


**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«Воронежский государственный технический университет»



УТВЕРЖДАЮ

Декан факультета  Ряжских В.И.  
«31» августа 2019 г.

**РАБОЧАЯ ПРОГРАММА  
дисциплины (модуля)**

«Конструкторско-технологическая информатика»

Направление подготовки 15.03.05 – Конструкторско-технологическое  
обеспечение машиностроительных производств

Профиль Технология машиностроения

Квалификация выпускника Бакалавр

Нормативный период обучения 4 года / 4 г. и 11 м.

Форма обучения Очная / Заочная

Год начала подготовки 2019 г.

Авторы программы



/ Пачевский Д.Е. /  
/Черных Д.М. /

Заведующий кафедрой  
автоматизированного оборудования  
машиностроительного производства



/ Петренко В.Р. /

Руководитель ОПОП



/ Смоленцев Е.В. /

**Воронеж 2019**

# 1 ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

## 1.1 Цели дисциплины

- изучение способов записи алгоритмов и их реализацию на языке программирования Python;
- изучение технологий создания программного обеспечения на языке программирования Python при решении прикладных задач в области машиностроительного производства.

## 1.2 Задачи освоения дисциплины

- овладеть основами алгоритмизации и разработки программного обеспечения применительно к машиностроительному производству на языке программирования Python;
- применять интегрированные среды разработки для создания программного обеспечения с учетом особенностей технологий функционального и объектно-ориентированного программирования.

## 2 МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Конструкторско-технологическая информатика» относится к обязательным дисциплинам базовой части (Б1.Б.1) блока Б1 учебного плана.

## 3 ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Конструкторско-технологическая информатика» направлен на формирование следующей компетенции:

ОПК-3 – Способность использовать современные информационные технологии, прикладные программные средства при решении задач профессиональной деятельности.

<b>Компетенция</b>	<b>Результаты обучения, характеризующие формирование компетенции</b>
ОПК-3	<b>Знать</b> стандартные программные средства для решения задач в области конструкторско-технологического обеспечения машиностроительных производств.
	<b>Знать</b> типы и структуры данных, используемые в языке программирования Python
	<b>Уметь</b> решать задачи в области конструкторско-технологического обеспечения машиностроительных производств, используя язык программирования Python.
	<b>Владеть</b> навыками разработки и использования современных информационных технологий, прикладных программных средств при решении задач в области конструкторско-технологического обеспечения машиностроительных производств.

#### 4. ОБЪЕМ ДИСЦИПЛИНЫ (МОДУЛЯ)

Общая трудоемкость дисциплины «Конструкторско-технологическая информатика» составляет 4 зачетные единицы.

Распределение трудоемкости дисциплины по видам занятий

##### Очная форма обучения

Вид учебной работы	Всего часов	Семестры			
		2			
<b>Аудиторные занятия (всего)</b>	36	36			
В том числе:					
Лекции	18	18			
Практические занятия (ПЗ)	-	-			
Лабораторные работы (ЛР)	18	18			
<b>Самостоятельная работа</b>	108	108			
Курсовая работа	+	+			
Контрольная работа	-	-			
Вид промежуточной аттестации	Зачет с оценкой	Зачет с оценкой			
Общая трудоемкость, часов	144	144			
Зачетных единиц	4	4			

##### Заочная форма обучения

Вид учебной работы	Всего часов	Семестры			
		2			
<b>Аудиторные занятия (всего)</b>	8	36			
В том числе:					
Лекции	4	18			
Практические занятия (ПЗ)	-	-			
Лабораторные работы (ЛР)	4	18			
<b>Самостоятельная работа</b>	132	108			
Курсовая работа	+	+			
Контрольная работа	-	-			
Вид промежуточной аттестации - зачет с оценкой	4	Зачет с оценкой			
Общая трудоемкость, часов	144	144			
Зачетных единиц	4	4			

