

Б1.В.ДВ.03.01

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА)

Учебная дисциплина	<i>Технология разработки объектно-ориентированных приложений на Python</i>
По направлению подготовки	<i>09.03.03</i>
	<i>«Прикладная информатика»</i>
Профиль (программа бакалавриата)	<i>«Прикладная информатика в цифровой экономике»</i>
Форма обучения	<i>Очная</i>

Оценочные материалы (средства) дисциплины рассмотрены (актуализированы) и утверждены на заседании кафедры прикладной информатики и высшей математики

Протокол заседания № 10 от «25» мая 2026 г.

Заведующий кафедрой Стрекалова Наталья Борисовна

Оглавление

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ И ЭТАПЫ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ	3
2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ОБЕСПЕЧИВАЮЩИХ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ	6
3. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ	26
4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНОВАНИЯ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ	29
5. ДИАГНОСТИЧЕСКИЕ ЗАДАНИЯ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ УРОВЕНЬ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ	36
6. ЛИСТ СОГЛАСОВАНИЯ	50

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ И ЭТАПЫ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Оценочные материалы (средства) сформированы в соответствии с ФГОС ВО - бакалавриат по направлению подготовки 09.03.03 «Прикладная информатика», утвержденный приказом Министерства образования и науки Российской Федерации от 19.09.2017 № 922 (с изменениями и дополнениями от 26.11.2020, 08.02.2021). В соответствии с матрицей компетенций основной профессиональной образовательной программы «Прикладная информатика в цифровой экономике» (уровень бакалавриата) в процессе обучения по дисциплине «Технология разработки объектно-ориентированных приложений на Python» происходит формирование закрепленных за дисциплиной компетенций обучающихся. Оценка сформированности компетенций на каждом этапе обучения происходит через оценку планируемых результатов обучения по дисциплине (знаний, умений, навыков).

Результаты освоения образовательной программы (компетенции)	Индикаторы достижения компетенций	Планируемые результаты обучения по дисциплине	Этапы формирования компетенции (семестры и темы)	Оценочное средство
ПК-2 Способен проектировать информационные системы	ПК-2.1. Моделирует прикладные процессы предметной области с учетом внедрения сквозных цифровых технологий и передовых ИТ-решений прикладной сферы	знать: - особенности разработки объектно-ориентированных программ и приемы объектно-ориентированного решения задач на языке Python; уметь: - проектировать семейства классов в заданной прикладной области и применять их для решения профессиональных задач на языке Python; владеть: навыками моделирования и описания моделей семейств классов;	4 семестр Тема 3. Разработка классов в языке Python Тема 4. Наследование и интерфейсы в языке Python. Документирование кода. Создание и подключение модулей Тема 5. Обобщённые типы и коллекции значений в языке Python	4 семестр - практические работы по темам 3, 4, 5 - ответ на теоретический вопрос экзамена - решение практической задачи на экзамене

	ПК-2.2. Разрабатывает модель информационной системы, в том числе с опорой на современные облачные решения	Знать: - возможности и преимущества языка Python для проектирования информационных систем разного назначения; - основные приемы создания компонентов информационных систем для обеспечения и анализа прикладных процессов Уметь: - проектировать программное обеспечение разного назначения (вычислительные системы, программы визуализации данных и т.п.) с применением библиотек языка Python (NumPy, Tkinter);. Владеть: - навыками анализа и выбора эффективных способов и средств разработки программных модулей информационных систем на языке Python для решения аналитических задач.	4 семестр Тема 3. Разработка классов в языке Python Тема 4. Наследование и интерфейсы в языке Python. Документирование кода. Создание и подключение модулей Тема 5. Обобщённые типы и коллекции значений в языке Python	4 семестр - практические работы по темам 3, 4, 5 - ответ на теоретический вопрос экзамена - решение практической задачи на экзамене
ПК-3 Способен разрабатывать информационные системы	ПК-3.1. Осуществляет кодирование на современных языках программирования, в том числе используя возможности специализированных цифровых платформ для индивидуальной и	знать: - синтаксис, типы данных и операторы языка Python, парадигму программ и подпрограмм; уметь: - разрабатывать объектно-ориентированные программы, используя классы и модули из библиотек языка Python для решения аналитических задач. владеть: - навыками применения объектно-	4 семестр Тема 1. Возможности языка Python. Первая программа. Знакомство со средой разработки IDLE. Тема 2. Основы Python Тема 4. Наследование и интерфейсы в	4 семестр - устный опрос по темам 1, 2 - практические работы по темам 4, 5, 6 - ответ на теоретический вопрос экзамена - решение практическ

	совместной разработки информационных систем	ориентированной технологии и стандартных библиотек языка Python для решения задач вычислительного и аналитического характера.	языке Python. Документирование кода. Создание и подключение модулей Тема 5. Обобщённые типы и коллекции значений в языке Python Тема 6. Работа со строками в языке Python	ой задачи на экзамене
	ПК-3.2. Проводит тестирование компонентов информационных систем, в том числе используя возможности специализированных цифровых платформ	Знать: - основные приемы тестирование компонентов информационных систем на языке Python; Уметь: - составлять блоки тестов компонентов информационных систем на языке Python Владеть: - навыками применения тестов для информационных систем на языке Python	4 семестр Тема 1. Возможности языка Python. Первая программа. Знакомство со средой разработки IDLE. Тема 2. Основы Python Тема 4. Наследование и интерфейсы в языке Python. Документирование кода. Создание и подключение модулей Тема 5. Обобщённые типы и коллекции значений в языке Python Тема 6. Работа со строками в языке Python	4 семестр - устный опрос по темам 1, 2 - практические работы по темам 4, 5, 6 - ответ на теоретический вопрос экзамена - решение практической задачи на экзамене

2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ОБЕСПЕЧИВАЮЩИХ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Устные опросы

*Вопросы для проведения устного опроса
по теме «Синтаксис языка Python, классы в языке Python.»*

1. Опишите правила построения имён переменных в языке Python.
2. Как записываются комментарии на Python? Подумайте, как комментирование можно использовать при поиске ошибок в алгоритме?
3. Расскажите о работе оператора вывода Python.
4. Что такое переменная? Как транслятор определяет тип переменной?
5. Зачем нужен тип переменной?
6. Как изменить значение переменной?
7. Что такое оператор присваивания?
8. Почему желательно выводить на экран подсказку перед вводом данных?
9. Подумайте, когда можно вычислять результат прямо в операторе вывода, а когда нужно заводить отдельную переменную.
10. Что такое форматный вывод? Как вы думаете, где он может быть полезен?
11. Какие типы данных вы знаете?
12. Какие данные записываются в логические переменные?
13. Расскажите об особенностях переменных в языке Python. Почему может получиться, что изменение одной переменной автоматически приводит к изменению другой?
14. Что такое приоритет операций? Зачем он нужен?
15. В каком порядке выполняются операции, если они имеют одинаковый приоритет?
16. Зачем используются скобки?
17. Что происходит, если в выражения входят переменные разных числовых типов? Какого типа будет результат?
18. Опишите операции // и %.
19. Расскажите о проблеме вычисления остатка от деления в различных языках программирования.
20. Какие стандартные математические функции вы знаете? В каких единицах задается аргумент тригонометрических функций?
21. Как выполнить округление вещественного числа к ближайшему целому?
22. Какие числа называют случайными? Зачем они нужны?
23. Как получить «естественное» случайное число? Почему такие числа почти не используются в цифровой технике?
24. Чем отличаются псевдослучайные числа от случайных?
25. Какие функции для получения псевдослучайных чисел вы знаете?

*Вопросы для проведения устного опроса
по теме «Наследование и инкапсуляция в языке Python.»*

1. Объявление класса
2. Управление атрибутами и методами класса
3. Объявление объектов
4. Множественное наследование
5. Заимствование свойств и методов у родительского класса

6. Через какие средства в Python выражен полиморфизм
7. Через какие средства в Python выражено наследование
8. Через какие средства в Python выражена инкапсуляция
9. Какие роли выполняет наследование?
10. Перечислите виды реализации инкапсуляции
11. Что такое магические методы, для чего нужны?
12. Как в классе сослаться на родительский класс?
13. Возможно ли множественное наследование?

Критерии оценивая устного опроса:

- оценка «отлично» выставляется студенту, если он ответил развернуто на один из заданных вопросов, активно дополнял ответы других студентов или задавал им дополнительные вопросы;
- оценка «хорошо» выставляется студенту, если он ответил развернуто на один из заданных вопросов или ответил кратко на ряд вопросов;
- оценка «удовлетворительно» выставляется студенту, если он кратко ответил на один из задаваемых вопросов или просто дополнял ответы других студентов, задавал им дополнительные вопросы;
- оценка «неудовлетворительно» выставляется студенту, если он не смог ответить правильно на заданный вопрос, либо не отвечал на вопросы и не участвовал в их обсуждении.

Практические работы

Практическое задание по теме «Разработка классов в языке Python»

Практическая работа №1.

1. Напишите программу для решения примера (по вариантам). Предусмотрите проверку деления на ноль. Все необходимые переменные пользователь вводит через консоль. Запись |пример| означает «взять по модулю», т.е. если значение получится отрицательным, необходимо сменить знак с минуса на плюс.
 Вариант 1. $|((a^2/b^2 + c^2*a^2)/(a+b+c*(k-a/b^3)) + c + (k/b - k/a)*c|$ Вариант 2. $|((a^2 - b^3 - c^3*a^2)*(b-c+c*(k-d/b^3)) - (k/b - k/a)*c)^2 - 20000|$ Вариант 3. $|1 - a*b^c - a*(b^2 - c^2) + (b-c+a)*(12+b)/(c-a)|$ Вариант 4. $|a - b*c*d^3 + (c^5 - a^2)/a + f^3*(a-213)|$
2. Дан произвольный список, содержащий и строки и числа. Вариант 1. Выведите все четные элементы построчно. Вариант 2. Выведите все нечетные элементы построчно. Вариант 3. Выведите все четные элементы в одной строке. Вариант 4. Выведите все нечетные элементы в одной строке.
3. Дан произвольный список, содержащий только числа.
 Вариант 1. Выведите результат сложения всех чисел больше 10.
 Вариант 2. Выведите результат сложения всех чисел от 1 до 10.
 Вариант 3. Выведите результат умножения всех чисел меньше 10.
 Вариант 4. Выведите результат умножения всех чисел меньше 10.
4. Дан произвольный список, содержащий только числа. Вариант 1. Выведите максимальное число. Вариант 2. Выведите минимальное число.
 Вариант 3. Выведите среднее арифметическое (сумма всех чисел, деленная на количество элементов).

Вариант 4. Выведите число, находящееся посередине массива.

Методические рекомендации к выполнению работы

Перед выполнением заданий, ознакомьтесь с теоретическими материалами по дисциплине. В заданиях, предусмотренных по вариантам,

Критерии оценивая:

- оценка «*отлично*» выставляется студенту, если задание выполнено в полном объеме, студент хорошо ориентируется в технологии разработки функций пользователя, программа логично и корректно демонстрирует работу всех функций;
- оценка «*хорошо*» выставляется студенту, если он выполнил задание почти в полном объеме (может отсутствовать пункт), допускает небольшие «проектно-технологические» ошибки в разработке функций;
- оценка «*удовлетворительно*» выставляется студенту, если он выполнил первые 3 пункта задания;
- оценка «*неудовлетворительно*» выставляется студенту, если он выполнил менее 3 пунктов задания.

Практическая работа №2.

