

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Пекаревский Борис Владимирович  
Должность: Проректор по учебной и методической работе  
Дата подписания: 02.11.2023 13:03:50  
Уникальный программный ключ:  
3b89716a1076b80b2c167df0f27c09d01782ba84



**МИНОБРНАУКИ РОССИИ**  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Санкт-Петербургский государственный технологический институт  
(технический университет)»

УТВЕРЖДАЮ  
Проректор по учебной  
и методической работе  
\_\_\_\_\_ Б.В. Пекаревский  
« 24 » мая 2021 г.

**Рабочая программа дисциплины**  
**АЛГОРИТМИЧЕСКИЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ**  
**ВЫСОКОГО УРОВНЯ**

Направление подготовки

**27.03.03 Системный анализ и управление**

Направленности программы бакалавриата

**«Системный анализ и управление химической технологии»;**  
**«Системный анализ в информационных технологиях»**

Квалификация

**Бакалавр**

Форма обучения

**Очная**

Факультет **информационных технологий и управления**  
Кафедра **системного анализа и информационных технологий**

Санкт-Петербург

2021

**Б1.В.05**

## ЛИСТ СОГЛАСОВАНИЯ

Должность разработчика	Подпись	Ученое звание, фамилия, инициалы
Доцент		А.В. Гайков

Рабочая программа дисциплины «Алгоритмические языки программирования высокого уровня» обсуждена на заседании кафедры системного анализа и информационных технологий

протокол от «28» 04 2021 № 7

Заведующий кафедрой

А.А. Мусаев

Одобрено учебно-методической комиссией факультета информационных технологий и управления

протокол от «19» 05 2021 № 8

Председатель

В.В. Куркина

## СОГЛАСОВАНО

Руководитель направления подготовки «Системный анализ и управление»		Д.А. Краснобородько
Директор библиотеки		Т.Н. Старостенко
Начальник методического отдела учебно-методического управления		Т.И. Богданова
Начальник учебно-методического управления		С.Н. Денисенко

## СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.....	04
2. Место дисциплины (модуля) в структуре образовательной программы.....	05
3. Объем дисциплины .....	05
4. Содержание дисциплины	
4.1. Разделы дисциплины и виды занятий.....	06
4.2. Занятия лекционного типа.....	07
4.3. Занятия семинарского типа.....	08
4.3.1. Семинары, практические занятия .....	08
4.4. Самостоятельная работа.....	10
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.....	11
6. Фонд оценочных средств для проведения промежуточной аттестации.....	11
7. Перечень учебных изданий, необходимых для освоения дисциплины.....	12
8. Перечень электронных образовательных ресурсов, необходимых для освоения дисциплины.....	12
9. Методические указания для обучающихся по освоению дисциплины.....	12
10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине	
10.1. Информационные технологии.....	12
10.2. Программное обеспечение.....	12
10.3. Базы данных и информационно-справочные системы .....	12
11. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине.....	12
12. Особенности освоения дисциплины инвалидами и лицами с ограниченными возможностями здоровья .....	12

Приложения: 1. Фонд оценочных средств для проведения промежуточной аттестации.

## 1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения образовательной программы обучающийся должен овладеть следующими результатами обучения по дисциплине:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения (дескрипторы)
<b>ПК-2</b> Способен разрабатывать требования и проектировать программное обеспечение	<b>ПК-2.5</b> Использование алгоритмических языков высокого уровня при проектировании программного обеспечения	<b>Знать:</b> - основные требования, синтаксис и принципы разработки программного обеспечения (ЗН-1); - основы алгоритмических языков высокого уровня и технологию составления программ (ЗН-2). <b>Уметь:</b> - разрабатывать консольные приложения и приложения с графическим пользовательским интерфейсом (У-1); - создавать программные комплексы на основе различных методов (У-2). <b>Владеть:</b> - навыками разработки приложений (Н-1); - методами и приемами проектирования и разработки программ комплексов и систем (Н-2).

## 2. Место дисциплины в структуре образовательной программы

Дисциплина относится к части формируемой участниками образовательных отношений (Б1.В.05) и изучается на 1 курсе во 1 семестре.