## 5 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

### 5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

#### очная форма обучения

№ п/п	Наименование раздела	Содержание раздела	Лекц	Прак зан.	Лаб. зан.	СРС	Всего, час
1	Основы синтаксиса Python	<b>Введение в язык программирования.</b> Интегрированная среда программирования. Концепция присваивания в Python. Динамическая типизация. Именованные переменные. Арифметические операции. Операции сравнения. Условная конструкция if. Самостоятельная работа: <i>Системы счисления. Позиционные и непозиционные системы. Использование двоичной, восьмеричной и шестнадцатеричной систем счисления.</i>	2	-	4	12	18
		<b>Циклы.</b> Основы алгебры логики. Таблицы истинности. Логические операции. Каскадные условные конструкции. Цикл while. Цикл for. Функция range(). Самостоятельная работа: <i>Перевод чисел из одной системы в другую. Арифметические действия в различных системах счисления.</i>	2	-	4	12	18
		<b>Типы данных.</b> Кортежи. Множества. Преобразование типов данных. Тип данных bool. Константы True, False. Логические операции в Python. Самостоятельная работа: <i>Законы логических операций.</i>	2	-	-	12	14
		<b>Списки.</b> Работа со списками. Перебор с помощью цикла for. Методы при работе со списка-	2	-	-	12	14

		ми. Функции при работе со списками. Срезы. Самостоятельная работа: <i>Многомерные списки, Генераторы списков.</i>					
2	Основы программирования на Python.	Функции. Определение функции. Аргументы функции. Аргументы по умолчанию. Переменное число аргументов. Именованные аргументы. Условия именованности. Дополнительные именованные аргументы. Возвращаемые значения. Правила видимости. Инструкция global. Анонимные функции. Самостоятельная работа: <i>Генераторы. Декораторы функций.</i>	2	-	4	12	18
		<b>Работа со строками.</b> Строки. Основные функции и методы строк. Форматирование строк. Оператор %. Спецификаторы и модификаторы. Самостоятельная работа: <i>Словари. Работа со словарями.</i>	2	-	2	12	16
		<b>Введение в объектно-ориентированное программирование.</b> Базовые понятия ООП: объект, атрибуты и методы, класс. Принципы ООП. Инкапсуляция. Наследование. Полиморфизм. Самостоятельная работа: <i>Рекурсия. Факториал числа.</i>	2	-	-	12	14
		<b>Модули.</b> Импорт модулей. Стандартная библиотека Python. Модуль random. Исключения. Обработка исключений. Самостоятельная работа: <i>Файлы. Работа с файлами.</i>	2	-	2	12	16
		<b>Прикладные библиотеки Python.</b> Введение в массивы библиотеки NumPy. Срезы массивов. Изменения формы массивов. Слияние и разбиение массивов. Визуализация с помощью	2	-	2	12	16

		библиотеки Matplotlib. Простые линейные графики. Самостоятельная работа: <i>Пользовательские настройки графиков. Множественные субграфики.</i>					
<b>Итого</b>			<b>18</b>	<b>-</b>	<b>18</b>	<b>108</b>	<b>144</b>

### Заочная форма обучения

№ п/п	Наименование раздела	Содержание раздела	Лекц	Прак зан.	Лаб. зан.	СРС	Всего, час
1	Основы синтаксиса Python	<b>Введение в язык программирования.</b> Интегрированная среда программирования. Концепция присваивания в Python. Динамическая типизация. Именованные переменные. Арифметические операции. Операции сравнения. Условная конструкция if. Самостоятельная работа: <i>Системы счисления. Позиционные и непозиционные системы. Использование двоичной, восьмеричной и шестнадцатеричной систем счисления.</i>	-	-	-	15	15
		<b>Циклы.</b> Основы алгебры логики. Таблицы истинности. Логические операции. Каскадные условные конструкции. Цикл while. Цикл for. Функция range(). Самостоятельная работа: <i>Перевод чисел из одной системы в другую. Арифметические действия в различных системах счисления.</i>	0,5	-	-	15	15,5
		<b>Типы данных.</b> Кортежи. Множества. Преобразование типов данных. Тип данных bool. Константы True, False. Логические операции в Python. Самостоятельная работа: <i>Законы логических операций.</i>	0,5	-	-	14	14,5