1. Пусть задано некоторое число `my_number`. Пользователь вводит с клавиатуры свое число `user_number`.
 - Вариант 1. Запрашивайте у пользователя вводить число `user_number` до тех пор, пока оно не будет меньше `my_number`.
 - Вариант 2. Запрашивайте у пользователя вводить число `user_number` до тех пор, пока оно не будет равно `my_number`.
 - Вариант 3. Запрашивайте у пользователя вводить число `user_number` если оно равно `my_number`.
 - Вариант 4. Запрашивайте у пользователя вводить число `user_number` до тех пор, пока оно не будет больше `my_number`.
2. Пусть задан список, содержащий строки.
 - Вариант 1. Выведите построчно все строки размером от 5 до 10 символов.
 - Вариант 2. Выведите построчно все строки размером менее 10 символов.
 - Вариант 3. Выведите все строки, заканчивающиеся буквой г.
 - Вариант 4. Выведите все строки, начинающиеся с буквы г.
3. Сгенерируйте и выведите:
 - Вариант 1. Случайную строку, состоящую из 5 символов, содержащую только заглавные буквы русского алфавита.
 - Вариант 2. Строку размером N символов (N вводится с клавиатуры) и состоящую из букв R.
 - Вариант 3. Случайную строку размером 6 символов, содержащую только цифры. Строка должна содержать хотя бы одну цифру 3.
 - Вариант 4. Случайную строку, состоящую из 8 символов и содержащую цифры и буквы. Строка должна содержать хотя бы одну цифру.
4. Пусть дана строка:
 - Вариант 1. На основе данной строки сформируйте новую, содержащую только цифры. Выведите новую строку.

- Вариант 2. На основе данной строки сформируйте новую, содержащую только буквы. Выведите новую строку.
- Вариант 3. На основе данной строки сформируйте новую, содержащую только буквы Л. Выведите новую строку.
- Вариант 4. На основе данной строки сформируйте две новые. Первая строка содержит только цифры, вторая — только буквы. Выведите новые строки построчно.

Методические рекомендации к выполнению работы

Перед выполнением заданий, ознакомьтесь с теоретическими материалами по дисциплине.

Краткая справка:

1. Для генерации случайного числа нужно импортировать специальную библиотеку `random` (она входит в стандартный пакет языка Python). И затем использовать одну из функций этой библиотеки. Например:

```
import random  
printrandom.randint(1,10) #число в диапазоне от 1 до 10
```
2. Для просмотра информации о модуле, библиотеке или классе, используйте специальную функцию `help(имя_модуля)`. Например: `help(random)`
3. Функция вычисления длины списка или строки (сколько элементов в списке или сколько символов в строке): `len(строка)`.

В заданиях, предусмотренных по вариантам.

Критерии оценивая:

- оценка «*отлично*» выставляется студенту, если задание выполнено в полном объеме, студент хорошо ориентируется в технологии разработки функций пользователя, программа логично и корректно демонстрирует работу всех функций;
- оценка «*хорошо*» выставляется студенту, если он выполнил задание почти в полном объеме (может отсутствовать пункт), допускает небольшие «проектно-технологические» ошибки в разработке функций;
- оценка «*удовлетворительно*» выставляется студенту, если он выполнил первые 3 пункта задания;
- оценка «*неудовлетворительно*» выставляется студенту, если он выполнил менее 3 пунктов задания.

Практическое задание по теме «Наследование и интерфейсы в языке Python.»

Практическая работа №3.

1. Пусть дана строка, состоящая из слов, пробелов и знаков препинания. На основании этой строки создайте новую (и выведите ее на консоль):
 - Вариант 1. Содержащую только слова больше 5 символов. Разделитель слов в строке — пробел.
 - Вариант 2. Содержащую только слова, в которых первые две буквы — «Ли».
 - Вариант 3. Содержащую только слова размером от 5 до 10 символов.
 - Вариант 4. Содержащую только слова, в которых две последние буквы — «ов».
2. Пусть дана строковая переменная, содержащая информацию о студентах:

my_string = «Ф;И;О;Возраст;Категория;_Иванов;Иван;Иванович;23 года;Студент 3 курса;_Петров;Семен;Игоревич;22 года;Студент 2 курса».

Вариант 1. Выведите информацию в виде:

ФИО	Категория	Возраст
Иванов Иван Иванович	Студент 3 курса	23 года
Петров Семен Игоревич	Студент 2 курса	22 года

Вариант 2. Выведите информацию в виде:

ФИО	Возраст	Категория
Иванов Иван Иванович	23 года	Студент 3 курса
Петров Семен Игоревич	22 года	Студент 2 курса

Вариант 3. Выведите информацию в виде:

Ф	И	О	О студенте
Иванов	Иван	Иванович	Студент 3 курса, 23 года
Петров	Семен	Игоревич	Студент 2 курса, 22 года

Вариант 4. Выведите информацию в виде:

ФИО	О студенте
-----	------------

Иванов Иван Иванович Студент 3 курса, 23 года

Петров Семен Игоревич Студент 2 курса, 22 года

3. Пусть дана строковая переменная, содержащая информацию о студентах вида:
my_string = «ФИО;Возраст;Категория;_Иванов Иван Иванович;23 года;Студент 3 курса;_Петров Семен Игоревич;22 года;Студент 2 курса;_Иванов Семен Игоревич;22 года;Студент 2 курса;_Акибов Ярослав Наумович;23 года;Студент 3 курса;_Борков Станислав Максимович;21 год;Студент 1 курса;_Петров Семен Семенович;21 год;Студент 1 курса;_Романов Станислав Андреевич;23 года;Студент 3 курса;_Петров Всеволод Борисович;21 год;Студент 2 курса».

- Вариант 1. Выведите построчно информацию о студентах, чья фамилия — «Петров».
- Вариант 2. Выведите построчно информацию о студентах, чей возраст — «21 год».
- Вариант 3. Выведите построчно информацию о студентах, чей возраст больше «21 года».
- Вариант 4. Выведите построчно информацию о студентах, чьи фамилии начинаются на букву «А» или «Б».

4. Пусть дана строка произвольной длины. Выведите информацию о том, сколько в ней символов и сколько слов.

Методические рекомендации к выполнению работы

Перед выполнением заданий, ознакомьтесь с теоретическими материалами по дисциплине.

В заданиях, предусмотренных по вариантам.

Критерии оценивая:

- оценка «*отлично*» выставляется студенту, если задание выполнено в полном объеме, студент хорошо ориентируется в технологии разработки функций пользователя, программа логично и корректно демонстрирует работу всех функций;
- оценка «*хорошо*» выставляется студенту, если он выполнил задание почти в полном объеме (может отсутствовать пункт), допускает небольшие «проектно-технологические» ошибки в разработке функций;
- оценка «*удовлетворительно*» выставляется студенту, если он выполнил первые 3 пункта задания;
- оценка «*неудовлетворительно*» выставляется студенту, если он выполнил менее 3 пунктов задания.

Практическая работа №4.

1. Пусть дана матрица чисел размером $N \times N$. Представьте данную матрицу в виде списка. Выведите результат сложения всех элементов матрицы.
2. Пусть дан список из 10 элементов.
 - Вариант 1. Удалите первые 2 элемента и добавьте 2 новых. Выведите список на экран.
 - Вариант 2. Удалите все четные элементы и добавьте 2 новых. Выведите список на экран.

Вариант 3. Удалите элементы с 4 по 8 и добавьте 2 новых. Выведите список на экран.

Вариант 4. Добавьте 5 новых элементов и оставьте все нечетные элементы. Выведите список на экран.

3. Пусть журнал по предмету «Информационные технологии» представлен в виде списка: `my_len = [['БО-331101', 'Акулова Алена', 'Бабушкина Ксения', .....], ['БОВ-421102', .....], ['БО-331103', .....]]`.

Вариант 1. Выведите список студентов конкретной группы построчно в виде:

<Название группы>

<ФИО>

<ФИО>

Вариант 2. Выведите список студентов конкретной группы в одной строке в виде: <Название группы>: <ФИО>, <ФИО>

Вариант 3. Выведите списки всех групп построчно в виде:

<Название группы>

<ФИО>

<ФИО>

Вариант 4. Выведите списки студентов, название группы которых начинается на «БО», в виде: <Название группы>: <ФИО>,

<ФИО>

4. Пусть журнал по предмету «Информационные технологии» представлен в виде списка: `my_len = [['БО-331101', 'Акулова Алена', 'Бабушкина Ксения', .....], ['БОВ-421102', .....], ['БО-331103', .....]]`.

Вариант 1. Выведите всех студентов (и их группы), если фамилия студента начинается на букву А.

Вариант 2. Выведите всех студентов (и их группы), чья фамилия меньше 7 букв.

Вариант 3. Выведите всех студентов (и их группы), чья фамилия начинается на букву «П», а имя на букву «А».

Вариант 4. Выведите всех студентов (и их группы), чей порядковый номер в журнале – четное число.

Методические рекомендации к выполнению работы

Перед выполнением заданий, ознакомьтесь с теоретическими материалами по дисциплине. В заданиях, предусмотренных по вариантам.

Критерии оценивая:

- оценка «отлично» выставляется студенту, если задание выполнено в полном объеме, студент хорошо ориентируется в технологии разработки функций пользователя, программа логично и корректно демонстрирует работу всех функций;
- оценка «хорошо» выставляется студенту, если он выполнил задание почти в полном объеме (может отсутствовать пункт), допускает небольшие «проектно-технологические» ошибки в разработке функций;
- оценка «удовлетворительно» выставляется студенту, если он выполнил первые 3 пункта задания;
- оценка «неудовлетворительно» выставляется студенту, если он выполнил менее 3 пунктов задания.

Практическая работа №5.

1. Пусть дана некоторая директория (папка). Посчитайте количество файлов в данной директории (папке) и выведите на экран.
2. Пусть дан файл students.csv, в котором содержится информация о студентах в виде:
№;ФИО;Возраст;Группа
1;Иванов Иван Иванович;23;БО-111111 2;Сидоров
Семен Семенович;23;БО-111111 3;Яшков Илья
Петрович;24;БО-222222
...
Считайте информацию из файла в структуру: [[№, ФИО, Возраст, Группа],[№, ФИО, Возраст, Группа],[№, ФИО, Возраст, Группа]] (список списков).
Вариант 1. Выведите информацию о студентах, отсортировав их по фамилии.
Вариант 2. Выведите информацию о студентах, отсортировав их по возрасту.
Вариант 3. Выведите информацию о студентах, отсортировав их по номеру группы.
Вариант 4. Выведите информацию о студентах, в возрасте старше 22 лет.
3. Добавьте к задаче №2 пользовательский интерфейс:
Вариант 1. По увеличению возраста всех студентов на 1.
Вариант 2. По уменьшению возраста всех студентов на 1.
Вариант 3. По увеличению возраста студентов в заданной пользователем группе на 1.
Вариант 4. По уменьшению возраста студентов в заданной пользователем группе на 1.
4. Добавьте к пользовательскому интерфейсу из задачи №3 возможность сохранения новых данных
обратно в файл. Выведите информацию о студентах, отсортировав их по фамилии.

Методически е рекомендации к выполнению работы

Перед выполнением заданий, ознакомьтесь с теоретическими материалами по дисциплине. В заданиях, предусмотренных по вариантам

Критерии оценивая:

- оценка «отлично» выставляется студенту, если задание выполнено в полном объеме, студент хорошо ориентируется в технологии разработки функций пользователя, программа логично и корректно демонстрирует работу всех функций;
- оценка «хорошо» выставляется студенту, если он выполнил задание почти в полном объеме (может отсутствовать пункт), допускает небольшие «проектно-технологические» ошибки в разработке функций;
- оценка «удовлетворительно» выставляется студенту, если он выполнил первые 3 пункта задания;
- оценка «неудовлетворительно» выставляется студенту, если он выполнил менее 3 пунктов задания.

Практическая работа №6.

1. Реализуйте задания предыдущих лабораторных работ (выполненные согласно вашему варианту) в виде пользовательских функций.
2. Реализуйте единое пользовательское меню выбора соответствующих функций из задания №1 в виде:
0 – Выход из программы
1 – Название функции №1.

2 – Название функции №2.

3 – ...

После выполнения каждой из функций запрашивайте у пользователя «Вы хотите продолжить?» Если ответ «да» (yes, Y, 1), то снова выводите меню. Если ответ «нет» (no, N, 0), то завершите программу.

