийный материал
2	2	2	Элементы графической библиотеки Tkinter и их применение для написания программ		
3		2	Canvas как основа для отображения графических элементов		
4		2	Построение графиков функций одной переменной на канве		
5		2	Формирование графических классов для отображения элементов управления на форме		
6		2	Python, работа с различными кодировками текста.		
7		2	Списки на Python, основные методы и свойства переменных типа list		
8		2	Генераторы списков: основы использования, типовые задачи, условное формирование списков		
9		2	Строчные переменные, свойства и методы, типовые задачи		
10		2	Обработка файлов строчных и бинарных, чтение, запись и изменение		
11		2	Множества и основы их использования		
12		2	Тип dict для хранения и использования данных. Основные методы, примеры использования		
13		2	Обработка даты и времени в Python		
14		2	Сжатие данных. Алгоритм Хаффмана		
15	3	2	Python: Подключение специализированных библиотек. Утилита pip		
16	4	2	Обработка звука средствами Python		
17		2	Обработка изображений графические библиотеки Python		
18	5	2	Системные функции. Сырое чтение диска		
Итого:		36			

Самостоятельная работа студента

Раздел дисциплины	№ п/п	Тема и вид СРС	Трудоемкость (в часах)
1	1	Алгоритмы. Свойства алгоритмов. Способы описания алгоритмов. Основные конструкции алгоритмического языка: линейный алгоритм, ветвление, цикл	8
	2	Программа. Программный продукт и его характеристики. Основные этапы решения задач на компьютере	8
2	3	Оператор присваивания. Составной оператор. Условный оператор. Оператор выбора	6
	4	Общие сведения о подпрограммах. Определение и вызов подпрограмм.	8
3	5	Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы	6
	6	Объявление массива. Инициализация. Действия над массивами. Заполнение массива данными. Вывод элементов массива	8
4	7	Поиск, удаление, замена и добавление символов в строке. Операции со строками	6
	8	Понятие множества. Объявление множества. Операции над множествами	6
	9	Типы файлов. Организация доступа к файлам. Файлы последовательного доступа. Открытие и закрытие файла последовательного доступа. Запись в файл и чтение из файла последовательного доступа.	8
5	10	Объявление класса, свойств и методов экземпляра класса. Наследование. Перегрузка методов	8
Итого:			72

5. Примерная тематика курсовых работ (работ)

Курсовые проекты (работы) по дисциплине не предусмотрены.

6. Образовательные технологии

Вид занятия (Л, ПР, ЛР)	Используемые интерактивные образовательные технологии	Количество часов
Л	Лекция-дискуссия.	2
	Ролевая игра	-
ЛР	Работа в малых группах.	2
	Творческие задания.	2
	Использование обучающих Интернет-ресурсов.	2
ПР	Творческие задания.	2
	Использование обучающих Интернет-ресурсов.	2
Итого:		12

7. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

Для оценки качества усвоения курса используются следующие формы контроля:

- **текущий** – контроль выполнения лабораторных и индивидуальных заданий;
- **рубежный** – контроль выполнения лабораторных и практических работ;
- **итоговый** осуществляется посредством экзамена.

Контроль самостоятельной работы студентов осуществляется с помощью ответов на лабораторных и практических занятиях, консультациях.

Вопросы сессионного контроля

1. Алгоритмы. Свойства алгоритмов. Способы описания алгоритмов. Основные конструкции алгоритмического языка: линейный алгоритм, ветвление, цикл.
2. Развитие языков программирования. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования.
3. Среда проектирования. Компиляторы и интерпретаторы. Жизненный цикл программы.
4. Программа. Программный продукт и его характеристики. Основные этапы решения задач на компьютере.
5. Переменные и константы. Объявление объектов данных. Внутреннее представление данных в памяти компьютера.
6. Типы данных. Простые типы данных.
7. Производные типы данных. Структурированные типы данных.
8. Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных.
9. Оператор присваивания. Составной оператор. Условный оператор. Оператор выбора.
10. Цикл с постусловием. Цикл с предусловием.
11. Цикл с параметром. Вложенные циклы.
12. Общие сведения о подпрограммах. Определение и вызов подпрограмм.
13. Область видимости и время жизни переменной. Механизм передачи параметров.
14. Рекурсия. Программирование рекурсивных алгоритмов.
15. Основы структурного программирования. Методы структурного программирования.
16. Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы.
17. Объявление массива. Инициализация. Действия над массивами. Заполнение массива данными. Вывод элементов массива.
18. Удаление и вставка элементов в массив. Обработка массива.
19. Символьный и строковый типы. Объявление типов. Стандартные функции и процедуры для работы со строками.
20. Поиск, удаление, замена и добавление символов в строке. Операции со строками.
21. Понятие множества. Объявление множества. Операции над множествами.
22. Определение типа записи. Правила работы с записями.
23. Типы файлов. Организация доступа к файлам. Файлы последовательного доступа. Открытие и закрытие файла последовательного доступа. Запись в файл и чтение из файла последовательного доступа.
24. Файлы произвольного доступа. Порядок работы с файлами произвольного доступа. Создание структуры записи. Открытие и закрытие файла произвольного доступа.
25. Запись и считывание из файла произвольного доступа. Использование файла произвольного доступа.
26. Стандартные процедуры и функции для файлов разного типа.
27. Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных.
28. Структуры данных на основе указателей.
29. История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.
30. Событийно-управляемая модель программирования. Компонентно-ориентированный подход. Классы объектов. Компоненты и их свойства.
31. Требования к аппаратным и программным средствам интегрированной среды разработчика. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов.
32. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.
33. Проектирование объектно-ориентированного приложения. Создание интерфейса пользователя.
34. Программирование приложения. Тестирование, отладка приложения. Создание документации.
35. Классы объектно-ориентированного языка программирования: виды, назначение, свойства, методы, события.
36. Объявление класса, свойств и методов экземпляра класса. Наследование. Перегрузка методов.