Изучение данной дисциплины базируется на знании студентами основ математики, информатики и основ алгоритмизации в пределах программы средней школы.

Полученные в процессе изучения дисциплины «Алгоритмические языки программирования высокого уровня» знания, умения и навыки могут быть использованы при изучении дисциплин «Интеллектуальный анализ данных», «Объектно-ориентированное программирование», «Программирование на языке Java», а также в научно-исследовательской работе бакалавра и при выполнении выпускной квалификационной работы.

## 3. Объем дисциплины

Вид учебной работы	Всего, академических часов
	Очная форма обучения
<b>Общая трудоемкость дисциплины</b> (зачетных единиц/ академических часов)	<b>4/ 144</b>
<b>Контактная работа с преподавателем:</b>	72
занятия лекционного типа	<b>18</b>
занятия семинарского типа, в т.ч.	54
семинары, практические занятия (в том числе практическая подготовка)	54 (4)
лабораторные работы	
курсовое проектирование (КР или КП)	
КСР	
другие виды контактной работы	
<b>Самостоятельная работа</b>	45
<b>Форма текущего контроля</b> (Кр, реферат, РГР, эссе)	Устный опрос
<b>Форма промежуточной аттестации</b> (КР, КП, зачет, экзамен)	<b>Экзамен (27)</b>

#### 4. Содержание дисциплины

##### 4.1. Разделы дисциплины и виды занятий

№ п/п	Наименование раздела дисциплины	Занятия лекционного типа, академ. часы	Занятия семинарского типа, академ. часы		Самостоятельная работа, академ. часы	Формируемые компетенции	Формируемые индикаторы
			Семинары и/или практические занятия	Лабораторные работы			
1.	Введение в программирование на языке Python.	1			4	ПК-2	ПК-2.5
2	Программирование линейных алгоритмов на языке Python.	1	4		4	ПК-2	ПК-2.5
3.	Программирование разветвляющихся и циклических процессов на языке Python.	1	4		4	ПК-2	ПК-2.5
4.	Операции над строками в языке Python.	1	4		4	ПК-2	ПК-2.5
5.	Структуры данных языка Python.	2	4		4	ПК-2	ПК-2.5
6.	Пользовательские функции в языке Python.	2	6		4	ПК-2	ПК-2.5
7.	Модули и пакеты языка Python	2	10		7	ПК-2	ПК-2.5
8.	Объектно-ориентированное программирование на языке Python	4	10		6	ПК-2	ПК-2.5
9.	Итераторы, контейнеры и перечисления в языке Python	2	6		4	ПК-2	ПК-2.5
10.	Работа с файлами и каталогами в языке Python	2	6		4	ПК-2	ПК-2.5

#### 4.2. Занятия лекционного типа

№ раздела дисциплины	Наименование темы и краткое содержание занятия	Объем, акад. часы	Инновационная форма
1	<u>Введение в программирование на языке Python.</u> Знакомство со средой разработки IDLE. Структура программы. Комментарии. Основные типы данных и операций в языке Python. Основные типы переменных. Преобразование типов переменных.	1	ЛВ
2	<u>Программирование линейных алгоритмов на языке Python.</u> Операторы ввода и вывода данных. Оператор присваивания. Встроенные функции и методы для работы с числами.	1	ЛВ
3	<u>Программирование разветвляющихся и циклических процессов на языке Python.</u> Операции сравнения. Операторы условного перехода. Операторы цикла.	1	ЛВ
4	<u>Операции над строками в языке Python.</u> Строки и двоичные данные. Создание строки. Специальные символы. Операции над строками.	1	ЛВ
5	<u>Структуры данных языка Python.</u> Списки. Создание списка. Операции над списками. Многомерные списки. Перебор элементов списка. Генераторы списков и выражения-генераторы. Функции для работы со списками. Добавление и удаление элементов списка. Поиск элемента в списке и получение сведений о значениях, входящих в список. переворачивание и перемешивание списка. Выбор элемента списка случайным образом.	2	ЛВ
6	<u>Пользовательские функции в языке Python.</u> Определение функции и ее вызов. Необязательные параметры функций и сопоставление по ключам. Переменное число параметров в функции.	2	ЛВ
7	<u>Модули и пакеты языка Python</u> Модули. Понятие модуля. Подключение модуля: инструкции import и from.	2	ЛВ
8	<u>Объектно-ориентированное программирование на языке Python.</u> Понятие класса, атрибута и метода. Определение класса и создание экземпляра класса. Конструкторы и деструкторы. Наследование. Множественное наследование.	4	ЛВ