Методические рекомендации к выполнению работы

Перед выполнением заданий, ознакомьтесь с теоретическими материалами по дисциплине. В заданиях, предусмотренных по вариантам

Критерии оценивая:

- оценка «отлично» выставляется студенту, если задание выполнено в полном объеме, студент хорошо ориентируется в технологии разработки функций пользователя, программа логично и корректно демонстрирует работу всех функций;
- оценка «хорошо» выставляется студенту, если он выполнил задание почти в полном объеме (может отсутствовать пункт), допускает небольшие «проектно-технологические» ошибки в разработке функций;
- оценка «удовлетворительно» выставляется студенту, если он выполнил первые 3 пункта задания;
- оценка «неудовлетворительно» выставляется студенту, если он выполнил менее 3 пунктов задания.

Практическое задание по теме «Работа со строками в языке Python.»

Практическая работа №7.

1. Пусть дан словарь. Посчитайте и выведите сколько в словаре ключей.
2. Пусть дан файл students.csv, в котором содержится информация о студентах в виде:
№;ФИО;Возраст;Группа
1;Иванов Иван Иванович;23;БО-111111 2;Сидоров
Семен Семенович;23;БО-111111 3;Яшков Илья
Петрович;24;БО-222222
...
Считайте информацию из файла в структуру: {№: [ФИО, Возраст, Группа], №:
[...], №: [...]} (словарь, где ключи – это порядковые номера студентов).
Вариант 1. Выведите информацию о студентах, отсортировав их по фамилии.
Вариант 2. Выведите информацию о студентах, отсортировав их по возрасту.
Вариант 3. Выведите информацию о студентах, отсортировав их по номеру группы.
Вариант 4. Выведите информацию о студентах, в возрасте старше 22 лет.
3. Добавьте к задаче №2 возможность:
Вариант 1. Увеличить возраст всех студентов на 1.
Вариант 2. Уменьшить возраст всех студентов на 1.
Вариант 3. Увеличить возраст студентов в заданной пользователем группе на 1.
Вариант 4. Уменьшить возраст студентов в заданной пользователем группе на 1.

4. Добавьте к пользовательскому интерфейсу из задачи №3 возможность сохранения новых данных в файл.

Методические рекомендации к выполнению работы

Перед выполнением заданий, ознакомьтесь с теоретическими материалами по дисциплине. В заданиях, предусмотренных по вариантам.

Практическая работа №8.

1. Пусть список студентов представлен в виде структуры [[№, ФИО, Возраст, Группа],[№, ФИО, Возраст, Группа],[№, ФИО, Возраст, Группа]]. Преобразуйте список в словарь вида: {№: [ФИО, Возраст, Группа], №: [...], №: [...]}
2. Добавьте к задаче №1 для словаря возможность (без преобразования словаря обратно в список):
 1. Увеличить возраст конкретного студента на 1. Поиск по «ФИО» («ФИО» студента необходимо ввести с клавиатуры).
 2. Изменить «ФИО» студента. Поиск по «ФИО» (старое и новое «ФИО» студента необходимо ввести с клавиатуры).
 3. Увеличить возраст конкретного студента на 1. Поиск по «№» («№» студента необходимо ввести с клавиатуры).
 4. Изменить группу студента. Поиск по «ФИО» («ФИО» студента и новый номер группы необходимо ввести с клавиатуры).
 5. Удалить запись о студенте. Поиск по «№» («№» студента, которого нужно удалить из списка, задается с клавиатуры)
 6. Если возраст студента больше 22 уменьшить его на 1.
 7. Если возраст студента равен 23, удалить его из списка.
 8. У всех студентов с фамилией «Иванов» увеличить возраст на 1.
 9. У студентов с фамилией «Иванов» изменить фамилию на «Сидоров».
 10. Поменять «ФИО» и «Группа» местами.
3. Добавьте к пользовательскому интерфейсу из задачи №2 возможность вывода из словаря (без его преобразования в список):
 1. Списка студентов (а также информацию о них) группы 'БО-111111'.
 2. Списка студентов (а также информацию о них) с номерами 1-10.
 3. Списка студентов (а также информацию о них) в возрасте 22 лет.
 4. Список студентов (а также информацию о них) с фамилией 'Иванов'.
 5. Списка студентов (а также информацию о них), чьи фамилии заканчиваются на «а».
 6. Списка студентов (а также информацию о них), чей возраст – четное число.
 7. Списка студентов (а также информацию о них), если в возрасте студента встречается число 5.
 8. Списка студентов (а также информацию о них), если их номера группы длиннее 7 символов.
 9. Списка студентов (а также информацию о них), если их «№» четное число.
 10. Списка студентов (а также информацию о них), если их номер группы заканчивается на «1».

Методические рекомендации к выполнению работы

Перед выполнением заданий, ознакомьтесь с теоретическими материалами по дисциплине.

Практическая работа №9.

1. Пусть список студентов представлен в виде структуры {№: [ФИО, Возраст, Группа], №: [....], №: [....]}. Реализуйте функционал по добавлению нового студента (данные вводятся через консоль).
2. Пусть список студентов представлен в виде структуры {№: [ФИО, Возраст, Группа], №: [....], №: [....]}. Реализуйте функционал по изменению всех данных о студенте (поиск по «№»).
3. Пусть список студентов представлен в виде структуры {№: [ФИО, Возраст, Группа], №: [....], №: [....]}. Реализуйте функционал по удалению данных о студенте (поиск по «№»).
4. Пусть список студентов представлен в виде структуры {№: [ФИО, Возраст, Группа], №: [....], №: [....]}. Выведите информацию о студенте с конкретным «№» («№» задается через консоль).

Методические рекомендации к выполнению работы

Перед выполнением заданий, ознакомьтесь с теоретическими материалами по дисциплине.

Практическая работа №10.

Пусть дана матрица:

1	2	3	4	5	6	7	8
8	7	6	5	4	3	2	1
2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2
1	3	5	7	9	7	5	3
3	1	5	3	2	6	5	7
1	7	5	9	7	3	1	5
2	6	3	5	1	7	3	2

Вариант №1. Определите структуру для хранения матрицы.

1. Напишите функцию возведения всех элементов матрицы в квадрат.
2. Напишите функцию возведения всех четных элементов в квадрат.
3. Напишите функцию возведения в квадрат всех элементов меньше 5.
4. Напишите функцию возведения первых четырех строк в квадрат. Вариант

№2. Определите структуру для хранения матрицы.

1. Напишите функцию сложения по строкам.
2. Напишите функцию сложения по столбцам.
3. Напишите функцию сложения по строкам четных элементов.
4. Напишите функцию сложения по столбцам четных элементов. Вариант

№3. Определите структуру для хранения матрицы.

1. Напишите функцию возведения в квадрат всех элементов четных столбцом.
2. Напишите функцию возведения в квадрат всех элементов четных строк.
3. Напишите функцию сложения по строкам четных элементов.
4. Напишите функцию сложения по столбцам четных элементов.

Вариант №4. Определите структуру для хранения матрицы.

1. Напишите функцию умножения по строкам.
2. Напишите функцию сложения всех элементов матрицы.
3. Напишите функцию сложения всех элементов матрицы меньших 5 (в одно число) и всех элементов матрицы больше или равных 5 (в другое число). Сравните числа и выведите, какое из них больше.
4. Напишите функцию замены значений всех элементов матрицы на 0. Вариант

Вариант №5. Определите структуру для хранения матрицы.

1. Напишите функцию замены всех четных элементов матрицы на 0.

2. Пользователь вводит через консоль число. Напишите функцию, которая заменит все числа в матрице, которые меньше введенного, на введенное число.
3. Напишите функцию возведения всех чисел 5 в квадрат.
4. Напишите функцию, которая удалит 4 последних строки.

Вариант №6. Определите структуру для хранения матрицы.

1. Пусть пользователь через консоль вводит число. Напишите функцию удаления строки в матрице, чей номер равен введенному числу.
2. Пусть пользователь через консоль вводит число. Напишите функцию удаления столбца в матрице, чей номер равен введенному числу.
3. Напишите функцию очистки матрицы.
4. Напишите функцию, которая поменяет первый и последний столбцы матрицы местами.

Вариант №7. Определите структуру для хранения матрицы.

1. Напишите функцию, которая поменяет первую и последнюю строку матрицы местами.
2. Напишите функцию создания матрицы любого размера, заполненной нулями (размер задается через консоль).
3. Напишите функцию, которая посчитает, сколько раз в матрице встречается число 3.
4. Пусть пользователь может ввести число через консоль. Напишите функцию, которая посчитает, сколько раз в матрице встречается заданное пользователем число.

Вариант №8. Определите структуру для хранения матрицы.

1. Пусть пользователь через консоль вводит два числа: первое – номер строки, второе – номер столбца. Напишите функцию, которая найдет число в данной позиции.
2. Пусть пользователь через консоль вводит число: номер строки. Напишите функцию, которая выведет все элементы данной строки, возведенные в квадрат.
3. Пусть пользователь через консоль вводит число: номер столбца. Напишите функцию, которая посчитает сумму элементов из данного столбца.
4. Пусть пользователь через консоль вводит три числа: первое – номер строки, второе – номер столбца, третье – число (на которое нужно перезаписать данные в введенной позиции). Напишите функцию, которая найдет число в данной позиции (пересечение введенных строки и столбца) и заменит на введенное пользователем.

Методические рекомендации к выполнению работы

Перед выполнением заданий, ознакомьтесь с теоретическими материалами по дисциплине. В заданиях, предусмотренных по вариантам.

Практическая работа №11.

Методические рекомендации к выполнению работы

Лабораторная работа рассчитана на 4 занятия (4 недели, 1 занятие в неделю). Итоговая оценка выставляется на последнем занятии.

Перед выполнением заданий, ознакомьтесь с теоретическими материалами по дисциплине. В заданиях, предусмотренных по вариантам.

Задания

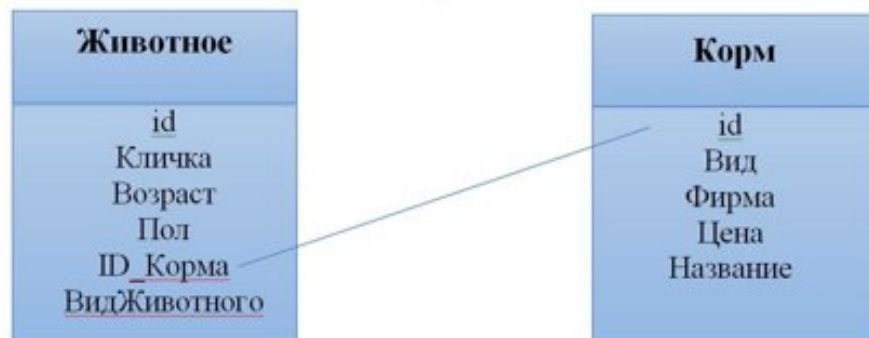
1. Представьте таблицы (согласно вашему варианту) в виде структур языка Python.
2. Реализуйте в консоли интерфейс по добавлению, удалению, изменению данных. Имейте в виду, что связанные операции (удаление, добавление, изменение) для связанных таблиц, должны изменять данных во всех связанных структурах.
3. Выведите следующую информацию в консоль построчно: Вариант 1. Для каждого преподавателя: «ФИО преподавателя»,
«название кафедры», «должность преподавателя». Вариант 2. Для каждого животного: «Кличка животного», «пол»,
«возраст», «название корма», «цена корма». Вариант 3. Для каждого ребенка: «ФИО ребенка», «ФИО отца», «ФИО матери»,
«ФИО врача», «возраст ребенка». Вариант 4. Для каждого товара: «№ товара», «название товара»,
«стоимость товара», «цвет товара». Вариант 5. Для каждого контента: «Название контента», «название меню»,
«ник автора», «аннотация». Вариант 6. Для каждого человека: «ФИО персоны», его должность, в какой фирме работает. Вариант 7. Для каждого преподавателя: «ФИО преподавателя», список дисциплин (которые он ведет). Вариант 8. Для каждой дисциплины: «Название дисциплины», в какой день недели ведется, на какой паре. Вариант 9. Для каждой дисциплины: «Название дисциплины», «количество лекций», «количество практик», «форма контроля»,
«название кафедры» (которая читает дисциплину). Вариант 10. Для каждой страны: «Название страны», список городов.
4. Посчитайте и выведите результат:
Вариант 1. Для каждой кафедры: сколько всего преподавателей.
Вариант 2. Для каждого корма: сколько животных им питается.
Вариант 3. Для каждого врача: сколько детей он лечит.
Вариант 4. Для каждого цвета: количество товаров.

