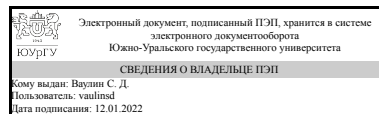


УТВЕРЖДАЮ:  
Директор института  
Политехнический институт



С. Д. Ваулин

## РАБОЧАЯ ПРОГРАММА

дисциплины В.1.04 Объектно-ориентированное программирование для направления 15.03.04 Автоматизация технологических процессов и производств

уровень бакалавр тип программы Академический бакалавриат

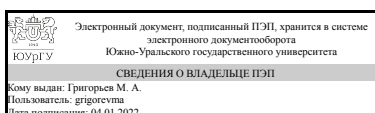
профиль подготовки Автоматизация технологических процессов в промышленности

форма обучения очная

кафедра-разработчик Электропривод и мехатроника

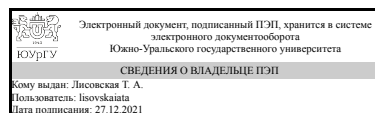
Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 15.03.04 Автоматизация технологических процессов и производств, утверждённым приказом Минобрнауки от 12.03.2015 № 200

Зав.кафедрой разработчика,  
д.техн.н., проф.



М. А. Григорьев

Разработчик программы,  
старший преподаватель



Т. А. Лисовская

## 1. Цели и задачи дисциплины

Цель освоения дисциплины "Объектно-ориентированное программирование" состоит в развитии у студентов базовых навыков программирования на языке высокого уровня (Python) в объектно-ориентированной парадигме программирования, создания объектно-ориентированных программ и алгоритмизации. Задачами курса является формирование систематизированного представления о концепциях, моделях и принципах организации в объектно-ориентированной парадигме, о современном состоянии и перспективных направлениях развития программирования, выработка практических навыков в области выбора и применения технологий программирования для задач автоматизации обработки информации. Целью курса является изучение основных понятий и принципов объектно-ориентированного программирования (ООП), алгоритмизации, а также получение практических навыков программирования на высокоуровневом языке.

## Краткое содержание дисциплины

В курсе рассмотрены концепции объектно-ориентированного программирования на примере языка программирования Python, основные типы и структуры данных, операции с ними, циклы, функции и классы: создание, экземпляры, наследование, инкапсуляция. В процессе освоения дисциплины практические навыки будут формироваться за счёт выполнения практических работ. В течении семестра студенты выполняют семестровую работу. Вид промежуточной аттестации - дифференцированный зачёт.

## 2. Компетенции обучающегося, формируемые в результате освоения дисциплины

Планируемые результаты освоения ОП ВО (компетенции)	Планируемые результаты обучения по дисциплине (ЗУНы)
ПК-8 способностью выполнять работы по автоматизации технологических процессов и производств, их обеспечению средствами автоматизации и управления, готовностью использовать современные методы и средства автоматизации, контроля, диагностики, испытаний и управления процессами, жизненным циклом продукции и ее качеством	Знать: Методы объектно-ориентированного программирования
	Уметь: Разрабатывать программный код в парадигме объектно-ориентированного программирования
	Владеть: технологиями объектно-ориентированного программирования на языке Python

## 3. Место дисциплины в структуре ОП ВО

Перечень предшествующих дисциплин, видов работ учебного плана	Перечень последующих дисциплин, видов работ
Б.1.13 Информатика и программирование	В.1.05 Компьютерное зрение

Требования к «входным» знаниям, умениям, навыкам студента, необходимым при освоении данной дисциплины и приобретенным в результате освоения предшествующих дисциплин:

Дисциплина	Требования
Б.1.13 Информатика и программирование	Знания: принцип простейшей алгоритмизации на естественном языке; Умения: корректно осуществлять постановку задачи программирования; Навыки: построения блок-схем.

#### 4. Объём и виды учебной работы

Общая трудоёмкость дисциплины составляет 4 з.е., 144 ч.

