

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ



УТВЕРЖДАЮ
Декан ЭФ

УТВЕРЖДАЮ /Н.М. Стрельникова/
(Ф.И.О. декана (директора института))

14.02.2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б.1.1.19 Объектно-ориентированное программирование

(код и наименование дисциплины по учебному плану)

Направление подготовки
(специальность)

09.03.03 Прикладная информатика

Квалификация выпускника

Бакалавр

(бакалавр/магистр/специалист)

Направленность

Прикладная информатика в экономике

Курс 2
Семестр 4

Распределение учебного времени

Трудоемкость по учебному плану	144 / 4	часов/зачетных единиц
Лекции	18	часов
Лабораторные работы	36	часов
Практические занятия	-	часов
Иная контактная работа	-	часов
Всего контактной работы (без учета экз.)	54	часов
Контактная работа по экзамену	-	часов
Курсовой проект (работа)	-	семестр
Самостоятельная работа обучающихся (без учета экз.)	90	часов
Самостоятельная работа по подготовке к экзамену	-	часов
Экзамен	-	семестр
Зачет	-	семестр
БРК, ДЗ	4	семестр

(год)

Программа составлена в соответствии с требованиями ФГОС ВО направления подготовки (специальности) 09.03.03 Прикладная информатика

Программу составили:

доцент	ИСЭ	СОГЛАСОВАНО	В.И. Абдулаев
(должность)	(кафедра)		(И.О. Фамилия)

РАССМОТРЕНА и ОДОБРЕНА на заседании кафедры, за которой закреплена дисциплина
Кафедра информационных систем в экономике

(наименование кафедры)		
26.01.2023	протокол №	6
(дата)		

Заведующий кафедрой	СОГЛАСОВАНО	Т.А. Уразаева
		(И.О. Фамилия)

Рабочая программа СОГЛАСОВАНА с факультетом (институтом), выпускающей(ими) кафедрой(ами).

СООТВЕТСТВУЕТ действующей ОП.

Заведующий кафедрой	СОГЛАСОВАНО	Т.А. Уразаева
		(И.О. Фамилия)

Председатель методической комиссии факультета (института), в который входит выпускающая кафедра

СОГЛАСОВАНО	О.Е. Иванов
	(И.О. Фамилия)

Эксперт(ы): Жубрин Алексей Анатольевич, помощник генерального директора ОАО «ММЗ» по информатизации – начальник управления информационных технологий

Рабочая программа проверена и зарегистрирована в УМЦ 22.02.2023 г.

Специалист учебно-методического центра СОГЛАСОВАНО /И.Р. Валиева/

Раздел 1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины является достижение планируемых результатов обучения, соответствующих установленным в ОПОП индикаторам достижения компетенций:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения
1. ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.1. Знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности	знания: ОПК-2.1. - Знать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности умения: навыки:
	ОПК-2.2. Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности	знания: умения: ОПК-2.2. - Уметь выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности навыки:
	ОПК-2.3. Владеет навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности	знания: умения: навыки: ОПК-2.3. - Владеть навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности

<p>2. ОПК-3 Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности</p>	<p>ОПК-3.1. Знает принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности</p>	<p>знания: ОПК-3.1. - Знать принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности</p> <p>умения:</p> <p>навыки:</p>
	<p>ОПК-3.2. Умеет решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности</p>	<p>знания:</p> <p>умения: ОПК-3.2. - Уметь решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности</p> <p>навыки:</p>
	<p>ОПК-3.3. Владеет навыками подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций, и библиографии по научно-исследовательской работе с учетом требований информационной безопасности</p>	<p>знания:</p> <p>умения:</p> <p>навыки: ОПК-3.3. - Владеть навыками подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций, и библиографии по научно-исследовательской работе с учетом требований информационной безопасности</p>

3. ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.1. Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий	знания: ОПК-7.1. - Знать основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий умения: навыки:
	ОПК-7.2. Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ	знания: умения: ОПК-7.2. - Уметь применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ навыки:
	ОПК-7.3. Владеет навыками программирования, отладки и тестирования прототипов программно-технических	знания: умения: навыки: ОПК-7.3. - Владеть навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач

Раздел 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к обязательной части ОПОП.