Вариант 5. Для каждого пользователя: количество контента, которое он добавил.
Вариант 6. Для каждой фирмы: сколько в ней сотрудников.
Вариант 7. Для каждого преподавателя: сколько дисциплин он ведет.
Вариант 8. Для каждого дня недели: сколько всего пар предусмотрено в этот день.
Вариант 9. Для каждой кафедры: количество дисциплин, которое ведется сотрудниками данной кафедры.
Вариант 10. Для каждого города: сколько в нем улиц.
5. Реализуйте функционал по сохранению данных в файлы формата .csv и считыванию информации из файлов.

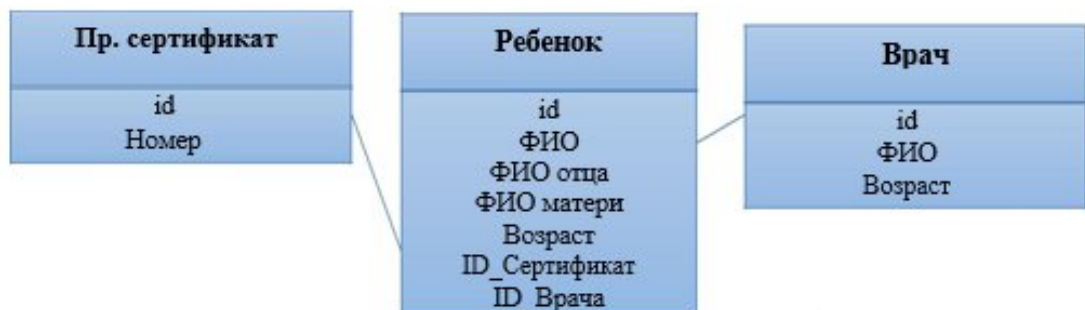
Вариант №1



Вариант №2



Вариант №3



Вариант №4



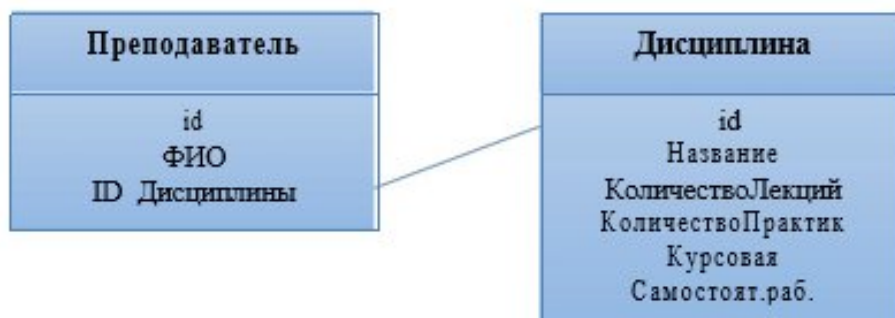
Вариант №5



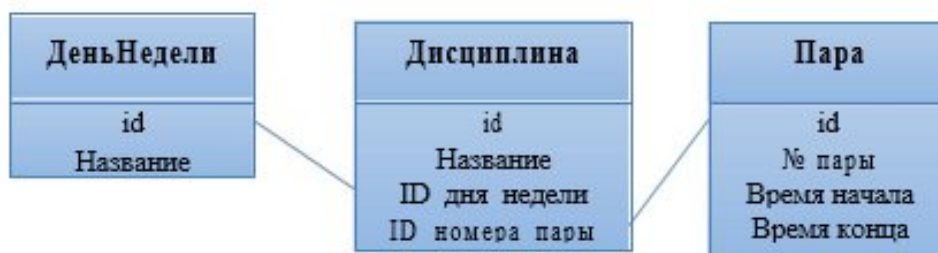
Вариант №6



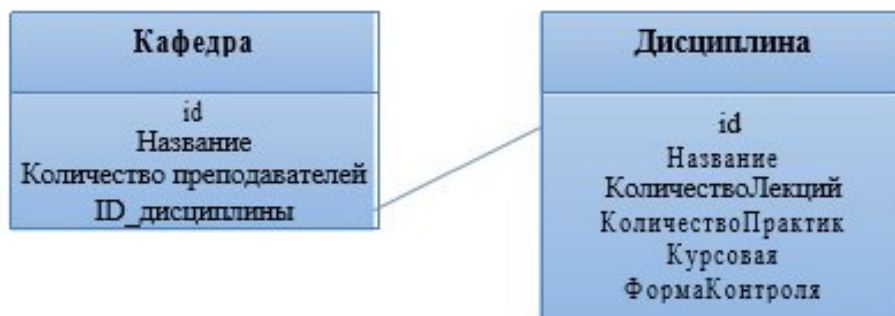
Вариант №7



Вариант №8



Вариант №9



Вариант №10



Практическая работа №12.

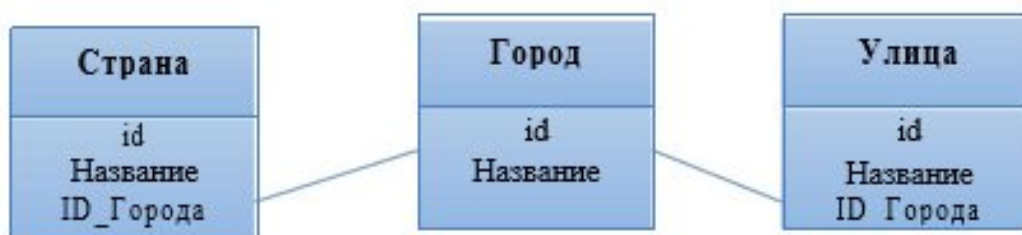
Методические рекомендации к выполнению работы

Лабораторная работа рассчитана на 2 занятия (2 недели, 1 занятие в неделю).
Итоговая оценка выставляется на последнем занятии.

Перед выполнением заданий, ознакомьтесь с теоретическими материалами по дисциплине. В заданиях, предусмотренных по вариантам.

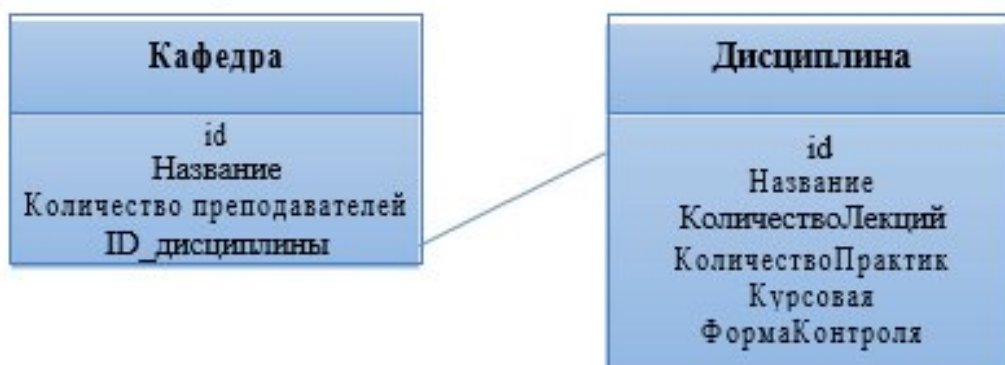
Вариант №1

1. Пусть дана база данных (приведена ниже). Используйте нужные структуры данных для ее хранения. Заполните БД. Выведите все страны, чье название начинается на букву А.
2. Для БД из задания 1, выведите все улицы, которые встречаются более чем в 5 городах.
3. Для БД из задания 6 выведите все улицы, для страны «РФ».



Вариант №2

1. Пусть дана база данных (приведена ниже). Используйте нужные структуры данных для ее хранения. Заполните БД. Выведите список дисциплин, по которым форма контроля – экзамен, а также предусмотрена курсовая работа.
2. Для БД из задания 1 выведите кафедры, которые читают более 5 дисциплин.
3. Для БД из задания 1 выведите данные о кафедрах с номерами id 10-20 (включительно).



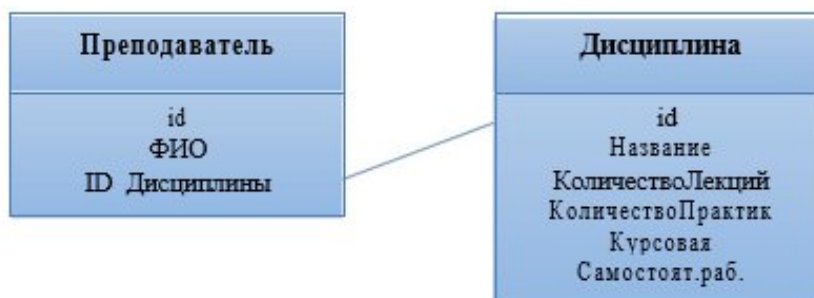
Вариант №3

1. Пусть дана база данных (приведена ниже). Используйте нужные структуры данных для ее хранения. Заполните БД. Выведите список дисциплин, которые проводятся в понедельник на 1 паре.
2. Для БД из задания 1 выведите те дисциплины, которые ведутся более чем на 1 паре.
3. Для БД из задания 1 выведите расписание занятий на все дни недели.



Вариант №4

1. Пусть дана база данных (приведена ниже). Используйте нужные структуры данных для ее хранения. Заполните БД. Выведите список всех преподавателей.
2. Для БД из задания 1 выведите список всех дисциплин преподавателя «Иванов И.И.», по которым предусмотрена курсовая работа и самостоятельная работа.
3. Для БД из задания 1 выведите все дисциплины, в чьем названии встречается буква «П».



Вариант №5

1. Пусть дана база данных (приведена ниже). Используйте нужные структуры данных для ее хранения. Заполните БД. Выведите список всех должностей для каждой фирмы.
2. Для БД из задания 1 выведите фирму, в которой работает больше всего сотрудников в возрасте от 20 до 30 лет.
3. Для БД из задания 1 выведите список всех сотрудников в должности «Директор» на фирме «НосковИко».



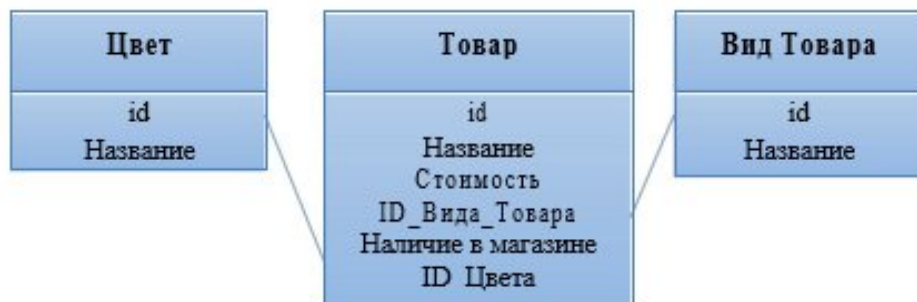
Вариант №6

1. Пусть дана база данных (приведена ниже). Используйте нужные структуры данных для ее хранения. Заполните БД. Выведите список всех меню.
2. Для БД из задания 1 выведите информацию об авторах, кто написал более 2 статей.
3. Для БД из задания 1 выведите количество написанных статей для каждого автора.



Вариант №7

1. Пусть дана база данных (приведена ниже). Используйте нужные структуры данных для ее хранения. Заполните БД. Выведите список товаров красного цвета, которые есть в наличии в магазине.
2. Для БД из задания 1 выведите хотя бы 2 товара одинакового вида, но разного цвета.
3. Для БД из задания 1 выведите виды товаров, для которых имеется товар разных цветов.