Вид учебной работы	Всего часов	Распределение по семестрам в часах	
		Номер семестра	
		5	
Общая трудоёмкость дисциплины	144	144	
<i>Аудиторные занятия:</i>	64	64	
Лекции (Л)	16	16	
Практические занятия, семинары и (или) другие виды аудиторных занятий (ПЗ)	48	48	
Лабораторные работы (ЛР)	0	0	
<i>Самостоятельная работа (СРС)</i>	80	80	
Выполнение семестрового задания	50	50	
Подготовка к диф. зачету	9	9	
Подготовка к контрольной работе № 1	7	7	
Подготовка к контрольной работе № 3	7	7	
Подготовка к контрольной работе № 2	7	7	
Вид итогового контроля (зачет, диф.зачет, экзамен)	-	диф.зачет	

#### 5. Содержание дисциплины

№ раздела	Наименование разделов дисциплины	Объем аудиторных занятий по видам в часах			
		Всего	Л	ПЗ	ЛР
1	Основные понятия	16	6	10	0
2	Циклы, функции и классы	32	6	26	0
3	Алгоритмизация и программирование	16	4	12	0

##### 5.1. Лекции

№ лекции	№ раздела	Наименование или краткое содержание лекционного занятия	Кол-во часов
1	1	Рекомендуемая литература. Начало работы на языке Python. Подготовка среды программирования.	2
2	1	Переменные и простые типы данных	2
3	1	Списки. Работа со списками	2
4	2	Циклы if и while. Словари	2
5	2	Функции	2
6	2	Классы	2

7	3	Проекты. Планирование проекта. Визуализация данных	2
8	3	Примеры использования шаблонов проектов. Разбор распространенных ошибок	2

## 5.2. Практические занятия, семинары

№ занятия	№ раздела	Наименование или краткое содержание практического занятия, семинара	Кол-во часов
1	1	Инсталляция Python и интегрированной среды разработки (IDE). Обзор библиотек. Работа в терминале	2
2	1	Знакомство с типами данных и простейшими командами.	2
3	1	Философия Python. Стиль программирования. Структура программы. Комментарии.	2
4	1	Списки	2
5	1	Числовые списки. Кортежи	2
6	2	Цикл if. Команда if в работе со списками	2
7	2	Словари. Простой словарь. Работы со словарями	2
8	2	While в работе со списками и словарями. Примеры программ	2
9	2	Практическое программирование. Выполнение упражнений по пройденному материалу	2
10	2	Контрольная работа № 1. Циклы и простые типы данных.	2
11	2	Функции. Определение. Передача аргументов. Передача списка	2
12	2	Практическое программирование. Использование циклов в структуре программы	2
13	2	Классы. Создание и использование класса. Работа с классами и экземплярами. Оформление классов	2
14	2	Контрольная работа № 2. Функции и классы	2
15	2	Файлы и исключения. Импорт, экспорт данных.	2
16	2	Тестирование и отладка кода. Тестирование функций и тестирование классов	2
17	2	Дополнительные возможности Python	2
18	2	Визуализация данных	2
19	3	Контрольная работа № 3. Проект на Python	2
20	3	Основы алгоритмизации	2
21	3	Планирование и анализ проекта	2
22	3	Разбор распространенных ошибок при составлении алгоритма	2
23	3	Оптимизация алгоритмов	2
24	3	Презентация и защита проектов, разработанных в рамках семестрового задания	2

## 5.3. Лабораторные работы

Не предусмотрены

## 5.4. Самостоятельная работа студента

Выполнение СРС		
Вид работы и содержание задания	Список литературы (с указанием разделов, глав, страниц)	Кол-во часов
Подготовка к контрольным работам	Дополнительная литература: [1] с. 19-499	21