		<b>Списки.</b> Работа со списками. Перебор с помощью цикла for. Методы при работе со списками. Функции при работе со списками. Срезы. Самостоятельная работа: <i>Многомерные списки, Генераторы списков.</i>	0,5	-	-	14	14,5
2	Основы программирования на Python.	<b>Функции.</b> Определение функции. Аргументы функции. Аргументы по умолчанию. Переменное число аргументов. Именованные аргументы. Условия именованности. Дополнительные именованные аргументы. Возвращаемые значения. Правила видимости. Инструкция global. Анонимные функции. Самостоятельная работа: <i>Генераторы. Декораторы функций.</i>	0,5	-	2	16	18,5
		<b>Работа со строками.</b> Строки. Основные функции и методы строк. Форматирование строк. Оператор %. Спецификаторы и модификаторы. Самостоятельная работа: <i>Словари. Работа со словарями.</i>	0,5	-	-	16	16,5
		<b>Введение в объектно-ориентированное программирование.</b> Базовые понятия ООП: объект, атрибуты и методы, класс. Принципы ООП. Инкапсуляция. Наследование. Полиморфизм. Самостоятельная работа: <i>Рекурсия. Факториал числа.</i>	0,5	-	-	14	14,5
		<b>Модули.</b> Импорт модулей. Стандартная библиотека Python. Модуль random. Исключения. Обработка исключений. Самостоятельная работа: <i>Файлы. Работа с файлами.</i>	0,5	-	1	14	15,5
		<b>Прикладные библиотеки Python.</b> Введение в массивы библиотеки NumPy. Срезы мас-	0,5	-	1	14	15,5

		сивов. Изменения формы массивов. Слияние и разбиение массивов. Визуализация с помощью библиотеки Matplotlib. Простые линейные графики. Самостоятельная работа: <i>Пользовательские настройки графиков. Множественные субграфики.</i>					
<b>Итого</b>			<b>4</b>	<b>-</b>	<b>4</b>	<b>132</b>	<b>140</b>

## 5.2 Перечень лабораторных работ

1. Интегрированная среда разработки.
2. Алгоритмы на Python.
3. Работа с файлами и файловой системой.
4. Обработка и визуализация данных на языке Python.

## 5.3 Перечень практических работ

Практические работы не предусмотрены.

# 6 ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

## 6.1 Курсовая работа

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсовой работы.

Целью выполнения курсовой работы является закрепление и развитие практических навыков разработки прикладных программ на языке программирования Python.

В курсовой работе обучающиеся используют материалы из справочной литературы, ГОСТов, повышают навыки работы в стандартных программных средах.

Курсовая работа выполняется по типовым заданиям кафедры. Объектами разработки заданий являются элементы конструкторско-технологической подготовки производства. В курсовой работе реализуется автоматизация данных элементов на языке программирования Python.

Примерная тематика курсовой работы: «Автоматизация технологической подготовки производства».

Задачи, решаемые при выполнении курсового проекта:

- проанализировать элемент процесса конструкторско-технологической подготовки производства;
- разработать алгоритм для реализации данного процесса;



- на основе алгоритма разработать программное обеспечение на языке Python;
- провести тестирование и оценить адекватность результатов.

6.2 Контрольные работы для обучающихся заочной формы обучения  
Контрольные работы не предусмотрены учебным планом.

## **7 ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

### **7.1 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания**

#### **7.1.1 Этап текущего контроля**

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

<b>Компетенция</b>	<b>Результаты обучения, характеризующие сформированность компетенции</b>	<b>Критерии оценивания</b>	<b>Аттестован</b>	<b>Не аттестован</b>
ОПК-3	<b>Знать</b> стандартные программные средства для решения задач в области конструкторско-технологического обеспечения машиностроительных производств.	Активная работа на занятиях, отвечает на теоретические вопросы при выполнении лабораторных работ	Выполнение работ в сроки, предусмотренные в рабочей программе	Невыполнение работ в сроки, предусмотренные в рабочей программе
	<b>Знать</b> типы и структуры данных, используемые в языке программирования Python	Активная работа на занятиях, отвечает на теоретические вопросы при выполнении лабораторных работ	Выполнение работ в сроки, предусмотренные в рабочей программе	Невыполнение работ в сроки, предусмотренные в рабочей программе
	<b>Уметь</b> решать задачи в области конструкторско-технологического обеспечения машиностроительных производств, используя язык программирования Python.	Активная работа на занятиях, отвечает на теоретические вопросы при защите лабораторных работ и выполнении курсовой работы	Выполнение работ в сроки, предусмотренные в рабочей программе	Невыполнение работ в сроки, предусмотренные в рабочей программе
	<b>Владеть</b> навыками разработки и использования современных информационных техноло-	Активная работа при защите лабораторных работ и курсовой работы	Выполнение работ в сроки, предусмотренные в	Невыполнение работ в сроки, предусмотренные в