Дисциплина является обязательной

Для продолжения формирования заявленных компетенций необходимы знания предшествующих дисциплин: Информационные технологии (ОПК-2), Вычислительные системы, сети и телекоммуникации (ОПК-3), Алгоритмизация и программирование (ОПК-7) Изучаемая дисциплина является основой для продолжения формирования указанных компетенций в следующих дисциплинах: Базы данных (ОПК-2), Информационная безопасность (ОПК-3), Программная инженерия (ОПК-7); государственной итоговой аттестации в форме: Выполнение и защита выпускной квалификационной работы (ОПК-2), Выполнение и защита выпускной квалификационной работы (ОПК-7)

Раздел 3. ОПИСАНИЕ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

Для формирования заявленных компетенций используются методологические технологии, реализующие деятельностный, личностно-ориентированный, практико-ориентированный подходы.

Основными стратегическими технологиями являются: игровые процедуры, исследовательские, лекционные занятия, практические и лабораторные занятия. На достижение конкретных целей обучения направлены применяемые тактические технологии: задания, классическая лекция, мини-проекты.

Раздел 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4 семестр

Виды и темы занятий	Количество часов	Формируемые компетенции
Основы процедурного программирования в Питоне	62	ОПК-2, ОПК-3, ОПК-7
Лекция. Простые типы данных, операторы и выражения. Ввод и вывод данных. преобразование типов. Управление потоком команд. Операторы while, if, elif, break, continue	2	
Лекция. Строки. Структуры данных. Списки и словари. Множества.	2	
Лекция. Функции. Параметры функций . Локальные переменные. Зарезервированное слово <<global>>. Зарезервированное слово <<nonlocal>>. Значения параметров по умолчанию. Ключевые параметры. Переменное число параметров varArgs. Только ключевые параметры. Оператор return. Создание собственных функций. Примеры функций.	2	
Лабораторная работа. Изучение простых типов данных. Ввод и вывод через консоль.	2	
Лабораторная работа. Строки. Преобразование типов.	2	
Лабораторная работа. Операторы if, цикл while, цикл for, break, continue	4	
Лабораторная работа. Списки. Вложенные списки	4	
Лабораторная работа. Словари	2	
Лабораторная работа. Контрольная по ЛР 2-5	2	