№ раздела дисциплины	Наименование темы и краткое содержание занятия	Объем, акад. часы	Инновационная форма
9	<u>Итераторы, контейнеры и перечисления в языке Python</u> Итератор класса. Понятие итератора класса и его использование.	2	ЛВ
10	<u>Работа с файлами и каталогами в языке Python.</u> Открытие файла. Методы для работы с файлами. Классы и модули используемые для доступа к файлам.	2	ЛВ

### 4.3. Занятия семинарского типа

#### 4.3.1. Семинары, практические занятия

№ раздела дисциплины	Наименование темы и краткое содержание занятия	Объем, акад. часы		Инновационная форма
		всего	в том числе на практическую подготовку	
2	<u>Программирование линейных алгоритмов на языке Python.</u> Написание арифметических выражений и программная реализация линейных алгоритмов. Модуль Math. Математические функции. Операторы для работы с последовательностью. Приоритет выполнения операторов. Генерация случайных чисел. Модуль random.	4		КтСм
3	<u>Программирование разветвляющихся и циклических процессов на языке Python.</u> Разработка программ, включающих ветвление и циклические процессы. Функции range() и enumerate(). Операторы перехода на следующую итерацию и прерывания цикла. Вложенные циклы.	4		КтСм
4	<u>Операции над строками в языке Python.</u> Разработка программ с использованием регулярных выражений. Форматирование строк. Метод format(). Функции и методы для работы со строками и символами. Регулярные выражения: синтаксис, поиск по шаблону	4		КтСм



№ раздела дисциплины	Наименование темы и краткое содержание занятия	Объем, акад. часы		Инновационная форма
		всего	в том числе на практическую подготовку	
5	<u>Структуры данных языка Python.</u> Работа с массивами. Разработка программ, использующих динамические списки. Сортировка списка. Заполнение списка числами. Преобразование списка в строку. Кортежи. Операции над кортежами. Множества. Операции над множествами. Диапазоны. Операции над диапазонами. Словари. Создание словаря. Операции над словарями. Перебор элементов словаря. Методы для работы над словарями. Генераторы словарей.	4	1	КтСм
6	<u>Пользовательские функции в языке Python.</u> Написание процедур и их использование в программном коде. Анонимные функции. Функции генераторы. Декораторы функций. Глобальные и локальные переменные. Рекурсивные функции. Вложенные функции.	6	1	КтСм
7	<u>Модули и пакеты языка Python</u> Разработка и тестирование модулей. Создание пакетов. Пути поиска модулей. Повторная загрузка модулей. Пакеты. Понятие пакета. Работа с пакетами.	10	1	КтСм
8	<u>Объектно-ориентированное программирование на языке Python.</u> Создание классов. Применение наследования и использование классов при написании программы. Понятие примесей и их использование. Специальные методы классов. Перегрузка операторов. Статические методы и методы классов. Абстрактные методы. Ограничение доступа к идентификаторам внутри класса. Свойства классов. Декораторы классов.	10	1	КтСм
9	<u>Итераторы, контейнеры и перечисления в языке Python.</u> Разработка объектно-ориентированных программ с использованием итераторов, контейнеров и перечислений. Контейнеры. Контейнеры-последовательности. Контейнеры словари. Перечисления. Атрибуты и методы перечислений.	6		КтСм

№ раздела дисциплины	Наименование темы и краткое содержание занятия	Объем, акад. часы		Инновационная форма
		всего	в том числе на практическую подготовку	
10	<u>Работа с файлами и каталогами в языке Python.</u> Создание программ на языке Python, с использованием файлового ввода-вывода. Права доступа к файлам и каталогам. Функции для манипулирования файлами. Преобразование пути к файлам и каталогам. Перенаправление ввода/вывода. Сохранение объектов в файл. Функции для работы с каталогами.	6		КтСм