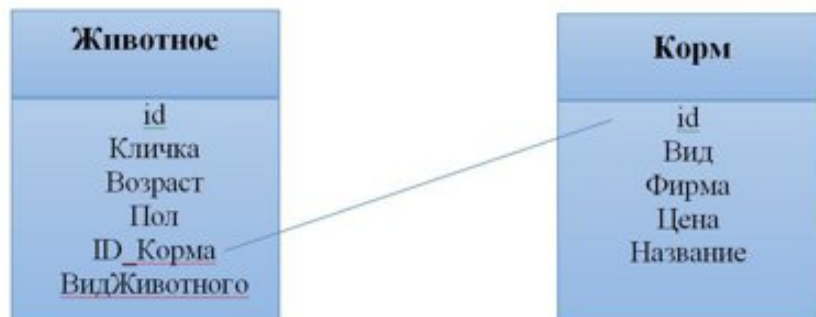
Вариант №8

1. Пусть дана база данных (приведена ниже). Используйте нужные структуры данных для ее хранения. Заполните БД. Выведите список детей в возрасте до 10 лет.
2. Для БД из задания 1 выведите информацию о детях, чьи родители имеют разные фамилии.
3. Для БД из задания 1 выведите список врачей в возрасте от 20 до 60 лет.



Вариант №9

1. Пусть дана база данных (приведена ниже). Используйте нужные структуры данных для ее хранения. Заполните БД. Выведите животных, чья кличка длиннее 5 символов.
2. Для БД из задания 1 выведите список кормов в порядке убывания их стоимости.
3. Для БД из задания 1 выведите список кормов, которым питается больше всего животных.



Вариант №10

1. Пусть дана база данных (приведена ниже). Используйте нужные структуры данных для ее хранения. Заполните БД. Выведите кафедру, на которой работает больше всего сотрудников.
2. Для БД из задания 1 выведите список кафедр в порядке убывания количества сотрудников.
3. Для БД из задания 1 выведите «самую молодую» кафедру (возраст кафедры = сложить возраст сотрудников и поделить на их количество).



Критерии оценивая:

- оценка «отлично» выставляется студенту, если задание выполнено в полном объеме, студент хорошо ориентируется в технологии разработки функций пользователя, программа логично и корректно демонстрирует работу всех функций;
- оценка «хорошо» выставляется студенту, если он выполнил задание почти в полном объеме (может отсутствовать пункт), допускает небольшие «проектно-технологические» ошибки в разработке функций;
- оценка «удовлетворительно» выставляется студенту, если он выполнил первые 8 пункта задания;
- оценка «неудовлетворительно» выставляется студенту, если он выполнил менее 8 пунктов задания.

3. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

ЭКЗАМЕН – 4 СЕМЕСТР ВОПРОСЫ ДЛЯ ПОДГОТОВКИ К ЭКЗАМЕНУ

ПК-2 Способен проектировать информационные системы

ПК-2.1. Моделирует прикладные процессы предметной области с учетом внедрения сквозных цифровых технологий и передовых ИТ-решений прикладной сферы

Обучающийся знает: - особенности разработки объектно-ориентированных программ и приемы объектно-ориентированного решения задач на языке Python.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам его участия в устных опросах и ответа на экзамене.

1. Встроенный класс String. Строковые операции.
2. Стандартные потоки ввода-вывода. Организация ввода и вывода данных. Класс Scanner.
3. Классы в языке Python: особенности реализации, определение класса.
4. Классы в языке Python: управление доступом к элементам класса; понятие пакета.
5. Классы в языке Python: поля класса.
6. Классы в языке Python: методы, конструкторы.
7. Интерфейсы в языке Python: определение интерфейса, реализация интерфейса.
8. Понятие Python API. Основные пакеты Python API.
9. События в Python: понятие события; типы событий; иерархия классов событий.
10. События в Python: модель делегирования событий.
11. События в Python: интерфейсы блоков прослушивания событий; способы реализации блока прослушивания.
12. События в Python: использование внутренних классов, классов-адаптеров для упрощения обработки событий.

ПК-2 Способен проектировать информационные системы

ПК-2.2. Разрабатывает модель информационной системы, в том числе с опорой на современные облачные решения

Обучающийся знает: возможности и преимущества языка Python для проектирования информационных систем разного назначения; особенности разработки объектно-ориентированных программ на языке Python.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам его участия в устных опросах и ответа на экзамене.

1. Массивы в Python: массивы простых типов и массивы объектов.
2. Оператор присваивания. Порядок действий (приоритет операторов).
3. Арифметические операторы. Операторы инкремента и декремента.
4. Встроенный класс Math. Псевдослучайные числа.
5. Операторы сравнения и логические операторы.
6. Операторы ветвления. Условный оператор. Минимизация количества проверок.

7. Операторы ветвления. Оператор множественного выбора. Его сравнение с условным оператором.

ПК-3 Способен разрабатывать информационные системы

ПК-3.1. Осуществляет кодирование на современных языках программирования, в том числе используя возможности специализированных цифровых платформ для индивидуальной и совместной разработки информационных систем

Обучающийся знает: синтаксис, типы данных и операторы языка Python, парадигму программ и подпрограмм.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам его участия в устных опросах и ответа на экзамене.

1. Особенности языка и платформы Python.
2. Классификация программ по типу исполнения (компилируемые, интерпретируемые, исполняемые на виртуальных машинах). Виртуальная машина Python. JIT-компиляция.
3. Создание простейшей программы на Python, её компиляция в байт-код и запуск.
4. Средства разработки Python-приложений. Интегрированные среды разработки.
5. Встроенные типы данных. Способы задания литералов различных типов.
6. Хранение данных в памяти ЭВМ.
7. Приведение типов (явное и автоматическое). Константы и переменные.
8. Типы Python-приложений, их особенности.
9. Типы данных в языке Python: простые и ссылочные типы.

ПК-3 Способен разрабатывать информационные системы

ПК-3.2. Проводит тестирование компонентов информационных систем, в том числе используя возможности специализированных цифровых платформ

Обучающийся знает: основные приемы тестирования компонентов информационных систем на языке Python.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам его участия в устных опросах и ответа на экзамене.

1. Исключения: понятие исключения; классы исключений; необходимость обработки исключений.
2. Исключения: операторы языка Python, используемые для обработки исключений.
3. Исключения: организация обработки исключений; определение собственных исключений.
4. Потоки вычислений: понятия процесса, потока.
5. Ввод/вывод в Python: основные понятия.
6. Ввод/вывод в Python: основные группы классов и интерфейсов пакета Python Tkinter.

ТИПОВЫЕ ЗАДАНИЯ И ЗАДАЧИ ДЛЯ ПОДГОТОВКИ К ЭКЗАМЕНУ

ПК-2 Способен проектировать информационные системы

ПК-2.1. Моделирует прикладные процессы предметной области с учетом внедрения сквозных цифровых технологий и передовых ИТ-решений прикладной сферы

Обучающийся умеет: проектировать компоненты информационных систем для обеспечения и анализа прикладных процессов на языке Python.

Обучающийся владеет: навыками применения стандартных библиотек классов при разработке компонентов информационных систем для обеспечения и анализа прикладных процессов на языке Python.

Задание 1. Разработать приложение для вывода должников библиотеки, если их фамилии и срок сдачи книг записаны в отдельный файл. Задолженность определять сравнением текущей даты с датой сдачи книг.

Задание 2. Разработать приложение для вывода фотографии сотрудника компании на отдельной форме, если фотографии хранятся в отдельном каталоге и названия файлов соответствуют фамилии сотрудника.

ПК-2 Способен проектировать информационные системы

ПК-2.2. Разрабатывает модель информационной системы, в том числе с опорой на современные облачные решения

Обучающийся умеет: проектировать программное обеспечение разного назначения (вычислительные системы, программы визуализации данных и т.п.) с применением библиотек языка Python (NumPy, Tkinter).

Обучающийся владеет: навыками анализа и выбора эффективных способов и средств разработки программных модулей информационных систем на языке Python для решения аналитических задач.

Задание 1. Спроектировать интерфейс и событийную модель приложения для тестирования знаний студентов по дисциплине «Информатика». В тесте должно быть не менее 3-х вопросов. Вопросы и результаты тестирования должны отражаться на разных формах. ФИО тестируемого и результат тестирования должны храниться в текстовом файле.

Задание 2. Спроектировать интерфейс и событийную модель приложения для анкетирования покупателя. Анкета ориентирована на вопросы о качестве обслуживания. Результаты вопросов должны храниться в БД.

ПК-3 Способен разрабатывать информационные системы

ПК-3.1. Осуществляет кодирование на современных языках программирования, в том числе используя возможности специализированных цифровых платформ для индивидуальной и совместной разработки информационных систем

Обучающийся умеет: разрабатывать объектно-ориентированные программы, используя классы и модули из библиотек языка Python для решения аналитических задач.

Обучающийся владеет: навыками применения объектно-ориентированной технологии и стандартных библиотек языка Python для решения задач вычислительного и аналитического характера.

Задание. Введите словарь городов. В качестве значения задайте кортеж, содержащий их географических координат. Программа должна выводить список трех городов, долгота или широта которых ближайшая до введенных.

ПК-3 Способен разрабатывать информационные системы

ПК-3.2. Проводит тестирование компонентов информационных систем, в том числе используя возможности специализированных цифровых платформ

Обучающийся умеет: составлять блоки тестов компонентов информационных систем на языке Python.

Обучающийся владеет: навыками применения тестов для информационных систем на языке Python.

Задание. Напишите класс Hand (рука), который хранит карты игрока. Класс Hand имеет методы: Взять карту в руку. Показать все карты из руки и их порядковые номера от 1 до 5 на экране, как это сделать с помощью только ООП.

ОБРАЗЕЦ ЭКЗАМЕНАЦИОННОГО БИЛЕТА

Экзаменационный билет состоит из одного теоретического вопроса и двух практических задания: первое на проектирование приложения (программы), второе – на программирование спроектированного приложения.

Тольяттинская Академия управления	Технология	Дисциплина	объектно-ориентированных приложений на Python
	разработки		
Вариант 1			
1. События в Python: использование внутренних классов, классов-адаптеров для упрощения обработки событий.			
2. Спроектировать событийную модель приложения для тестирования знаний студентов по любой из дисциплин состоящую из трех вопросов.			
3. По спроектированной модели приложения написать программный код.			