<p>Задания для самостоятельной работы, в том числе выполнение</p> <p>1. В банк сделали 2 вклада – один вклад 100 000 рублей под 10% годовых, второй 200 000 рублей под 2% годовых. Вывести изменение суммы вкладов по годам до тех пор, пока сумма на первом вкладе не превысит сумму на втором.</p> <p>2. Ежегодный прирост рыбы в пруду составляет 15%. Запасы рыбы оценены в А тонн. Ежегодный план отлова В тонн. Наименьший запас рыбы, ниже которого запас уже не восстанавливается, составляет С тонн. Составить алгоритм и программу, подсчитывающую, сколько лет можно выдержать заданный план.</p> <p>3. Создать функцию <code>bukva_from_s(s)</code> параметром которой является произвольной длины строка <code>s</code> (напр. <code>sdfwertsdlkfjgri</code>). Функция возвращает случайно выбранный символ строки. Создать далее функцию <code>pair()</code>, которая дважды вызывает функцию <code>ch1= bukva_from_s(s)</code> и <code>ch2= bukva_from_s(s)</code> и возвращает строку, составленную в алфавитном порядке из <code>ch1</code> и <code>ch2</code>.</p> <p>4. Дан список, содержащий текстовые элементы (напр., <code>['mix', 'xyz', 'apple', 'xanadu', 'aardvark']</code>). Написать функцию, получающую через аргумент <code>spis</code> заданный список, и возвращающую новый список, в котором строки, начинающиеся на <code>x</code> помещены в начало списка, а остальные размещены в новом списке в алфавитном порядке.</p> <p>5. Создать функцию, получающую как параметры - список слов <code>s</code> и букву <code>bukva</code>. Функция возвращает из списка <code>s</code> слово с наибольшим количеством букв равных <code>bukva</code>.</p>	40	
ООП в Питоне	82	ОПК-2, ОПК-3, ОПК-7
Лекция. Классы и объекты: Специальный параметр <code>self</code> . Классы. Создание и использование классов. Объявление метода. Создание экземпляров класса. Специальные методы <code>init ()</code> и <code>str ()</code> .	2	
Лекция. Поля класса и свойства объекта. Обращение к методу класса. Пространство имен классов и объектов. Локальные и глобальные переменные объекта и класса. Приватные (<code>private</code>) переменные. Метод <code>del</code> .	2	
Лекция. Графика и анимация в Питоне. Импорт внешних библиотек	2	
Лекция. Наследование. Использование наследования в графических приложениях	2	
Лекция. Модули и пакеты в Питоне. Создание модулей с функциями и классами	2	
Лекция. Обзорная лекция	2	
Лабораторная работа. Функции как инструмент повторного использования кода	4	
Лабораторная работа. Классы	2	
Лабораторная работа. Контрольная по темам ЛР 7-10	2	
Лабораторная работа. Классы в графических приложениях	2	
Лабораторная работа. Использование модульной структуры для	2	

графических приложений		
Лабораторная работа. Использование наследования для графических приложений	2	
Лабораторная работа. Мини проекты графических приложений	6	
Задания для самостоятельной работы, в том числе выполнение 1. Составить описание класса для представления комплексных чисел с возможностью задания вещественной и мнимой частей как числами типов double, так и целыми числами. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел. 2. Составить описание класса для работы с цепными списками строк (строки произвольной длины) с операциями включения в список, удаления из списка элемента с заданным значением данного, удаления всего списка или конца списка, начиная с заданного элемента. 3. Составить описание класса для объектов-векторов, задаваемых координатами концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами. 4. Составить описание класса прямоугольников со сторонами, параллельными осям координат. Предусмотреть возможность перемещения прямоугольников на плоскости, изменения размеров, построения наименьшего прямоугольника, содержащего два заданных прямоугольника, и прямоугольника, являющегося общей частью (пересечением) двух прямоугольников. 5. Составить описание класса для определения одномерных массивов целых чисел (n-мерных векторов). Предусмотреть возможность обращения к отдельному элементу массива с контролем выхода за пределы индексов, выполнения операций поэлементного сложения и вычитания массивов с одинаковыми границами индексов, умножения и деления всех элементов массива на скаляр, печати (вывода на экран) элементов массива по индексам и всего массива.	50	
Иная контактная работа:	0	

Раздел 5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Изучение дисциплины рекомендуется начать с ознакомления с рабочей программой, ее структурой и содержанием разделов. Учебный материал структурирован, изучение дисциплины осуществляется в тематической последовательности. **Занятия лекционного типа** дают систематизированные знания по дисциплине, концентрируют внимание на наиболее сложных и важных вопросах. Во время лекционных занятий рекомендуется вести конспектирование учебного материала; обращать внимание на формулировки и категории, раскрывающие суть проблемы, явления или процесса; зафиксировать выводы и практические рекомендации. Подготовка к **занятиям семинарского типа** включает ознакомление с вопросами по темам **лабораторного** занятия; работу с конспектом лекций, выполнение домашнего задания, работу с учебной и учебно-методической литературой, научными изданиями и электронными образовательными ресурсами, рекомендованными рабочей программой дисциплины.

Содержание **самостоятельной работы** определяется рабочей программой дисциплины, оценочными и методическими материалами, заданиями и указаниями преподавателя. Самостоятельная работа может осуществляться в аудиторной и внеаудиторной формах. Эффективным средством осуществления самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к образовательной программе, рабочей программе дисциплины, к электронным библиотечным системам, профессиональным базам данных и информационным справочным системам.