#### 4.4. Самостоятельная работа обучающихся

№ раздела дисциплины	Перечень вопросов для самостоятельного изучения	Объем, акад. часы	Форма контроля
1	<u>Введение в программирование на языке Python.</u> Знакомство со средой разработки IDLE. Структура программы. Комментарии. Основные типы данных и операций в языке Python.	4	Устный опрос №1
2	<u>Программирование линейных алгоритмов на языке Python.</u> Операторы ввода и вывода данных. Оператор присваивания. Встроенные функции и методы для работы с числами. Модуль Math. Математические функции.	4	Устный опрос №2
3	<u>Программирование разветвляющихся и циклических процессов на языке Python.</u> Операции сравнения. Операторы условного перехода. Операторы цикла. Функции range() и enumerate().	4	Устный опрос №3
4	<u>Операции над строками в языке Python.</u> Строки и двоичные данные. Создание строки. Специальные символы. Операции над строками. Форматирование строк	4	Устный опрос №4

№ раздела дисциплины	Перечень вопросов для самостоятельного изучения	Объем, акад. часы	Форма контроля
5	<u>Структуры данных языка Python.</u> Списки. Создание списка. Операции над списками. Многомерные списки. Перебор элементов списка. Генераторы списков и выражения-генераторы. Функции для работы со списками. Добавление и удаление элементов списка. Поиск элемента в списке и получение сведений о значениях, входящих в список. переворачивание и перемешивание списка. Выбор элемента списка случайным образом. Сортировка списка.	4	Устный опрос №5
6	<u>Пользовательские функции в языке Python.</u> Определение функции и ее вызов. Необязательные параметры функций и сопоставление по ключам. Переменное число параметров в функции. Анонимные функции.	4	Устный опрос №6
7	<u>Модули и пакеты языка Python</u> Модули. Понятие модуля. Подключение модуля: инструкции import и from. Пути поиска модулей. Повторная загрузка модулей.	7	Устный опрос №7
8	<u>Объектно-ориентированное программирование на языке Python</u> Понятие класса, атрибута и метода. Определение класса и создание экземпляра класса. Конструкторы и деструкторы. Наследование. Множественное наследование. Понятие примесей и их использование.	6	Устный опрос №8
9	<u>Итераторы, контейнеры и перечисления в языке Python</u> Итератор класса. Понятие итератора класса и его использование. Контейнеры. Контейнеры- последовательности. Контейнеры словари. Перечисления. Атрибуты и методы перечислений.	4	Устный опрос №9
10	<u>Работа с файлами и каталогами в языке Python.</u> Открытие файла. Методы для работы с файлами. Классы и модули используемые для доступа к файлам. Права доступа к файлам и каталогам.	4	Устный опрос №10

## 5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Методические указания для обучающихся по организации самостоятельной работы по дисциплине, включая перечень тем самостоятельной работы, формы текущего контроля по дисциплине и требования к их выполнению размещены в электронной информационно-образовательной среде СПбГТИ(ТУ) на сайте: <http://media.technolog.edu.ru>

## 6. Фонд оценочных средств для проведения промежуточной аттестации

Промежуточная аттестация по дисциплине проводится в виде экзамена.

Экзамен предусматривает выборочную проверку освоения предусмотренных элементов компетенций и комплектуется вопросами (заданиями) двух видов: теоретические вопросы (для проверки знаний) и комплексная задача (для проверки умений и навыков).

При сдаче экзамена студент получает два вопроса из перечня вопросов и задачу, время подготовки студента к ответу - до 45 мин.

Пример варианта вопросов на экзамене:

**Вариант № 1**

1. Операторы ввода и вывода данных. Оператор присваивания.
2. Наследование. Множественное наследование.
3. Задача

Результаты освоения дисциплины считаются достигнутыми, если для всех элементов компетенций достигнут пороговый уровень освоения компетенции на данном этапе – оценка «удовлетворительно».