4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНОВАНИЯ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ

КРИТЕРИИ ОЦЕНИВАНИЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
ПК-2 Способен проектировать информационные системы					
ПК-2.1. Моделирует прикладные процессы предметной области с учетом внедрения сквозных цифровых технологий и передовых ИТ-решений прикладной сферы					
Знать: особенности разработки объектно-ориентированных программ и приемы объектно-ориентированн	Не знает	Фрагментарные знания особенностей разработки объектно-ориентированных программ и приемов объектно-	Общие, но не структурированные знания особенностей разработки объектно-ориентированных программ и приемов	Сформированные, но содержащие отдельные пробелы знания особенностей разработки объектно-	Сформированные систематизированные прочные знания особенностей разработки объектно-ориентированн

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
ого решения задач на языке Python		ориентированно го решения задач на языке Python	объектно-ориентированн ого решения задач на языке Python	ориентирован ных программ и приемов объектно-ориентирован ного решения задач на языке Python	ых программ и приемов объектно-ориентированн ого решения задач на языке Python
Уметь: проектировать семейства классов в заданной прикладной области и применять их для решения профессиональ ных задач на языке Python	Не умеет	Частично освоенные умения проектировать семейства классов в заданной прикладной области и применять их для решения профессиональн ых задач на языке Python	В целом освоенные, но не подкрепляемые знаниями и самостоятельно стью выполнения умения проектировать семейства классов в заданной прикладной области и применять их для решения профессиональ ных задач на языке Python	В целом сформированн ые и самостоятельн о выполняемые, но не всегда интеллектуаль но обоснованные умения проектировать семейства классов в заданной прикладной области и применять их для решения профессионал ьных задач на языке Python	Успешно сформированн ые, самостоятельно и осознанно выполняемые умения проектировать семейства классов в заданной прикладной области и применять их для решения профессиональ ных задач на языке Python
Владеть: навыками моделирования и описания моделей семейств классов	Не владеет	Фрагментарно сформированны е навыки моделирования и описания моделей семейств классов	В целом сформированны е, но выполняемые на элементарном уровне навыки моделирования и описания моделей семейств классов	Сформирован ные, но не устойчивые в сложных ситуациях навыки моделировани я и описания моделей семейств классов	Успешно сформированн ые, устойчивые, технологически грамотно выполняемые навыки моделирования и описания моделей семейств классов
ПК-2 Способен проектировать информационные системы ПК-2.2. Разрабатывает модель информационной системы, в том числе с опорой на современные облачные решения					
Знать: - возможности и	Не знает	Фрагментарные знания возможностей и	Общие, но не структурирован ные знания	Сформирован ные, но содержащие	Сформированн ые систематизиров

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
преимущества языка Python для проектирования информационных систем разного назначения; - основные приемы создания компонентов информационных систем для обеспечения и анализа прикладных процессов		преимущества языка Python для проектирования информационных систем разного назначения, особенностей разработки объектно-ориентированных программ на языке Python	возможностей и преимуществ языка Python для проектирования информационных систем разного назначения, особенностей разработки объектно-ориентированных программ на языке Python	отдельные пробелы знания возможностей и преимуществ языка Python для проектирования информационных систем разного назначения, особенностей разработки объектно-ориентированных программ на языке Python	аннйе прочные знания возможностей и преимуществ языка Python для проектирования информационных систем разного назначения, особенностей разработки объектно-ориентированных программ на языке Python
Уметь: - проектировать программное обеспечение разного назначения (вычислительные системы, программы визуализации данных и т.п.) с применением библиотек языка Python (NumPy, Tkinter)	Не умеет	Частично освоенные умения проектировать программное обеспечение разного назначения (вычислительные системы, программы визуализации данных и т.п.) с применением библиотек языка Python (NumPy, Tkinter)	В целом освоенные, но не подкрепляемые знаниями и самостоятельно стью выполнения умения проектировать программное обеспечение разного назначения (вычислительные системы, программы визуализации данных и т.п.) с применением библиотек языка Python (NumPy, Tkinter)	В целом сформированные и самостоятельное выполняемые, но не всегда интеллектуально обоснованные умения проектировать программное обеспечение разного назначения (вычислительные системы, программы визуализации данных и т.п.) с применением библиотек языка Python (NumPy, Tkinter)	Успешно сформированные, самостоятельно и осознанно выполняемые умения проектировать программное обеспечение разного назначения (вычислительные системы, программы визуализации данных и т.п.) с применением библиотек языка Python (NumPy, Tkinter)
Владеть:	Не владеет	Фрагментарно	В целом	Сформирован	Успешно

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
навыками анализа и выбора эффективных способов и средств разработки программных модулей информационных систем на языке Python для решения аналитических задач		сформированные навыки анализа и выбора эффективных способов и средств разработки программных модулей информационных систем на языке Python для решения аналитических задач	сформированные, но выполняемые на элементарном уровне навыки анализа и выбора эффективных способов и средств разработки программных модулей информационных систем на языке Python для решения аналитических задач	ные, но не устойчивые в сложных ситуациях навыки анализа и выбора эффективных способов и средств разработки программных модулей информационных систем на языке Python для решения аналитических задач	сформированные, устойчивые, технологически грамотно выполняемые навыки анализа и выбора эффективных способов и средств разработки программных модулей информационных систем на языке Python для решения аналитических задач
ПК-3 Способен разрабатывать информационные системы ПК-3.1. Осуществляет кодирование на современных языках программирования, в том числе используя возможности специализированных цифровых платформ для индивидуальной и совместной разработки информационных систем					
Знать: - синтаксис, типы данных и операторы языка Python, парадигму программ и подпрограмм	Не знает	Фрагментарные знания синтаксиса, типов данных и операторов языка Python, парадигм программ и подпрограмм	Общие, но не структурированные знания синтаксиса, типов данных и операторов языка Python, парадигм программ и подпрограмм	Сформированные, но содержащие отдельные пробелы знания синтаксиса, типов данных и операторов языка Python, парадигм программ и подпрограмм	Сформированные систематизированные прочные знания синтаксиса, типов данных и операторов языка Python, парадигм программ и подпрограмм
Уметь: - разрабатывать объектно-ориентированные программы, используя классы и модули из библиотек языка Python для решения	Не умеет	Частично освоенные умения разрабатывать объектно-ориентированные программы, используя классы и модули из библиотек	В целом освоенные, но не подкрепляемые знаниями и самостоятельно выполнению умения разрабатывать объектно-	В целом сформированные и самостоятельное выполняемые, но не всегда интеллектуально обоснованные умения	Успешно сформированные, самостоятельно и осознанно выполняемые умения разрабатывать объектно-ориентированные программы,

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
аналитических задач		языка Python для решения аналитических задач	ориентированные программы, используя классы и модули из библиотек языка Python для решения аналитических задач	разрабатывать объектно-ориентированные программы, используя классы и модули из библиотек языка Python для решения аналитических задач	используя классы и модули из библиотек языка Python для решения аналитических задач
Владеть: - навыками применения объектно-ориентированной технологии и стандартных библиотек языка Python для решения задач вычислительного и аналитического характера	Не владеет	Фрагментарно сформированные навыки применения объектно-ориентированной технологии и стандартных библиотек языка Python для решения задач вычислительного и аналитического характера	В целом сформированные, но выполняемые на элементарном уровне навыки применения объектно-ориентированной технологии и стандартных библиотек языка Python для решения задач вычислительного и аналитического характера	Сформированные, но не устойчивые в сложных ситуациях навыки применения объектно-ориентированной технологии и стандартных библиотек языка Python для решения задач вычислительного и аналитического характера	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки применения объектно-ориентированной технологии и стандартных библиотек языка Python для решения задач вычислительного и аналитического характера
ПК-3 Способен разрабатывать информационные системы					
ПК-3.2. Проводит тестирование компонентов информационных систем, в том числе используя возможности специализированных цифровых платформ					
Знать: основные приемы тестирования компонентов информационных систем на языке Python	Не знает	Фрагментарные знания основных приемов тестирования компонентов информационных систем на языке Python	Общие, но не структурированные знания основных приемов тестирования компонентов информационных систем на языке Python	Сформированные, но содержащие отдельные пробелы знания основных приемов тестирования компонентов информационных систем на	Сформированные систематизированные прочные знания основных приемов тестирования компонентов информационных систем на языке Python

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
				языке Python	
Уметь: составлять блоки тестов компонентов информационных систем на языке Python	Не умеет	Частично освоенные умения составлять блоки тестов компонентов информационных систем на языке Python	В целом освоенные, но не подкрепляемые знаниями и самостоятельно выполнению умения составлять блоки тестов компонентов информационных систем на языке Python	В целом сформированные и самостоятельное выполняемые, но не всегда интеллектуально обоснованные умения составлять блоки тестов компонентов информационных систем на языке Python	Успешно сформированные, самостоятельно и осознанно выполняемые умения составлять блоки тестов компонентов информационных систем на языке Python
Владеть: навыками применения тестов для информационных систем на языке Python	Не владеет	Фрагментарно сформированные навыки применения тестов для информационных систем на языке Python	В целом сформированные, но выполняемые на элементарном уровне навыки применения тестов для информационных систем на языке Python	Сформированные, но не устойчивые в сложных ситуациях навыки применения тестов для информационных систем на языке Python	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки применения тестов для информационных систем на языке Python

ПРОЦЕДУРА ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ И КРИТЕРИИ ОЦЕНКИ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Для оценки результатов обучения по дисциплине учебным планом предусмотрен экзамен, проводимый в письменной или устной формах. При устном ответе обязательным является подготовка студентом простого или развернутого ответа. При выставлении итоговой оценки за дисциплину необходимо учитывать результаты текущего контроля (или: результаты текущего контроля являются допуском к экзамену) и критерии оценивания сформированности компетенций:

- оценка «отлично» выставляется студенту за такие знания, когда обучающийся показывает всестороннее, систематическое и глубокое усвоение всего объема учебно-программного материала, умеет свободно выполнять задания, предусмотренные программой курса, осмысленно применяет полученные знания на практике, усвоивший основную и знакомый с дополнительной литературой, рекомендованной программой курса, не допускает ошибок

при воспроизведении знаний в объеме пройденного программного материала, грамотное и логически стройное изложение материала при ответе, а также в письменных работах и выполняет последние уверенно и аккуратно, легко отвечает на видоизмененные вопросы, на которых нет прямых ответов в учебной литературе; практическое задание (заданий) выполнено безошибочно и в полном соответствии с требованиями, обучающийся демонстрирует устойчивые навыки работы и осознанно исполняемые действия для разработки программных продуктов, способность вносить коррективы и обосновывать свои действия;

- оценка «хорошо» выставляется студенту тогда, когда обучающийся выявляет полное знание учебно-программного материала, отвечает без особых затруднений на вопросы преподавателя, успешно выполняет предусмотренные в программе курса задания, знакомый с основной литературой, рекомендованной программой курса, умеет применять полученные знания на практике, в устных ответах не допускает серьезных ошибок и легко устраняет отдельные неточности с помощью дополнительных вопросов преподавателя, в письменных работах делает незначительные ошибки; практическое задание (задания) выполнено с незначительными замечаниями и ошибками, в соответствии с требованиями, обучающийся демонстрирует навыки работы и осознанно исполняемые действия для разработки программных продуктов, но затрудняется внести коррективы;
- оценка «удовлетворительно» обучающийся обнаруживает усвоение основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и дальнейшей профессиональной деятельности, справляющийся с выполнением предусмотренных в программе курса заданий, но испытывает затруднение при его самостоятельном воспроизведении и требует дополнительных и уточняющих вопросов преподавателя, предпочитает отвечать на вопросы воспроизводящего характера и путается при ответах на видоизмененные вопросы, умеет применять полученные знания на практике, знакомый с основной литературой, рекомендованной программой курса, допускает ошибки в письменных работах; практическое задание (задания) выполнено с грубыми ошибками и множеством замечаний, с отклонениями от требований, обучающийся демонстрирует непрочные навыки работы, затрудняется в объяснении применяемых средств и конструктивов для разработки программных продуктов, но способен внести отдельные коррективы в работу;
- оценка «неудовлетворительно» выставляется студенту, обнаружившему в знаниях учебно-программного материала, принципиальные ошибки в выполнении предусмотренных программой курса заданий, в письменных работах допускаются грубые ошибки; практическое задание (задания) не выполнено (программа не работает) или выполнено со значительными отклонениями от требований, с грубыми ошибками и множеством замечаний, обучающийся демонстрирует частично сформированные навыки работы, затрудняется в объяснении применяемых средств и конструктивов для разработки программных продуктов, не способен внести коррективы в работу.

5. ДИАГНОСТИЧЕСКИЕ ЗАДАНИЯ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ УРОВЕНЬ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Компетенция ПК-2 Способен проектировать информационные системы.

Обучающийся знает: особенности разработки объектно-ориентированных программ и приемы объектно-ориентированного решения задач на языке Python; возможности и преимущества языка Python для проектирования информационных систем разного назначения; основные приемы создания компонентов информационных систем для обеспечения и анализа прикладных процессов.

Обучающийся умеет: проектировать семейства классов в заданной прикладной области и применять их для решения профессиональных задач на языке Python; проектировать программное обеспечение разного назначения (вычислительные системы, программы визуализации данных и т.п.) с применением библиотек языка Python (NumPy, Tkinter).

Обучающийся владеет: навыками моделирования и описания моделей семейств классов; навыками анализа и выбора эффективных способов и средств разработки программных модулей информационных систем на языке Python для решения аналитических задач.

ЗАДАНИЯ ЗАКРЫТОГО ТИПА

Задание 1. Конструктор класса задается методом с именем...

- a) `__new__`
- b) `__construct__`
- c) `__init__`
- d) `__len__`

Ответ: c

Задание 2. Сколько конструкторов в классе может создать в Python?