Выполнение семестрового задания	Основная литература: [1] с. 37-50; Дополнительная литература: [1] с. 31-80, с. 346-476; Методическое пособие к семестровой работе по дисциплине "Объектно-ориентированное программирование" с. 2-20.	50
Подготовка к зачету	Основная литература: [1] с. 37-50; Дополнительная литература: [1] с. 19-499	9

## 6. Инновационные образовательные технологии, используемые в учебном процессе

Инновационные формы учебных занятий	Вид работы (Л, ПЗ, ЛР)	Краткое описание	Кол-во ауд. часов
Проблемная лекция	Лекции	Проблемная лекция начинается с вопросов, с постановки проблемы, которую в ходе изложения материала необходимо решить. При этом выдвигаемая проблема требует не однотипного решения, готовой схемы которого нет. Данный тип лекции строится таким образом, что деятельность студента по её усвоению приближается к поисковой, исследовательской. На подобных лекциях обязателен диалог преподавателя и студентов.	6

## Собственные инновационные способы и методы, используемые в образовательном процессе

Не предусмотрены

Использование результатов научных исследований, проводимых университетом, в рамках данной дисциплины: нет

## 7. Фонд оценочных средств (ФОС) для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

### 7.1. Паспорт фонда оценочных средств

Наименование разделов дисциплины	Контролируемая компетенция ЗУНы	Вид контроля (включая текущий)	№№ заданий
Все разделы	ПК-8 способностью выполнять работы по автоматизации технологических процессов и производств, их обеспечению средствами автоматизации и управления, готовностью использовать современные методы и средства автоматизации, контроля, диагностики, испытаний и управления процессами, жизненным циклом продукции и ее качеством	Текущий контроль (семестровая работа)	1-7
Все разделы	ПК-8 способностью выполнять работы по автоматизации технологических процессов и производств, их обеспечению средствами автоматизации и управления, готовностью	Промежуточный контроль (диф. зачет)	1-53

	использовать современные методы и средства автоматизации, контроля, диагностики, испытаний и управления процессами, жизненным циклом продукции и ее качеством		
Все разделы	ПК-8 способностью выполнять работы по автоматизации технологических процессов и производств, их обеспечению средствами автоматизации и управления, готовностью использовать современные методы и средства автоматизации, контроля, диагностики, испытаний и управления процессами, жизненным циклом продукции и ее качеством	Текущий контроль (контрольная работа)	1-3

## 7.2. Виды контроля, процедуры проведения, критерии оценивания

Вид контроля	Процедуры проведения и оценивания	Критерии оценивания
Текущий контроль (семестровая работа)	Обучающий в течении семестра самостоятельно выполняет ряд заданий по темам 5-го семестра. Задание на семестровую работу должно быть выдано не позднее 2-й академической недели семестра. Обучающийся сдает на проверку семестровую работу преподавателю на 15 неделе. Семестровая работа оценивается по 100 бальной шкале, каждое задание имеет индивидуальный вес в баллах.	Отлично: Набрано более 85 баллов Хорошо: Набрано от 75 до 84 баллов Удовлетворительно: Набрано от 60 до 74 баллов Неудовлетворительно: Набрано менее 59 баллов
Текущий контроль (контрольная работа)	Контрольная работа проводится по завершению раздела дисциплины. Задание предполагает написание кода программы в соответствии с заданием. Работа оценивается зачтено/не зачтено	Зачтено: Код программы работает исправно и в соответствии с заданием Не зачтено: Код программы не работает и/или не соответствует заданию
Промежуточный контроль (диф. зачет)	К диф. зачету допускаются студенты получившие "зачтено" за все контрольные работы. Диф. зачет проводится в устной форме. Каждому студенту выдается билет, в котором присутствует по пять теоретических вопроса из любого раздела дисциплины. При неправильном ответе студенту могут быть заданы уточняющие или новые вопросы по той же теме. Тема считается освоенной, если студент смог ответить на 60% вопросов, заданных по данной теме.	Отлично: Студент должен ответить на более 85% заданных вопросов, наиболее полно раскрыть содержание материала в объеме программы дисциплины, четко и правильно дать необходимые определения, привести доказательства, показать навыки решения стандартных задач в области электроники. Ответ должен быть самостоятельным, при ответе использованы приобретенные ранее знания. Хорошо: Студент должен ответить от 75% до 84% заданных вопросов, раскрыть содержание материала в объеме программы дисциплины, в основном правильно дать основные определения и понятия предмета. При ответе могут быть допущены неточности, нарушения последовательности изложения, а также могут быть небольшие неточности при выводах и использовании