## **7. Перечень учебных изданий, необходимых для освоения дисциплины**

### **а) печатные издания:**

1. Халимон, В.И. Основы процедурно-структурного программирования (задачи) : методические указания к выполнению контрольных работ / В. И. Халимон, А. Ю. Рогов, О. В. Проститенко; Министерство образования и науки Российской Федерации, Санкт-Петербургский государственный технологический институт (технический университет), Кафедра системного анализа. - Санкт-Петербург : СПбГТИ (ТУ), 2014. - 32 с
2. Халимон, В.И. Методы объектно-ориентированного программирования (задачи) : методические указания к выполнению контрольных работ / В. И. Халимон, А. Ю. Рогов, О. В. Проститенко; Министерство образования и науки Российской Федерации, Санкт-Петербургский государственный технологический институт (технический университет), Кафедра системного анализа. - Санкт-Петербург : СПбГТИ (ТУ), 2015. - 56 с

### **б) электронные учебные издания:**

1. Халимон, В.И. Основы процедурно-структурного программирования : УЧЕБНЫЕ ПОСОБИЯ / В. И. Халимон, А. Ю. Рогов, О. В. Проститенко ; Министерство образования и науки Российской Федерации, Санкт-Петербургский государственный технологический институт (технический университет), Кафедра системного анализа. - Санкт-Петербург : СПбГТИ (ТУ), 2014. - 104 с // СПбГТИ. Электронная библиотека. – URL: <https://technolog.bibliotech.ru> (дата обращения : 10.03.2021). Режим доступа: для зарегистрированных пользователей.
2. Халимон, В.И. Методы объектно-ориентированного программирования (задачи) : методические указания к выполнению контрольных работ / В. И. Халимон, А. Ю. Рогов, О. В. Проститенко; Министерство образования и науки Российской Федерации, Санкт-Петербургский государственный технологический институт (технический университет), Кафедра системного анализа. - Санкт-Петербург : СПбГТИ (ТУ), 2015. - 56 с // СПбГТИ. Электронная библиотека. – URL: <https://technolog.bibliotech.ru> (дата обращения : 10.03.2021). Режим доступа: для зарегистрированных пользователей.

## **8. Перечень электронных образовательных ресурсов, необходимых для освоения дисциплины**

учебный план, РПД и учебно-методические материалы: <http://media.technolog.edu.ru>

электронно-библиотечные системы:

«Электронный читальный зал – БиблиоТех» <https://technolog.bibliotech.ru/>;

«Лань» <https://e.lanbook.com/books/>.

## **9. Методические указания для обучающихся по освоению дисциплины**

Все виды занятий по дисциплине «Алгоритмические языки программирования высокого уровня» проводятся в соответствии с требованиями следующих СТП:

СТП СПбГТИ 040-02. КС УКДВ. Виды учебных занятий. Лекция. Общие требования;

СТО СПбГТИ 018-2014. КС УКДВ. Виды учебных занятий. Семинары и практические занятия. Общие требования к организации и проведению.

СТП СПбГТИ 048-2009. КС УКДВ. Виды учебных занятий. Самостоятельная планируемая работа студентов. Общие требования к организации и проведению.

СТП СПбГТИ 044-2012. КС УКДВ. Виды учебных занятий. Курсовой проект. Курсовая работа. Общие требования.

Планирование времени, необходимого на изучение данной дисциплины, лучше всего осуществлять на весь семестр, предусматривая при этом регулярное повторение пройденного материала.

Основными условиями правильной организации учебного процесса для студентов является:

плановость в организации учебной работы;

серьезное отношение к изучению материала;

постоянный самоконтроль.

На занятия студент должен приходить, имея знания по уже изученному материалу.

## **10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине**

### **10.1. Информационные технологии**

В учебном процессе по данной дисциплине предусмотрено использование информационных технологий:

чтение лекций с использованием слайд-презентаций;

взаимодействие с обучающимися посредством ЭИОС.

### **10.2. Программное обеспечение**

Программы Microsoft Office (Microsoft Word, Microsoft PowerPoint), интегрированные среды Microsoft Visual Studio, IDLE Python, операционная система MS Windows.