	гий, прикладных программных средств при решении задач в области конструкторско-технологического обеспечения машиностроительных производств.		рабочей программе	рабочей программе
--	---	--	-------------------	-------------------

### 7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля и оценивание для очной формы обучения проводятся перед сессией 2 семестра, для заочной формы обучения на сессии 2 семестра. Формой контроля является зачет с оценкой, по которому выставляются оценки:

«отлично»;

«хорошо»;

«удовлетворительно»;

«неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл	Неудовл
ОПК-3	<b>Знать</b> стандартные программные средства для решения задач в области конструкторско-технологического обеспечения машиностроительных производств.	Тест	Выполнение теста на 90-100 %	Выполнение теста на 80-90 %	Выполнение теста на 60-80 %	В тесте менее 60% правильных ответов
	<b>Знать</b> типы и структуры данных, используемые в языке программирования Python	Тест	Выполнение теста на 90-100 %	Выполнение теста на 80-90 %	Выполнение теста на 60-80 %	В тесте менее 60 % правильных ответов
	<b>Уметь</b> решать задачи в области конструкторско-технологического обеспечения машиностроительных производств, используя язык программирования Python.	Тест	Выполнение теста на 90-100 %	Выполнение теста на 80-90 %	Выполнение теста на 60-80 %	В тесте менее 60 % правильных ответов
	<b>Владеть</b> навыками разработки и использования современных информационных технологий, прикладных программных средств при решении задач в области конструкторско-технологического обеспечения машиностроительных производств.	Тест	Выполнение теста на 90-100 %	Выполнение теста на 80-90 %	Выполнение теста на 60-80 %	В тесте менее 60 % правильных ответов

**7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)**

### 7.2.1 Примерный перечень заданий для подготовки к тестированию

1. Что будет напечатано при исполнении следующего кода? Используется Python 3.x.

```
print(type(5/2))
```

1. type 'int'
2. type 'number'
3. type 'double'
4. type 'tuple'

2. Что выведет следующий фрагмент кода?

```
x = 4.5
```

```
y = 2
```

```
print(x // y)
```

1. 2.0
2. 2.25
3. 9.0
4. False

3. Что будет напечатано?

```
x = True
```

```
y = False
```

```
if not x:
```

```
    print(1)
```

```
elif not x and y:
```

```
    print(2)
```

```
elif not x or not y:
```

```
    print(3)
```

```
else:
```

```
    print(4)
```

1. 1
2. 2
3. 3
4. 4

4. Что будет напечатано?

```
x = "summer sun"
```

```
print("%s" % x[3:5])
```

1. mer
2. mme
3. me
4. Syntax Error

5. Что выведет следующий код?

```
d = lambda p: p * 2
```

```
t = lambda p: p * 3
x = 2
x = d(x)
x = t(x)
x = d(x)
print(x)
```

1. 12
2. 24
3. 36
4. 48

6. Что делает следующий код?

```
def a(b, c, d): pass
```

1. Определяет список и инициализирует его.
2. Определяет функцию, которая ничего не делает.
3. Определяет функцию, которая передает параметры.
4. Определяет пустой класс.

7. Что будет напечатано?

```
dict = {"one", "two", "three", "four"}
print(dict['three'])
```

1. two
2. three
3. four
4. TypeError.

8. Что выведет следующая программа?

```
a = [1,2,3,None(),[],.]
print(len(a))
```

1. 4
2. 5
3. 6
4. 7

9. Что выведет следующий цикл?

```
for i in range(5):
    if i == 3:
        continue
    print(i)
```

1. 0, 1, 2, 3, 4
2. 0, 1, 2, 4
3. 0, 1, 2, 4, 5
4. 1, 2, 3, 4, 5

10. Что выведет следующая программа?

```
a=[0, 1, 2]
```

```
b=a
```

```
b[0]=10
```

```
print(a)
```

```
1. [0, 1, 2]
```

```
2. [10, 1, 2]
```

```
3. [1, 2]
```

```
4. Type Error
```

#### **7.2.4 Примерный перечень практических заданий для подготовки к зачету с оценкой**

1. Найти уравнение прямой, проходящей через две точки.