8. Учебно-методическое и информационное обеспечение дисциплины

8.1. Основная литература

1. Лутц, М. Программирование на Python. Т. 2 / М. Лутц. – М.: Символ, 2016. – 992 с.
2. Лутц, М. Программирование на Python. Т. 1 / М. Лутц. – М.: Символ, 2016. – 992 с.
3. Мэтиз, Э. Изучаем PYTHON. Программирование игр, визуализация данных, веб-приложения / Э. Мэтиз. – СПб.: Питер, 2017. – 496 с.
4. Саммерфилд, М. Программирование на Python 3. Подробное руководство / М. Саммерфилд. – М.: Символ, 2016. – 608 с.

8.2. Дополнительная литература

1. Лутц, М. Программирование на Python, II том / М. Лутц. – СПб.: Символ-плюс, 2015. – 992 с.
2. Лутц, М. Программирование на Python, I том / М. Лутц. – СПб.: Символ-плюс, 2015. – 992 с.
3. МакГрат, М. Программирование на Python для начинающих / М. МакГрат. – М.: Эксмо, 2015. – 192 с.
4. Мэтиз, Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения ч2 / Э. Мэтиз. – СПб.: Питер, 2017. – 320 с.

8.3. Программное обеспечение и Интернет-ресурсы

Программное обеспечение, необходимое для проведения лекций-визуализаций: пакет *Python 3.4.*

Интернет-ресурсы:

1. py001 Питон - практический старт. [Электронный ресурс] / Режим доступа: https://www.youtube.com/watch?v=K3hE5C2kgII&list=PLku9se_HAVOp8pSgrIWGIjHLP-EOW_ywq
2. py002 Питон - инженерный калькулятор. [Электронный ресурс] / Режим доступа: https://www.youtube.com/watch?v=K3hE5C2kgII&list=PLku9se_HAVOp8pSgrIWGIjHLP-EOW_ywq

8.4. Методические указания и материалы по видам занятий

Методические указания по выполнению лабораторных и практических работ предоставляются студентам в виде электронных и печатных пособий к лабораторным и практическим работам.

9. Материально-техническое обеспечение дисциплины

Для осуществления образовательного процесса по дисциплине «Программирование на языке высокого уровня» необходима лекционная аудитория, оборудованная мультимедийными средствами для проведения лекций-визуализаций. Для проведения лабораторных занятий используется компьютерная аудитория, оснащенная компьютерами, объединенными локальной сетью, с доступом в Интернет.

10. Методические рекомендации по организации изучения дисциплины

Рабочая учебная программа по дисциплине «Программирование на языке высокого уровня» составлена в соответствии с требованиями Федерального Государственного образовательного стандарта ВО по направлению подготовки 09.03.04 «Программная инженерия» и учебного плана по профилю подготовки «Разработка программно-информационных систем».

Изучение дисциплины «Программирование на языке высокого уровня» включает лекционные, практические и лабораторные занятия. Во время выполнения заданий лабораторной работы в учебной аудитории студент может консультироваться с преподавателем, определять наиболее эффективные методы решения поставленных задач. Если какая-то часть задания остается не выполненной, студент может продолжить её выполнение во время внеаудиторной самостоятельной работы.

Работа с информационными источниками считается одним из основных видов самостоятельной работы.

Текущий контроль усвоения знаний по дисциплине предполагает использование разных форм контроля, в том числе проверки выполнения лабораторных заданий. Итоговый контроль осуществляется в форме зачета. Вопросы к зачету приведены. Выполнение лабораторных заданий является необходимым условием для допуска к зачету.

11. Технологическая карта дисциплины

Кредитно-модульная система оценивания по дисциплине не предусмотрена.

Составитель _____ А.Б. Глазов, ст. преподаватель

Зав. кафедрой информатики и программной инженерии _____ А. Тягульская, доцент