### **10.3 Базы данных и информационно-справочные системы**

Справочно-поисковая система «Консультант-Плюс»

## **11. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине.**

Для ведения лекционных и практических занятий используется компьютерный класс, оснащенный объединенными в сеть персональными компьютерами, оборудованием и техническими средствами обучения на необходимое количество посадочных мест.

## **12. Особенности освоения дисциплины инвалидами и лицами с ограниченными возможностями здоровья.**

Для инвалидов и лиц с ограниченными возможностями учебный процесс осуществляется в соответствии с Положением об организации учебного процесса для обучения инвалидов и лиц с ограниченными возможностями здоровья СПбГТИ(ТУ), утвержденным ректором 28.08.2014г.

**Фонд оценочных средств  
для проведения промежуточной аттестации  
по дисциплине «Алгоритмические языки программирования высокого уровня»**

**1. Перечень компетенций и этапов их формирования.**

Индекс компетенции	Содержание	Этап формирования
ПК-2	Способен разрабатывать требования и проектировать программное обеспечение	начальный

## 2. Показатели и критерии оценивания компетенций на различных этапах их формирования, шкала оценивания

Код и наименование индикатора достижения компетенции	Показатели сформированности (дескрипторы)	Критерий оценивания	УРОВНИ СФОРМИРОВАННОСТИ (описание выраженности дескрипторов)		
			«удовлетворительно» (пороговый)	«хорошо» (средний)	«отлично» (высокий)
ПК-2.5 Использование алгоритмических языков высокого уровня при проектировании программного обеспечения	<b>Рассказывает</b> основные требования, синтаксис и принципы разработки программного обеспечения на языке Python (ЗН-1).	Ответы на вопросы №1 - 57 к экзамену	Путается в синтаксисе языка Python и принципах разработки программного обеспечения на языке Python	Рассказывает синтаксис языка Python и принципы разработки программного обеспечения на языке Python с ошибками	Уверенно и без ошибок рассказывает синтаксис языка Python и принципы разработки программного обеспечения на языке Python
	<b>Объясняет</b> принципы разработки консольных приложений и приложений с графическим пользовательским интерфейсом (У-1).		С ошибками объясняет принципы создания консольных приложений на языке Python	С небольшими погрешностями объясняет принципы создания консольных приложений и приложений с графическим пользовательским интерфейсом на языке Python	Объясняет принципы создания консольных приложений и приложений с графическим пользовательским интерфейсом на языке Python без ошибок, приводит примеры
	<b>Демонстрирует</b> навыки разработки приложений на языке Python (Н-1).		Имеет слабые навыки разработки приложений на языке Python	Демонстрирует навыки разработки приложений на языке Python, но допускает 1-2 ошибки	Демонстрирует уверенные навыки разработки приложений на языке Python

<p><b>Рассказывает</b> основы алгоритмических языков высокого уровня и технологию составления программ (ЗН-2).</p>	<p>Ответы на вопросы №58 - 63 к экзамену</p>	<p>Путается в технологии составления программ</p>	<p>Рассказывает принципы технологии составления программ с ошибками</p>	<p>Уверенно и без ошибок рассказывает принципы технологии составления программ</p>
<p><b>Объясняет</b> принципы создания программных комплексов на основе различных методов (У-2).</p>		<p>С ошибками объясняет принципы создания программных комплексов на основе различных методов</p>	<p>С небольшими погрешностями объясняет принципы создания программных комплексов на основе различных методов</p>	<p>Объясняет принципы создания программных комплексов на основе различных методов без ошибок, приводит примеры</p>
<p><b>Демонстрирует</b> навыки использования методов и приемов проектирования и разработки программ комплексов и систем (Н-2).</p>		<p>Имеет слабые навыки использования методов и приемов проектирования и разработки программ комплексов и систем</p>	<p>Демонстрирует навыки разработки использования методов и приемов проектирования и разработки программ комплексов и систем, но допускает 1-2 ошибки</p>	<p>Демонстрирует уверенные навыки использования методов и приемов проектирования и разработки программ комплексов и систем</p>

Шкала оценивания соответствует СТО СПбГТИ (ТУ).