Изучение дисциплины включает выполнение **лабораторных работ**.

Раздел 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Учебно-методическое обеспечение

№№ п/п	Список используемой литературы	Количество экземпляров печатных изданий, имеющих в библиотеке, или электронный адрес издания (ресурса) в сети Интернет
УЧЕБНЫЕ, УЧЕБНО-МЕТОДИЧЕСКИЕ И НАУЧНЫЕ ИЗДАНИЯ		
1.	Хахаев, И. А. Практикум по алгоритмизации и программированию на Python [Электронный ресурс] / Хахаев И. А. 2-е изд. Москва: ИНТУИТ, 2016. - 178 с.	https://e.lanbook.com/book/100377
2.	Щерба, А. В. Программирование на Python. Первые шаги [Электронный ресурс] / Щерба А. В. Москва: Лаборатория знаний, 2022. - 250 с. ISBN 978-5-93208-578-3.	https://e.lanbook.com/book/221678
3.	Федоров, Дмитрий Юрьевич. Программирование на языке высокого уровня Python [Текст : Электронный ресурс] : учебное пособие для вузов / Д. Ю. Федоров. 5-е изд. Москва: Юрайт, 2023. - 210 с ISBN 978-5-534-14638-7.	https://urait.ru/book/programirovanie-na-yazyke-vysokogo-urovnya-python-532868
4.	Sweigart, A. Разработка компьютерных игр на языке Python [Электронный ресурс] / Sweigart A. 2-е изд. Москва: ИНТУИТ, 2016. - 504 с.	https://e.lanbook.com/book/100454

6.2. Материально-техническая база и программное обеспечение

№№ п/п	Аудитории для проведения учебных занятий, самостоятельной работы и проведения государственной итоговой аттестации	Перечень основного оборудования	Программное обеспечение
1.	314а (III)	ПК 1 - ICL RAY P222.1, клавиат.,мышь.,монитор NEC 23" LCD EX 231WP-BK (1), ПК 2 - ICL RAY P222.2, клавиат.,мышь.,монитор NEC 23" LCD EX 231WP-BK (15), Проектор мультимедийный Hitachi CP-RX93 (1), Шкаф IBM Netbay	Microsoft Windows Enterprise, Справочная правовая система "Консультант Плюс", Microsoft Office Standard, Агент Dr.Web, Комплект ГАРАНТ-Мастер, Microsoft

		25U (1), Экран настенный рулонный 200x200 см (1), Комплект учебной мебели (1)	Access, Microsoft Visio Professional, Microsoft Project Professional, Microsoft Visual Studio Enterprise, Комплект ПО для решения основных пользовательских задач
--	--	---	---

Раздел 7. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ/ ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Критерии оценивания индикаторов достижения компетенций направлены на:

- усвоение теоретического материала (объем знаний, глубина усвоения), предусмотренного рабочей программой;
- умение излагать материал (четкость, грамотность изложения материала, точность и полнота воспроизведения учебного материала);
- умение применять теоретические знания при решении практических заданий.

Шкала оценивания представлена ниже.

Уровень сформированности элементов компетенции	Критерии оценивания	Шкала оценивания
Пороговый уровень	Обучающийся имеет знания основного материала, проявляет умение логично его излагать, но может допускать неточности в изложении материала, недостаточно правильные формулировки, испытывает затруднения в выполнении практических заданий.	удовлетворительно
Продвинутый уровень	Обучающийся твердо знает программный материал, излагает его грамотно и по существу, не допускает существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения	хорошо
Высокий уровень	Обучающийся глубоко и прочно усвоил программный материал, грамотно и логически стройно его излагает, дает исчерпывающие ответы на поставленные вопросы. В ответе тесно увязывается теория с практикой, при этом обучающийся не затрудняется с ответом при видоизменении задания, свободно справляется с задачами, вопросами и другими видами применения знаний, показывает знакомство с монографической литературой, периодическими изданиями, правильно обосновывает принятые решения, свободно владеет разносторонними навыками, приемами выполнения практических работ	отлично