		<p>терминов, практические навыки нетвёрдые.</p> <p>Удовлетворительно: Студент должен ответить от 60% до 74% заданных вопросов, усвоить основное содержание материала в объеме программы дисциплины. При ответе определения и понятия даны не чётко, допущены ошибки в выводах, практические навыки слабые.</p> <p>Неудовлетворительно: Студент ответил менее чем на 59% заданных вопросов, не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач, основное содержание учебного материала не раскрыто. При ответе допущены грубые ошибки в определениях, не даны ответы на дополнительные вопросы преподавателя, отсутствуют навыки решения стандартных задач.</p>
--	--	---

### 7.3. Типовые контрольные задания

Вид контроля	Типовые контрольные задания
Текущий контроль (семестровая работа)	<p>Типовые вопросы к семестровой работе:</p> <ol style="list-style-type: none"> <li>1. Напишите программу демонстрирующую работу с разными типами данных и сообщениями пользователю.</li> <li>2. Напишите программу выводящую приветственное сообщения, содержащее ваше Имя и Фамилию, начинающиеся с заглавных букв и знаки препинания. Продемонстрируйте работу с регистрами.</li> <li>3. Напишите программу в которой будет создан список и продемонстрируйте различные варианты добавления, удаления и обращения к его элементам.</li> <li>4. Напишите программу в которой будут демонстрироваться возможности сортировки списков.</li> <li>5. Напишите программу демонстрирующую работу срезов.</li> <li>6. Напишите программу содержащую функцию.</li> <li>7. Напишите программу демонстрирующую работу с классами.</li> </ol>
Текущий контроль (контрольная работа)	<p>Вопросы к контрольной №1.</p> <ol style="list-style-type: none"> <li>1. Типы данных, операции, операторы.</li> <li>2. Особенности ввода/вывода.</li> <li>3. Рекурсия, передача и возврат функций в качестве параметров, обработка последовательностей и итераторы.</li> </ol> <p>Вопросы к контрольной работе №2</p> <ol style="list-style-type: none"> <li>1. Встроенные типы объектов: числа, строки, кортежи, списки, словари, множества. Решение задач по теме с применением этих объектов.</li> <li>2. Файловый ввод-вывод.</li> <li>3. Чтение строк с помощью итераторов файлов. Работа с двоичными файлами.</li> </ol> <p>Вопросы к контрольной работе №3</p> <ol style="list-style-type: none"> <li>1. Библиотека numpy для реализации математических объектов и вычислений. Использование объектов array и matrix. Решение задач матричной алгебры.</li> </ol>