Шкала оценивания экзамена - балльная («отлично», «хорошо», «удовлетворительно», «неудовлетворительно», «неудовлетворительно»).



### 3. Типовые контрольные задания для проведения промежуточной аттестации

#### а) Вопросы для оценки знаний, умений и навыков, сформированных у студента по компетенции ПК-2:

1. Операторы ввода и вывода данных. Оператор присваивания.
2. Встроенные функции и методы для работы с числами.
3. Модуль Math. Математические функции.
4. Операторы для работы с последовательностью.
5. Приоритет выполнения операторов. Генерация случайных чисел. Модуль random.
6. Операции сравнения. Операторы условного перехода.
7. Операторы перехода на следующую итерацию и прерывания цикла.
8. Функции range() и enumerate().
9. Вложенные циклы.
10. Строки и двоичные данные. Создание строки.
11. Специальные символы. Операции над строками.
12. Форматирование строк. Метод format().
13. Функции и методы для работы со строками и символами.
14. Регулярные выражения: синтаксис, поиск по шаблону.
15. Списки. Создание списка. Операции над списками.
16. Многомерные списки. Перебор элементов списка.
17. Генераторы списков и выражения-генераторы.
18. Функции для работы со списками. Добавление и удаление элементов списка.
19. Добавление и удаление элементов списка.
20. Поиск элемента в списке и получение сведений о значениях, входящих в список. переворачивание и перемешивание списка.
21. Выбор элемента списка случайным образом. Сортировка списка.
22. Заполнение списка числами. Преобразование списка в строку.
23. Кортежи. Операции над кортежами.
24. Множества. Операции над множествами.
25. Диапазоны. Операции над диапазонами.
26. Словари. Создание словаря. Операции над словарями.
27. Методы для работы над словарями. Генераторы словарей.
28. Определение функции и ее вызов. Необязательные параметры функций и сопоставление по ключам.
29. Переменное число параметров в функции.
30. Анонимные функции.
31. Функции-генераторы.
32. Декораторы функций.
33. Глобальные и локальные переменные.
34. Рекурсивные функции. Вложенные функции.
35. Модули. Понятие модуля. Подключение модуля: инструкции import и from.
36. Пути поиска модулей. Повторная загрузка модулей.
37. Пакеты. Понятие пакета. Работа с пакетами.
38. Понятие класса, атрибута и метода.
39. Определение класса и создание экземпляра класса. Конструкторы и деструкторы.
40. Наследование. Множественное наследование.
41. Понятие примесей и их использование.
42. Специальные методы классов. Перегрузка операторов.
43. Статические методы и методы классов.
44. Абстрактные методы.
45. Ограничение доступа к идентификаторам внутри класса.
46. Свойства классов. Декораторы классов.

47. Итератор класса. Понятие итератора класса и его использование.
48. Понятие контейнера. Контейнеры-последовательности. Контейнеры-словари.
49. Перечисления. Атрибуты и методы перечислений.
50. Открытие файла. Методы для работы с файлами.
51. Классы и модули используемые для доступа к файлам.
52. Права доступа к файлам и каталогам.
53. Функции для манипулирования файлами.
54. Python. Преобразование пути к файлам и каталогам.
55. Перенаправление ввода/вывода.
56. Сохранение объектов в файл.
57. Функции для работы с каталогами.
58. Принципы построения графических пользовательских интерфейсов.
59. Основы технологии MVC.
60. Основные парадигмы программирования
61. Принципы разработки информационных систем и программных комплексов
62. Принципы процедурно-структурного программирования.
63. Основные принципы объектно-ориентированного программирования.

При сдаче экзамена, студент получает два вопроса из перечня, приведенного выше и одну задачу.

Пример задачи: Составить программу для определения суммы трех положительных чисел.

Время подготовки студента к устному ответу на вопросы - до 45 мин.

#### **4. Методические материалы для определения процедур оценивания знаний, умений и навыков, характеризующих этапы формирования компетенций.**

Промежуточная аттестация по дисциплине проводится в соответствии с требованиями СПб СТО СПбГТИ(ТУ) 016-2015. КС УКДВ Порядок проведения зачетов и экзаменов.