7.1. Текущий контроль успеваемости

Текущий контроль успеваемости обеспечивает оценивание хода освоения дисциплины (модуля) и производится с применением технологии рейтингового контроля в соответствии с технологической картой дисциплины. Порядок составления технологической карты и алгоритм проведения процедуры оценивания видов деятельности обучающихся, направленных на освоение знаний, умений, навыков и/или опыта деятельности, по

накопительной системе в баллах устанавливается положением о системе РИТМ в ФГБОУ ВО «ПГТУ»

7.2. Промежуточная аттестация обучающихся

Промежуточная аттестация обучающихся направлена на оценивание результатов обучения по дисциплине (модулю) и проводится с использованием фондов оценочных средств.

Примеры типовых контрольных заданий из базы фонда оценочных средств по образовательной программе.

Примеры типовых контрольных заданий из базы фонда оценочных средств по образовательной программе

1. Написать программу нахождения суммы большего и меньшего из трех чисел.
2. Найдите ЧЕТВЕРКИ простых чисел, лежащих внутри одного десятка. Например, 11,13,17,19..
3. Создать программу, в которой задается список целых чисел и печатаются из списка четные числа, в том же порядке, в каком они встречаются в списке. Если при обходе списка встречается число 248, выполнение программы прекращается

ТЕСТ (задачи) для текущего контроля № 1 (нулевой вариант)

1. Создать программу, в которой задается кортеж целых чисел и которая вычисляет число четных и нечетных чисел в этом кортеже.
2. Создать программу, в которой вводятся строки символов, пока не будет введена пустая строка. Затем программа их печатает.
3. Создать программу, в которой вводится строка символов, и считается число букв и число цифр в строке.

ТЕСТ (задачи) для текущего контроля № 2 (нулевой вариант)

1. Создать функцию, которая получает через параметр список элементов и соединяет их в строку.
2. Создать функцию, которая получает два целых числа x и y и возвращает их общее наименьшее делимое
3. Создать функцию, которая получает два целых числа x и y ($x > y$) и возвращает их наибольший общий делитель.

Перечень вопросов для проведения промежуточной аттестации

Вопросы к зачету по дисциплине «Объектно-ориентированное программирование»

4. Что такое неявное объединение строк и символ \
5. Для чего в Питоне используются отступы

6. Как работает метод `format` в строке:
`print('улица {0}, дом {1}, кв. {3}'.format(street, house, apartments))`
7. Для чего используются тройные кавычки?
8. Что нужно использовать, чтобы напечатать строку: *Лично я люблю шоколад «Балет»*
9. Как в Питоне закомментировать блок подряд идущих строк?
10. Как вывести оператором `print()` значения нескольких переменных?
11. Как добиться, чтобы следующий оператор `print()` выводил информацию в ту же строку?
12. Знак операции целочисленного деления в Питоне.
13. Знак операции вычисления остатка от деления в Питоне. Как вычисляется остаток от деления дробных чисел?
14. Что нужно сделать, чтобы в выражениях можно было использовать математические функции и число `pi` (т.е. чтобы интерпретатор «увидел» их)?
15. Нужно ли в Питоне определять тип простых переменных до их первого использования (как в Java)?
16. Какая функция генерирует целые случайные числа в диапазоне `[n, m)`?
17. Что можно сравнивать? Приведите пример ошибочного сравнения объектов.
18. Как в Питоне выделяется блок кода, выполняемый при **истинном** значении условия `if`?
19. Что такое `elif`?
20. Для чего используются отрицательные индексы строки?
21. Как работает цикл `for i in stroka:` где `stroka` – строка символов.
22. Что будет в переменной `st = "dfg123hjk"[4:6]` и `st = "dfg123hjk"[:6]`?
23. Что возвращает срез `[:]` и срез `::3` ?
24. Как использовать функцию `find` модуля `String`?
25. Что называют **параметрами** функции и что называют **аргументами** функции?
26. Что называется областью видимости переменных, объявленных в теле функции?
27. Как работают зарезервированные слова **global** и **nonlocal**?
28. Что такое «значения аргументов по умолчанию»?
29. Что такое «ключевые аргументы»?
30. Как работает зарезервированное слово **return**?
31. Что такое класс и объект класса?
32. Как создать класс и затем создать объект класса?
33. Что такое состояние объекта, в чем оно выражается?