- a) ноль, в них нет необходимости
- b) два, если в этом есть необходимость
- c) один, в Python можно создать только один конструктор
- d) Бесконечно много

Ответ: c

Задание 3. Что выдаст следующий код:

```
class Test1:
    def info(self):
        print("Это класс-родитель")
class Test2(Test1):
    def info(self):
        print("Это класс-потомок")

test=Test2()
test.info()
```

- a) Это класс-родитель
- b) Это класс-потомок
- c) `SyntaxError: invalid syntax`
- d) Это класс-родитель Это класс-потомок

Ответ: b

Задание 4. Ключевое слово *self* выполняет...

- a) обращение к классу-предку
- b) резервное копирование определенного файла
- c) обращение к параметрам класса
- d) обращение к атрибутам объекта внутри класса и его методам

Ответ: d

Задание 5. Дан класс

```
class Car:  
    model = 'lada'
```

Как правильно обратиться к атрибуту класса и вывести его значение?

- a) `print(Car.model)`
- b) `print(model)`
- c) `print(Car)`
- d) `print(Car.lada)`

Ответ: a

Задание 6. Дан класс

```
class Calculation:  
    def mult(self, x, y):  
        print(x*y)
```

```
c = Calculation ()
```

Какой вызов метода `mult` является правильным?

- a) `Calculation.mult(5,10)`
- b) `c.mult(c,5,10)`
- c) `c.mult(5,10)`
- d) `mult(c,5,10)`

Ответ: c

Задание 7. Какой из перечисленных вариантов является верным объявлением `private` поля `val`?

- a) `private val = 0`
- b) `val = 0`
- c) `_val = 0`
- d) `__val = 0`

Ответ: d

Задание 8. Какой результат выдаст данный код?

```
class Product:  
    price = 1000  
class Stock:  
    price = 2000  
class Shop:  
    price = 3000  
class tovar (Stock, Product, Shop):
```

```

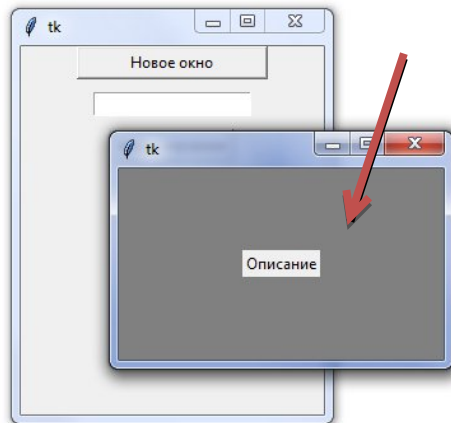
pass
test = tovar ()
print(test. price)

```

- a) 1000
- b) 2000
- c) 3000
- d) ошибка

Ответ: b

Задание 9. В приложение содержащее главное окно нужно добавить ещё одно окно с описанием программы. От какого класса оно будет порождаться



- a) Tk()
- b) Toplevel()
- c) Ttk()
- d) Window()

Ответ: b

Задание 10. В Tkinter существуют менеджеры геометрии для расположения виджетов в окне. Укажите все менеджеры геометрии.

- a) pack()
- b) coordinates()
- c) grid()
- d) location()
- e) place()
- f) puck()

Ответ: a, c, e

ЗАДАНИЯ ОТКРЫТОГО ТИПА (15 заданий, из них 5 – с развернутым ответом)

Задание 1. Дан код программы:

```

class Rect:
    def __init__(self, color="green", width=100, height=100):
        self.color = color
        self.width = width
        self.height = height
    def square(self):

```

```
return self.width * self.height
```

```
r1=Rect()  
r1.color  
print(r1.color)  
print(r1.square())
```

- a) Запишите имя класса.
- b) Запишите имя экземпляра класса.
- c) Перечислите атрибуты класса.
- d) Запишите строку обращения к методу вычисления площади.
- e) Какой результат выдаст строка вызова метода вычисления площади.

Ответ:

- a) *Rect*
- b) *r1*
- c) *color, width, height*
- d) *print(r1.square())*
- e) *10000*

Задание 2. Сопоставьте методы перегрузки математических операторов с их описанием.

Метод		Описание	
1	x.__sub__(x)	А	сложение
2	x.__mul__(x)	Б	остаток от деления
3	x.__mod__(x)	В	возведение в степень
4	x.__add__(x)	Г	умножение
5	x.__truediv__(x)	Д	вычитание
6	x.__pow__(x)	Е	деление

Ответ: 1-Д, 2-Г, 3-Б, 4-А, 5-Е, 6-В

Задание 3. Перечислите и опишите атрибуты и методы обеспечивающие приватность в классах Python.

Ответ:

В Python есть три типа атрибутов и методов: **public**, **protected** и **private**. Они определяют уровень доступности к атрибутам и методам внутри класса и за его пределами (извне).

Public атрибуты и методы доступны из любого места программы. Это самые обычные атрибуты и методы, без нижних подчёркиваний в начале.

Protected атрибуты и методы начинаются с одного символа подчеркивания (**_**). Атрибуты такого типа, не отличаются по функциональности от публичных, они лишь сигнализируют программисту, что такие атрибуты нужно использовать внутри класса и его наследников.

Private атрибуты и методы начинаются с двух символов подчеркивания (__). Private атрибуты и методы доступны только внутри класса, и используются для полной инкапсуляции данных и функциональности класса. Они недоступны извне класса и его наследников, т.е. их нельзя напрямую вызвать, например функцией print. Это обеспечивает высокий уровень защиты данных и предотвращает их случайное изменение или использование.

Задание 4. Что такое итерируемый объект? Перечислите итерируемые объекты в Python.

Ответ: Итерируемый объект (iterable) - это объект, который способен возвращать элементы по одному. Кроме того, это объект, из которого можно получить итератор.

Примеры итерируемых объектов: список, строка, кортеж, словари, файлы

Задание 5. Расположите в правильном порядке этапы создания GUI-программы

1. Определить события, то есть то, на что будет реагировать программа.
2. Расположить виджеты в главном окне.
3. Создать виджеты и выполнить конфигурацию их свойств (опций).
4. Создать главное окно.
5. Описать обработчики событий, то есть то, как будет реагировать программа.
6. Запустить цикл обработки событий.

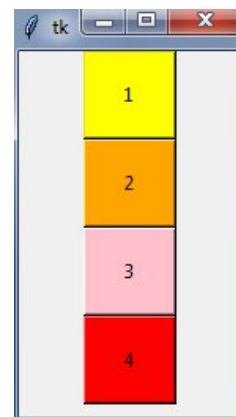
Ответ: 4, 3, 1, 5, 2, 6

Задание 6. В программе созданы 4 кнопки:

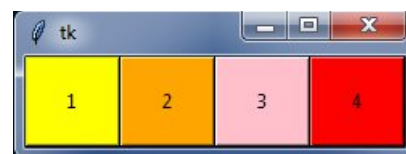
```
from tkinter import *
root=Tk()
```

```
b1 = Button(width=7, height=3, bg='yellow', text="1")
b2 = Button(width=7, height=3, bg='orange', text="2")
b3 = Button(width=7, height=3, bg='pink', text="3")
b4 = Button(width=7, height=3, bg='red', text="4")
```

```
b1.pack()
b2.pack()
b3.pack()
b4.pack()
```



Какой параметр нужно добавить в метод pack(), чтобы кнопки расположились горизонтально:



Ответ: side=LEFT

Задание 7. Какие существуют три способа конфигурирования свойств виджетов в Tkinter?

Ответ:

- в момент создания объекта
- с помощью метода config
- обращение к свойству как к элементу словаря.

Задание 8. Сопоставьте свойства виджетов с их описанием.

Свойство		Описание	
1	width	А	устанавливается надпись в виджете

2	bg	Б	устанавливается цвет текста
3	text	В	устанавливается высота виджета
4	height	Г	устанавливается цвет фона виджета
5	font	Д	устанавливается ширина виджета
6	fg	Е	устанавливается шрифт

Ответ: 1-Д, 2-Г, 3-А, 4-В, 5-Е, 6-Б

Задание 9. Как в Python создать и использовать собственный модуль?

Ответ: Создание и использование собственных модулей в Python — удобный способ организации кода, который позволяет разделить логику на отдельные компоненты и повторно использовать код.

Для создания модуля необходимо создать файл с расширением .py, в котором будут храниться все функции и классы, относящиеся к модулю (например, my_module.py). Чтобы использовать функции из модуля my_module в другой программе, нужно импортировать модуль с помощью ключевого слова *import*. Если вам нужны только определенные функции из модуля, можно импортировать только их, используя конструкцию *from ... import ...*

Задание 10. Расставьте этапы разработки объектно-ориентированной программы в правильном порядке

- 1) Проектирование классов, на основе которых будут создаваться объекты. В случае необходимости установление между классами наследственных связей.
- 2) Создание объектов.
- 3) Определение объектов, участвующих в ее решении.
- 4) Создание классов, определение их полей и методов.
- 5) Формулирование задачи.
- 6) Решение задачи путем организации взаимодействия объектов.
- 7) Определение ключевых для данной задачи свойств и методов объектов.

Ответ: 5, 3, 1, 7, 4, 2, 6

-----**Владеть**-----

Задание 11. Напишите код по описанию, сопровождая комментариями:

Создать класс «Студент». Класс содержит имя студента *full_name*, номер группы *group_number* и список полученных оценок *progress*. В программе вводится список студентов.

Ответ: Один из вариантов решения

class Student:

```

def __init__(self, full_name="", group_number=0, progress=[]): # конструктор
    self.full_name = full_name # имя
    self.group_number = group_number # номер группы
    self.progress = progress # оценки
def __str__(self): # печатаемое представление экземпляра класса
    txt = 'Студент: ' + self.full_name + ' Группа: ' + self.group_number
    txt += ' Оценки:'
    for x in self.progress:
        txt += ' ' + str(x) # добавляем список оценок
    return txt

```

Задание 12. Что такое деструктор в Python?

Ответ: Перед уничтожением экземпляра автоматически вызывается метод, называемый деструктором или финализатором. В языке Python деструктор реализуется в виде предопределенного метода `__del__()`. Метод не будет вызван, если на экземпляре класса существует хотя бы одна ссылка.

Но поскольку интерпретатор самостоятельно заботится об удалении объектов, использование деструктора в языке Python не имеет особого смысла.

Задание 13. Какой результат выдаст данный код?

```
class Parents:
    name1 = 'King'
    name2 = 'Queen'

class Daughter(Parents):
    name1 = 'Elsa'
    name2 = 'Anna'
```

```
name_is = Parents()
print(name_is.name1)
```

Ответ: King

Задание 14. Какой результат выдаст данный код?

```
class Words:
    def __init__(self, word):
        self.word = word
    def __eq__(self, other):
        return self.word == other.word

word_1 = Words('Cat')
word_2 = Words('Dog')
print(word_1 != word_2)
```

Ответ: True

Задание 15. Допишите код так, чтобы в конце, на экране появилось слово snowman.

```
class Present:
    def __init__(self):
        self.present = ['book', 'Iphone', 'TV', 'snowman', 'car']

# добавленные методы

holiday = Present()
if len(holiday) == 5:
    print(holiday[3])
```

Ответ:

```
class Present:
    def __init__(self):
        self.present = ['book', 'Iphone', 'TV', 'snowman', 'car']

# добавленные методы
def __len__(self):
```

```
    return len(self.present)
def __getitem__(self, item):
    return self.present[item]
```

```
holiday = Present()
if len(holiday) == 5:
    print(holiday[3])
```

ПК-3 Способен разрабатывать информационные системы.

Обучающийся знает: синтаксис, типы данных и операторы языка Python, парадигму программ и подпрограмм; основные приемы тестирования компонентов информационных систем на языке Python.

Обучающийся умеет: разрабатывать объектно-ориентированные программы, используя классы и модули из библиотек языка Python для решения аналитических задач. составлять блоки тестов компонентов информационных систем на языке Python.