На вход подаются координаты  $x_1$ ,  $y_1$ ,  $x_2$ ,  $y_2$  через пробел. На выходе должна выводиться функция прямой, проходящей через заданные точки.

```
in: 0 1 3 4
```

```
out:  $y = 1.00*x + 1.00$ 
```

2. Найти площадь полной поверхности цилиндра.

На вход подается радиус цилиндра и его высоты. На выходе имеем площадь полной поверхности цилиндра.

```
in: 1
```

```
in: 10
```

```
out: 69.12
```

3. Определить, в какой четверти находится точка.

На вход подаются координаты точки. На выходе выводится номер четверти, в которой располагается данная точка.

```
in: -1
```

```
in: 10
```

```
out: Точка в II четверти
```

4. Определить количество четных и нечетных цифр в числе.

На вход подается случайное число произвольной длины. На выход выводится количество четных и нечетных цифр, содержащихся в числе.

```
in: 1122
```

```
out: Четных: 2
```

```
out: Нечетных: 2
```

5. Найти максимальную цифру случайного числа.

На вход подается случайное число произвольной длины. На выход выводится самая большая цифра в числе.

```
in: 1123.756
```

*out: 7*

6. Напишите программу, которая считывает целые числа с консоли по одному числу в строке.

Для каждого введённого числа проверить:

если число меньше 10, то пропускаем это число;

если число больше 100, то прекращаем считывать числа.

В остальных случаях вывести это число обратно на консоль в отдельной строке

*in: 55*

*9*

*46*

*111*

*out: 55*

*46*

7. Вычислить площадь треугольника, используя формулу Герона.

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

где  $p = (a+b+c)/2$  – полупериметр треугольника.

На вход программе подаются целые числа, выводом программы должно являться вещественное число, соответствующее площади треугольника

*in: 3*

*4*

*5*

*out: 6*

8. Напишите программу, принимающую на вход целое число, которая выводит `True`, если переданное значение попадает в интервал  $(-15, 12] \cup (14, 17) \cup [19, +\infty)$  и `False` в противном случае (регистр символов имеет значение).

*in: 15*

*out: True*

*in: 17*

*out: False*

9. Напишите программу, которая считывает со стандартного ввода целые числа, по одному числу в строке, и после первого введенного нуля выводит сумму полученных на вход чисел.

*in: 3*

*4*

*5*

*0*

*out: 12*

10. Напишите программу, на вход которой подается одна строка с целыми числами.

Программа должна вывести сумму этих чисел.

*in:* 1 2 3 4 5 6 7 8 9

*out:* 45

### 7.2.5 Примерный перечень вопросов для подготовки к экзамену

Экзамен не предусмотрен.

### 7.2.6 Методика выставления оценки при проведении промежуточной аттестации

Промежуточная аттестация проводится в соответствии с календарным графиком; учебным планом при промежуточной аттестации по дисциплине предусмотрен **зачет с оценкой**.

К промежуточной аттестации допускаются обучающиеся, получившие оценку «зачтено» по выполненным лабораторным работам, защитившие курсовую работу, и получившие оценку по результатам текущей аттестации.

Фонд оценочных средств промежуточной аттестации разработан в форме тестовых заданий по разделам дисциплины, темам выполняемых лабораторных работ и курсовой работе, с помощью которых оценивается степень освоения дисциплины.

После защиты курсовой работы обучающимся выставляются оценки:

- «отлично» – при выполнении всех пунктов курсовой работы;
- «хорошо» – при выполнении только расчетной части и графического плана;
- «удовлетворительно» – при выполнении только расчетной части;
- «неудовлетворительно» – при невыполнении графической и расчетной части.

Промежуточная аттестация проводится путем организации опроса в устной и письменной форме с использованием прикладных программных средств по заданиям, каждое из которых состоит из тестового задания и практического задания. При проведении промежуточной аттестации в форме «зачета с оценкой» преподавателем выставляется оценка:

- «отлично» - при 90-100 % правильных ответов;
- «хорошо» - при 80-90 % правильных ответов;
- «удовлетворительно» - при 60-80 % правильных ответов;
- «неудовлетворительно» - менее 60 % правильных ответов.