	<p>2. Классы в Python. Определение данных, методов, операций.  3. Наследование. Множественное наследование. Композиция при разработке классов.</p>
<p>Промежуточный контроль (диф. зачет)</p>	<ol style="list-style-type: none"> <li>1. Типы данных.</li> <li>2. Переменные.</li> <li>3. Числовые типы данных.</li> <li>4. Операции над числовыми типами данных.</li> <li>5. Строки.</li> <li>6. Строки unicod.</li> <li>7. Вывод данных.</li> <li>8. Ввод данных.</li> <li>9. Форматированный ввод/вывод.</li> <li>10.Списки.</li> <li>11.Выражения в списках.</li> <li>12.Оператор del.</li> <li>13.Использование списков, как стеков.</li> <li>14.Использование списков, как очередей.</li> <li>15.Операции сравнения для списков.</li> <li>16.Диапазоны.</li> <li>17.Кортежи. Отличие кортежей от словарей.</li> <li>18.Словари.</li> <li>19.Оператор if. Особенности операторов сравнения.</li> <li>20.Операторы цикла. Оператор for. Оператор while.Завершение цикла.</li> <li>21.Продолжение цикла.Оператор pass.</li> <li>22.Определение функции.</li> <li>23.Пространство имен функции.</li> <li>24.Передача параметров. Ключи.</li> <li>25.Передача в функцию переменного числа аргументов.</li> <li>26.Элементы функционального программирования.</li> <li>27.Использование лямбда функций.</li> <li>28.Функции работы со структурами данных.</li> <li>29.Функция map(). Примеры применения</li> <li>30.Функция filter().Примеры применения</li> <li>31.Функция reduce().Примеры применения</li> <li>32.Документирование функций.</li> <li>33.Создание модулей.</li> <li>34.Указание кодировки.</li> <li>35.Поиск модулей.</li> <li>36.Компиляция модулей на Python.</li> <li>37.Стандартные модули Python.</li> <li>38.Использование функции <code>__dir()</code>.</li> <li>39.Структурирование модулей в пакеты.40.Импорт модулей и их составляющих из пакета.</li> <li>41.Ссылки в пакетах.</li> <li>42.Пакеты и файловая система.</li> <li>43.Класс File.</li> <li>44.Открытие файла.</li> <li>45.Методы класса для File ввода-вывода.</li> <li>46.Взаимодействие с файловой системой.</li> <li>47.Модуль path.</li> <li>48.Объекты и файловый ввод-вывод.</li> <li>49.Объявление класса</li> <li>50.Управление атрибутами и методами класса</li> <li>51.Объявление объектов</li> <li>52.Множественное наследование</li> <li>53.Заемствование свойств и методов у родительского класса</li> </ol>



## 8. Учебно-методическое и информационное обеспечение дисциплины

### Печатная учебно-методическая документация

#### а) основная литература:

1. Иванова, Г. С. Объектно-ориентированное программирование Учеб. для вузов по направлению подгот. дипломир. специалистов "Информатика и вычисл. техника" Г. С. Иванова, Т. Н. Ничушкина, Е. К. Пугачев; Под ред. Г. С. Ивановой. - 2-е изд., перераб. и доп. - М.: Издательство МГТУ им. Н. Э. Баумана, 2003. - 367 с. ил.

#### б) дополнительная литература:

Не предусмотрена

#### в) отечественные и зарубежные журналы по дисциплине, имеющиеся в библиотеке:

Не предусмотрены

#### г) методические указания для студентов по освоению дисциплины:

1. Методическое пособие к семестровой работе по дисциплине "Объектно-ориентированное программирование"

#### из них: учебно-методическое обеспечение самостоятельной работы студента:

1. Методическое пособие к семестровой работе по дисциплине "Объектно-ориентированное программирование"

### Электронная учебно-методическая документация

Нет

## 9. Информационные технологии, используемые при осуществлении образовательного процесса

Перечень используемого программного обеспечения:

1. Microsoft-Office(бессрочно)
2. -Python(бессрочно)

Перечень используемых информационных справочных систем:

Нет

## 10. Материально-техническое обеспечение дисциплины

Вид занятий	№ ауд.	Основное оборудование, стенды, макеты, компьютерная техника, предустановленное программное обеспечение, используемое для различных видов занятий
Лекции	812-2 (3б)	Мультимедийное оборудование: проектор, интерактивная доска, персональный компьютер с предустановленной ПО.
Практические занятия и семинары	812-2 (3б)	Мультимедийное оборудование: проектор, интерактивная доска, персональный компьютер с предустановленной ПО.