Обучающийся владеет: навыками применения объектно-ориентированной технологии и стандартных библиотек языка Python для решения задач вычислительного и аналитического характера; навыками применения тестов для информационных систем на языке Python.

ЗАДАНИЯ ЗАКРЫТОГО ТИПА

Задание 1. Какой тип данных отобразит команда `print(type(-18))`?

- e) *float*
- f) *int*
- g) *bool*
- h) *str*
- i) *complex*

Ответ: b

Задание 2. Как называется присваивание вида `a = b = c = 0`?

- a) *Каскадное присваивание*
- b) *Множественное присваивание*
- c) *цепное присваивание*
- d) *такая запись не допустима*

Ответ: a

Задание 3. Какую команду следует записать, чтобы подключить модуль `math` к программе?

- a) *math*
- b) *import from math*
- c) *import math*
- d) *from math*
- e) *using System.math*

Ответ: c

Задание 4. Что возвращает файловый метод `readlines`?

- a) *список из всех строк файла*
- b) *число строк в файле*

- c) прочитанные строки из файла в виде многострочной строки
- d) одну текущую прочитанную строку из файла

Ответ: a

Задание 5. Что делает команда `pip list`?

- a) отображает список команд для функции `pip`
- b) отображает список импортированных в программу модулей
- c) отображает список установленных модулей для текущего интерпретатора
- d) показывает номер версии языка Python

Ответ: c

Задание 6. Когда выполняется блок `finally` в конструкции `try / except / finally`?

- a) всегда сразу после блока `try`
- b) всегда после выполнения блоков `try` и `except`
- c) после выполнения блоков `try` и `except`, если не возникало ошибок
- d) сразу после блока `try`, если не было ошибок

Ответ: b

Задание 7. Что такое рекурсивная функция?

- a) это функция, которая вызывает саму себя
- b) это функция, которая вызывается в сторонних рекурсивных алгоритмах
- c) это функция, которая реализует рекурсивный алгоритм через операторы циклов
- d) любая функция также является и рекурсивной

Ответ: a

Задание 8. Дан список `s = [-8, False, "Сапара", 5, "ТЛТ", True]`

По какому индексу можно обратиться к значению 5 списка. Отметьте все возможные варианты.

- a) 4
- b) 3
- c) -2
- d) -4
- e) -3
- f) 5

Ответ: b, e

Задание 9. Для чего используется ключевое слово **`global`**?

- a) чтобы локальное окружение делать глобальным
- b) чтобы менять глобальные переменные в локальном окружении (например, внутри функций)
- c) чтобы любую локальную переменную превращать в глобальную
- d) чтобы глобальную переменную превращать в локальную

Ответ: b

Задание 10. Что произойдет при выполнении кода:

```
f = open("file.txt", encoding="utf-8")  
x = f.read(4)
```

- a) из файла *file.txt* будут прочитаны первые 4 символа (символы могут состоять из нескольких байтов)
- b) в файл *file.txt* будет записано число 4
- c) из файла *file.txt* будут прочитаны первые 4 байта
- d) позиция файла будет установлена на четвертый символ

Ответ: a

ЗАДАНИЯ ОТКРЫТОГО ТИПА (15 заданий, из них 5 – с развернутым ответом)

Уметь

Задание 1. Сопоставьте методы списков и их описание.

Метод		Описание	
1	append	А	Очищает список (удаляет все элементы)
2	insert	Б	Меняет порядок следования элементов на обратный
3	remove	В	Возвращает индекс первого найденного элемента
4	pop	Г	Возвращает число элементов с указанным значением
5	clear	Д	Добавляет элемент в конец списка
6	count	Е	Вставляет элемент в указанное место списка
7	index	Ж	Удаляет последний элемент, либо элемент с указанным индексом
8	reverse	З	Удаляет элемент по значению

Ответ: 1-Д, 2-Е, 3-З, 4-Ж, 5-А, 6-Г, 7-В, 8-Б

Задание 2. Дан код программы:

```
def NOK(a,b):
    m=a*b
    while a!=0 and b!=0:
        if a>b:
            a%=b
        else:
            b%=a
    return m//(a+b)
while 1:
    x=int(input("a="))
    y=int(input("b="))
    if x!=0:
        print('НОК=',NOK(x,y))
    else:
        break
```

- f) Запишите имя функции.
- g) Запишите строку объявления функции.
- h) Запишите строку, в которой вызывается функция .

- i) Перечислите локальные параметры.
- j) Перечислите глобальные аргументы.
- k) Зачем в программе цикл While?
- l) При каком условии будет происходить выход из цикла?

Ответ:

- a) *NOK*
- b) *def NOK(a,b):*
- c) *print('HOK=',NOK(x,y))*
- d) *a, b, m*
- e) *x, y*
- f) *Бесконечный цикл в программе для того, чтобы пользователю не приходилось перезапускать программу для ввода новых значений.*
- g) *Выход из цикла произойдёт, когда пользователь введёт значение 0 для переменной x.*

Задание 3. Напишите функцию с именем `dl_str`, которая принимает строку (в качестве аргумента) и возвращает `False`, если длина строки меньше 5 символов. Иначе возвращается значение `True`.

Ответ:

```
def dl_str(s):
    if len(s)>=5:
        t=True
    else:
        t=False
    return t
```

Задание 4. Напишите анонимную (лямбда) функцию с двумя параметрами для деления одного целого числа на другое. Если происходит деление на ноль, то функция должна возвращать значение `None`, иначе - результат деления. Свяжите эту функцию переменной `dv`.

Ответ:

```
dv = lambda x,y: None if x == 0 or y == 0 else x/y
```

Задание 5. Дан фрагмент программы:

```
f = open("abc.txt")
r = f.read(1)
f.close()
```

При его выполнении возникает ошибка `FileNotFoundError`. Запишите конструкцию `try / except`, чтобы отображалось сообщение "File Not Found", если файл не удастся открыть.

Ответ:

```
try:
    f = open("abc.txt")
    r = f.read(1)
    f.close()
except FileNotFoundError:
    print("File Not Found")
```

Задание 6. Дан фрагмент программы:

```
file = open("my_file.txt")
s = file.readline()
```

```
file.close()
```

Оформите этот код с помощью менеджера контекста.

Ответ: `with open("my_file.txt") as file: file.readline()`

Или

```
with open("my_file.txt", encoding='utf-8') as file:
    s = file.readline()
```

Задание 7. На вход программы поступают два целых числа a и b ($a < b$), записанные в одну строчку через пробел. На их основе запишите генератор для формирования квадратов чисел в диапазоне $[a; b]$. Преобразуйте этот генератор в кортеж чисел (без использования операторов циклов) и присвойте эту коллекцию переменной `tup`.

Ответ:

```
a, b = map(int, input().split()) # ввод значений a и b в одну строку
gen=(x**2 for x in range(a,b+1))
tup=tuple(gen)
```

Задание 8. Запишите функцию без аргументов, которая считывает с клавиатуры имя и фамилию, записанные в одну строку через пробел, и выводит на экран сообщение (без кавычек): "Уважаемый, «имя» «фамилия»! Вы верно выполнили это задание!" Запишите вызов этой функции.

Ответ:

```
def func():
    im,fam=input().split()
    print(f"Уважаемый, {im} {fam}! Вы верно выполнили это задание!")
func()
```

Задание 9. Прокомментируйте следующий код:

```
a = input("Введи число: ")
i = -1
c = ""
while i >= -1 * len(a):
    c = c + a[i]
    i = i - 1
print(c)
```

Что будет получено в результате?

Ответ:

```
a = input("Введи число: ") #нужно ввести многозначное число
i = -1 #задаётся начальное значение для цикла - это будут индексы цифр входящих в число
c = "" #объявляется пустая строковая переменная
while i >= -1 * len(a): #объявляется цикл, в котором перебираются индексы цифр входящих в
число справа налево
    c = c + a[i] # "склеиваем" цифры, получая "обратное" число
    i = i-1 # счётчик цикла
print(c) #вывод "обратного" числа
```

Результат. Вводится многозначное целое число. Выводится число, обратное по порядку составляющих его цифр введенному. Например, введено 1234, будет выводиться 4321.

Задание 10. Составьте код по следующему описанию:

С клавиатуры вводится целое число X. Если X больше 0, то вывести 1, а если X меньше 0, то -1. Если X равен 0, то вывести 0.

Ответ: один из возможных вариантов оформления

```
X = int(input("Введи число: "))
```

```
if X > 0:
```

```
    X = 1
```

```
elif X < 0:
```

```
    X = -1
```

```
else:
```

```
    X = 0
```

```
print(X)
```

-----**Владеть**-----

Задание 11. Что представляет собой переменная в Python?

Ответ: Переменная в Python – это ссылка на объект и можно обращаться к этому объекту (хранилищу) по имени этой ссылки.

Задание 12. Как в Python объявляются переменные?

Ответ: Python поддерживает динамическую типизацию. То есть значения присваиваются переменным не при компиляции, а построчно во время выполнения программы.

Поэтому в Python объявлять переменную отдельно не нужно — достаточно сразу её инициализировать, то есть присвоить значение, используют знак равно =.

Задание 13. С клавиатуры вводится натуральное число n. С помощью цикла for найти все делители этого числа. Найденные делители выводить сразу в столбик без формирования списка.

Ответ:

```
n = int(input())
```

```
for i in range(1, n + 1):
```

```
    if n % i == 0:
```

```
        print(i)
```

Задание 14. Для чего используется функция isinstance и в чём её отличие от функции type?

Ответ: В Python функции type() и isinstance() используются для проверки типа объекта. Однако есть несколько ключевых различий в их использовании и поведении.

Функция type() возвращает тип объекта, который передается ей в качестве аргумента. Это обычно используется для определения, к какому классу принадлежит определенный объект.

Например, можно проверить, является ли переменная целым числом:

```
x = 10
```

```
print(type(x) == int) # вернет True
```

Функция isinstance() проверяет, является ли объект экземпляром указанного класса или его подкласса. Если требуется проверить, принадлежит ли объект к определенному классу или его подклассу, лучше использовать isinstance().

Например, можно проверить, является ли переменная экземпляром класса int или его подкласса:

```
x = 10
```

```
print(isinstance(x, int)) # вернет True
```

Основное различие между этими двумя функциями заключается в том, что type() не учитывает наследование (то есть не считает объект экземпляром класса, если он является экземпляром подкласса), в то время как isinstance() это делает.

Задание 15. Найдите все ошибки в тексте программы и опишите их подробно. Перепишите текст программы безошибочно.

```
1) x=input("1-прямоугольник,2-круг")
2) if x=="1":
3)     n=int(input("Введите ширину:"))
4)     m=int(input("Введите высоту:"))
5)     print("Площадь",n*m)
6) elif x=="2"
7)     r=input("Введите радиус"))
8) print("Площадь",3,14*k^2)
```

Ответ:

Строка 2) – сравнение оформлено не верно. Должно быть `if x=="1":`

Строка 6) – не хватает двоеточия. Должно быть `elif x=="2":`

Строка 7) – не хватает функции `int`. Должно быть `r=int(input("Введите радиус"))`

Строка 8) – не верно оформлена дробь Должно быть `3.14`

Не верно оформлено возведение в степень.

Не объявлена переменная `k`, должна быть `г`

Эта строка должна быть сдвинута вправо

```
print("Площадь",3.14*r**2)
```

```
1) x=input("1-прямоугольник,2-круг")
2) if x=="1":
3)     n=int(input("Введите ширину:"))
4)     m=int(input("Введите высоту:"))
5)     print("Площадь",n*m)
6) elif x=="2":
7)     r=int(input("Введите радиус"))
8)     print("Площадь",3.14*r**2)
```

6. ЛИСТ СОГЛАСОВАНИЯ

Составил:

Н.А. Иванова, ст.преп



(подпись)

Заведующий кафедрой

Н.Б. Стрекалова, д.п.н., доцент



(подпись)

Заведующий выпускающей кафедрой

Н.Б. Стрекалова, д.п.н., доцент



(подпись)