### 7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы	Код контролируемой компетенции	Наименование оценочного средства
-------	------------------------	--------------------------------	----------------------------------

	(темы) дисциплины	тенции (или ее части)	
1	Основы синтаксиса Python	ОПК-3	Защита лабораторных работ; защита курсовой работы: оценка; тестовое и практическое задание: опрос, зачет с оценкой.
2	Основы программирования на Python.	ОПК-3	Защита лабораторных работ; защита курсовой работы: оценка; тестовое и практическое задание: опрос, зачет с оценкой.

### **7.3 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Проверка знаний на лабораторных занятиях, которая проводится в форме фронтального устного опроса, фиксируется преподавателем и доводится до сведения каждого обучающегося. Проверка правильности выполнения лабораторной работы, итогом которой является оценка «зачтено» или «не зачтено», характеризует практическую освоенность материала по теме лабораторной работы.

Защита курсовой работы осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного обучающегося составляет 20 мин.

Промежуточная аттестация осуществляется путем выполнения тестового задания и решения практического задания при помощи программного обеспечения на компьютере. Время на выполнение тестового задания - 30 мин. Время на выполнение практического задания – 30 мин. Затем, преподавателем осуществляется проверка тестового и практического задания, и выставляется оценка по методическим материалам выставления оценки при проведении промежуточной аттестации в форме зачета с оценкой.

## **8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

### **8.1 Перечень учебной литературы, необходимой для освоения дисциплины**

#### **8.1.1 Основная литература**

1. Чижов, М.И. Информатика и информационные системы [Электронный ресурс]: конспект лекций по дисциплине «Информатика» / М.И. Чижов, А.Н. Юров. – Электрон. текстовые, граф. дан. – Воронеж: ВГТУ, 2003. 148 с. – 1 диск. – Режим доступа: <http://catalog.vorstu.ru>.

2. Алексеев, А.П. Информатика 2007 [Текст]: учеб. пособие / А.П. Алексеев. – М.: Солон-Пресс, 2007. – 608 с.



3. Острейковский, В.А. Информатика [Текст]: учеб. для вузов / В.А. Острейковский. – М.: Высшая школа, 2007. – 511 с.

4. Жуков, Р.А. Язык программирования Python: практикум: учеб, пособие / Р.А. Жуков. – М.: ИНФРА-М, 2019. – 216с.

### **8.1.2 Дополнительная литература**

1. Плас Дж. Вандер. Python для сложных задач: наука о данных и машинное обучение [Текст]: учебник / Плас Дж. Вандер. – СПб.: Питер, 2018. – 576 с.

2. Бизли Д. Python. Подробный справочник [Текст]: справочник / Бизли Д.; пер. с англ. – СПб.: Символ-Плюс, 2010. – 864 с., ил.

### **8.1.3 Методические разработки**

1. Чижов, М.И. МУ к выполнению лабораторных работ №№ 1-6 по дисциплине «Информатика» [Электронный ресурс] / М.И.Чижов, А.Н. Юров; ГОУВПО «Воронеж гос. техн. ун-т». – Электрон. текстовые, граф. дан. – Воронеж: ВГТУ, 2005. Изд. № 348-05. – 1 диск. – Режим доступа: <http://catalog.vorstu.ru>.

2. Чижов, М.И. МУ к выполнению лабораторных работ «Алгоритмический язык C<sup>++</sup> для создания консольных приложений» по дисциплине «Информатика» для студентов очной и очно-заочной форм обучения [Текст] / М.И.Чижов, А.Н. Юров. – Воронеж: ВГТУ, 2009. – Изд. №144-2009.

3. Демидов, А.В. Методические указания и задания к курсовой работе по дисциплине «Информатика» для студентов направления 15.03.05 «Конструкторско-технологическое обеспечение машиностроительных производств» (все профили) очной формы обучения [Электронный ресурс] / ФГБОУВПО «Воронеж гос. техн. ун-т»; А.В. Демидов – Воронеж: ВГТУ, 2015. – Изд. № 146-2015. – 1 диск. – Режим доступа: <http://catalog.vorstu.ru>.

**8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем**

- 1) Текстовый редактор Microsoft Word
- 2) Python 3.7
- 3) Visual Studio Code

**9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА**

Для проведения лекционных и лабораторных занятий используются следующие аудитории, оснащенные техническими средствами обучения, в том числе и компьютерами:

- компьютерные классы: 312/1 – 10 шт.; 01.6/1 – 15 шт.;
- интерактивная доска 78" SMART board 680i2 со встроенным проектором;
- учебно методические материалы: фильмы, видеоролики, видеофрагменты.