34. Что такое атрибуты класса?
35. В чем разница между методом-модификатором (mutator method) и методом доступа (accessor method)?
36. Что такое конструктор класса?
37. Что такое пространство имен классов и объектов? (Укус Питона, стр104)
38. Чем отличается поле (переменная) класса от поля (переменной) объекта?
39. Почему нельзя дать одно имя полю класса и полю объекта?
40. Что называется доступом к атрибутам?
41. Что такое локальные и глобальные переменные объекта(класса)?
42. Какие методы называются статическими?
43. Что такое инкапсуляция?
44. Для чего может использоваться пустой класс?
45. Как просмотреть или изменить словари класса?
46. Как в Python открыть файл для ввода/вывода? (!!minecraftbook...p.200)
47. Как в Python прочесть строку из файла? (!!minecraftbook...p.203)
48. Как используются резервные имена withи closed? (!!minecraftbook...p.205)
49. Как записать строку символов в файл?
50. Что возвращает метод f.read(), где f – текстовый файл?
51. Как используются функции seek()и tell()?
52. Что будет, если ошибка или исключение не обработаны?
53. Что называют базовым классом (надклассом) и производным классом (подклассом)?
54. Как вызывается в производном классе конструктор базового класса?
55. Как вызвать метод базового класса в производном классе?
56. Что такое абстрактный базовый класс и метакласс?
57. Как работает разрешение (поиск) имен атрибутов?
58. Что такое множественное наследование?
59. Как создается модель приложения Django?
60. Какие типы данных используются в полях моделей-классов Django?
61. Для чего используются параметры полей?
62. Как организуются связи между моделями Django?
63. Для чего используются методы модели Django? Привести пример метода.

Задачи (для продвинутого и высокого уровня)

64. Составить программу, позволяющую компьютеру и человеку играть в слова. Предварительно программа объясняет правила игры и позволяет уточнить их в любой момент.

Тематикой игры могут быть по выбору города, животные, растения и т.д. Тематику из предложенных компьютером (не менее 5) выбирает человек. Для игры компьютер использует собственную базу данных (для каждой тематики свою), хранящуюся в виде текстового файла. Если названное человеком слово отсутствует в базе, уточняется, правильно ли оно названо, и в случае правильности заносится в базу. Правила игры: первый игрок называет слово, а второй должен предложить другое, начинающееся с той буквы, на которую оканчивается названное.

65. Определить класс ShortLine с полями lenthLine (длина отрезка), angleLine(угол наклона к оси x), beginLine(x,y) и endLine(x,y). Определить методы getEnd(beginLine, lenthLine, angleLine) вычисление координат конца по заданным координатам начала, длине отрезка и углу наклона.

С помощью класса по заданному списку $s = [[d1, \alpha_1], [d2, \alpha_2], \dots, [dk, \alpha_k]]$ где d_i – длина i -го отрезка, α_i – угол наклона, вычислить элементы списка $[[x1, y1], [x2, y2], \dots, [xk, yk]]$ – координат угловых точек ломанной, выходящей из начала координат и имеющей длины и углы наклона сегментов, которые заданы списком s .

66. Построить систему классов для описания плоских геометрических фигур: круга, квадрата, прямоугольника. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и поворота на заданный угол.
67. Составить программу, которая раздает игральные карты заданному количеству игроков (одним из игроков является человек, за остальных играет компьютер) и моделирует игру в «дурака». Компьютерная программа играет случайным образом, без анализа уже вышедших карт. Количество игроков не превышает шести.
- 68.