## **10 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)**

По дисциплине «Конструкторско-технологическая информатика» читаются лекции, проводятся лабораторные занятия, выполняется курсовая работа.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Выполнение лабораторных работ направлено на изучение и приобретение практических навыков работы с компьютером, разработки прикладных программ на языке программирования Python; работу с файловой системой, обработкой и визуализацией данных.

Выполнение курсовой работы дает возможность получить навыки использования справочной литературой, ГОСТов, разработки прикладных программ на языке программирования Python для решения задач в области конструкторско-технологического обеспечения машиностроительных производств.

Методика выполнения курсовой работы изложена в учебно-методическом пособии /3/. Поэтапное выполнение курсовой работы проводится своевременно и в установленные сроки.

Большое значение по закреплению и совершенствованию знаний имеет самостоятельная работа студентов над освоением теоретического материала, при подготовке к лабораторным работам и выполнении курсовой работы.

Информацию о планируемой самостоятельной работе над тем или иным материалом студенты получают на занятиях.

Вид учебных занятий	Деятельность обучающегося
Лекция	Написание конспекта лекций: - кратко, схематично, последовательно фиксировать основные положения, формулировки, обобщения, графики и схемы, выводы; - выделять важные мысли, ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, сло-

	<p>варей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, на лабораторной работе.</p>
Лабораторные занятия	<p>Конспектирование рекомендуемых источников. Работа с конспектом лекций, подготовка ответов к контрольным вопросам, просмотр рекомендуемой литературы.</p> <p>При выполнении лабораторных работ применяется метод решения творческой задачи группой студентов, который предлагает ее членам коллективную работу и обсуждение проблем, затем оценку и выбор нужного варианта принятия решения.</p>
Курсовая работа	<p>Перед выполнением курсовой работы, обучающийся должен: ознакомиться с методическими указаниями по ее выполнению, повторить изученный теоретический материал и рекомендованную литературу; уяснить цели и задачи задания; подготовиться и познакомиться с нормативной литературой, собрать из всех источников необходимые материалы, выбрать основные формулы и методики; составить план работы и правильно организовать ее. Для этого целесообразно познакомиться с объяснениями, данными преподавателем к основным типовым и нестандартным заданиям, обратить внимание на наиболее частые заблуждения, разобрать самостоятельно проблемные вопросы, найти ответы и выполнить заданную курсовую работу.</p>
Подготовка к промежуточной аттестации	<p>При промежуточной аттестации по дисциплине необходимо ориентироваться на конспекты лекций, основную и рекомендуемую литературу, выполненные лабораторные работы и курсовую работу.</p> <p>Работа обучающегося при подготовке к текущей и промежуточной аттестации должна включать: изучение учебных вопросов; распределение времени на подготовку; консультирование у преподавателя по трудно усваиваемым вопросам; рассмотрение наиболее сложных из них в дополнительной литературе, или других информационных источниках, предложенных преподавателем.</p>

**АННОТАЦИЯ**  
к рабочей программе дисциплины  
«Конструкторско-технологическая информатика»

**Направление подготовки** 15.03.05 – Конструкторско-технологическое обеспечение машиностроительных производств

**Профиль** Технология машиностроения

**Квалификация выпускника** Бакалавр

**Нормативный период обучения** 4 года / 4 г. и 11 м.

**Форма обучения** Очная / -

**Год начала подготовки** 2018 г.

**Цели дисциплины**

- изучение способов записи алгоритмов и их реализацию на языке программирования Python;
- изучение технологий создания программного обеспечения на языке программирования Python при решении прикладных задач в области машиностроительного производства.

**Задачи освоения дисциплины**

- овладеть основами алгоритмизации и разработки программного обеспечения применительно к машиностроительному производству на языке программирования Python;
- применять интегрированные среды разработки для создания программного обеспечения с учетом особенностей технологий функционального и объектно-ориентированного программирования.

**Перечень формируемых компетенций: ОПК-3.**

ОПК-3 – Способность использовать современные информационные технологии, прикладные программные средства при решении задач профессиональной деятельности.

**Общая трудоемкость дисциплины ЗЕТ: 4.**

**Форма итогового контроля по дисциплине: курсовая работа, зачет с оценкой.**