

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ  
ПО ДИСЦИПЛИНЕ**  
Разработка приложений на языке PYTHON

**Код модуля**  
1160957(1)

**Модуль**  
Разработка приложений на языке PYTHON

**Екатеринбург**

Оценочные материалы составлены автором(ами):

<b>№ п/п</b>	<b>Фамилия, имя, отчество</b>	<b>Ученая степень, ученое звание</b>	<b>Должность</b>	<b>Подразделение</b>
1	Бабич Михаил Владимирович	кандидат технических наук, без ученого звания	Доцент	радиоэлектроники и телекоммуникаций

**Согласовано:**

Управление образовательных программ

Т.Г. Комарова

**Авторы:**

- **Бабич Михаил Владимирович, Доцент, радиоэлектроники и телекоммуникаций**

**1. СТРУКТУРА И ОБЪЕМ ДИСЦИПЛИНЫ Разработка приложений на языке PYTHON**

<b>1.</b>	<b>Объем дисциплины в зачетных единицах</b>	3	
<b>2.</b>	<b>Виды аудиторных занятий</b>	Лекции Лабораторные занятия	
<b>3.</b>	<b>Промежуточная аттестация</b>	Зачет	
<b>4.</b>	<b>Текущая аттестация</b>	Контрольная работа	1
		Домашняя работа	1

**2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ (ИНДИКАТОРЫ) ПО ДИСЦИПЛИНЕ МОДУЛЯ Разработка приложений на языке PYTHON**

Индикатор – это признак / сигнал/ маркер, который показывает, на каком уровне обучающийся должен освоить результаты обучения и их предъявление должно подтвердить факт освоения предметного содержания данной дисциплины, указанного в табл. 1.3 РПМ-РПД.

Таблица 1

<b>Код и наименование компетенции</b>	<b>Планируемые результаты обучения (индикаторы)</b>	<b>Контрольно-оценочные средства для оценивания достижения результата обучения по дисциплине</b>
<b>1</b>	<b>2</b>	<b>3</b>
ОПК-4 -Способен разрабатывать технические объекты, системы и технологические процессы в своей профессиональной деятельности с учетом экономических, экологических, социальных ограничений	Д-1 - Демонстрировать креативное мышление, творческие способности З-3 - Привести примеры сравнения предложенных решений с мировыми аналогами П-1 - Выполнять в рамках поставленного задания разработки технических объектов, систем, в том числе информационных, и технологических процессов в своей профессиональной деятельности с учетом экономических, экологических, социальных ограничений	Домашняя работа Зачет Контрольная работа Лабораторные занятия Лекции

	У-3 - Оценить экологические и социальные риски внедрения предложенных инженерных решений	
ПК-6 -Способен управлять рисками разработки программного обеспечения, а также управлять процессами оценки сложности и трудоемкости в коллективной среде разработки	<p>З-1 - Сделать обзор лучших практик управления разработкой программного обеспечения</p> <p>З-2 - Различать методологии управления проектами разработки программного обеспечения</p> <p>З-3 - Идентифицировать наборы библиотек повторно используемых модулей для разработки программного обеспечения</p> <p>З-4 - Сделать обзор средств создания и ведения репозитория, учета задач, сборки и непрерывной интеграции, базы знаний</p> <p>П-1 - Иметь практический опыт выбора адекватного набора библиотек повторно используемых модулей для разработки программного обеспечения</p> <p>П-2 - Иметь практический опыт выбора средств создания и ведения репозитория, учета задач, сборки и непрерывной интеграции, базы знаний</p> <p>У-1 - Систематизировать и оценивать научно-техническую информацию о методологиях разработки программного обеспечения</p>	<p>Домашняя работа</p> <p>Зачет</p> <p>Контрольная работа</p> <p>Лабораторные занятия</p> <p>Лекции</p>

### **3. ПРОЦЕДУРЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ В РАМКАХ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ МОДУЛЯ В БАЛЬНО-РЕЙТИНГОВОЙ СИСТЕМЕ (ТЕХНОЛОГИЧЕСКАЯ КАРТА БРС)**

#### **3.1. Процедуры текущей и промежуточной аттестации по дисциплине**

**1. Лекции: коэффициент значимости совокупных результатов лекционных занятий – 0.5**

Текущая аттестация на лекциях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
<i>контрольная работа</i>	1,16	100
Весовой коэффициент значимости результатов текущей аттестации по лекциям – <b>0.5</b>		
Промежуточная аттестация по лекциям – <b>зачет</b>		
Весовой коэффициент значимости результатов промежуточной аттестации по лекциям – <b>0.5</b>		
<b>2. Практические/семинарские занятия: коэффициент значимости совокупных результатов практических/семинарских занятий – не предусмотрено</b>		
Текущая аттестация на практических/семинарских занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент значимости результатов текущей аттестации по практическим/семинарским занятиям – <b>не предусмотрено</b>		
Промежуточная аттестация по практическим/семинарским занятиям – <b>нет</b>		
Весовой коэффициент значимости результатов промежуточной аттестации по практическим/семинарским занятиям – <b>не предусмотрено</b>		
<b>3. Лабораторные занятия: коэффициент значимости совокупных результатов лабораторных занятий – 0.5</b>		
Текущая аттестация на лабораторных занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
<i>домашняя работа</i>	1,16	100
Весовой коэффициент значимости результатов текущей аттестации по лабораторным занятиям – <b>1</b>		
Промежуточная аттестация по лабораторным занятиям – <b>нет</b>		
Весовой коэффициент значимости результатов промежуточной аттестации по лабораторным занятиям – <b>не предусмотрено</b>		
<b>4. Онлайн-занятия: коэффициент значимости совокупных результатов онлайн-занятий – не предусмотрено</b>		
Текущая аттестация на онлайн-занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент значимости результатов текущей аттестации по онлайн-занятиям – <b>не предусмотрено</b>		
Промежуточная аттестация по онлайн-занятиям – <b>нет</b>		
Весовой коэффициент значимости результатов промежуточной аттестации по онлайн-занятиям – <b>не предусмотрено</b>		

### 3.2. Процедуры текущей и промежуточной аттестации курсовой работы/проекта

Текущая аттестация выполнения курсовой работы/проекта	Сроки – семестр, учебная неделя	Максимальная оценка в баллах

Весовой коэффициент текущей аттестации выполнения курсовой работы/проекта– **не предусмотрено**

Весовой коэффициент промежуточной аттестации выполнения курсовой работы/проекта– защиты – **не предусмотрено**

#### 4. КРИТЕРИИ И УРОВНИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ МОДУЛЯ

4.1. В рамках БРС применяются утвержденные на кафедре/институте критерии (признаки) оценивания достижений студентов по дисциплине модуля (табл. 4) в рамках контрольно-оценочных мероприятий на соответствие указанным в табл.1 результатам обучения (индикаторам).

Таблица 4

##### Критерии оценивания учебных достижений обучающихся

Результаты обучения	Критерии оценивания учебных достижений, обучающихся на соответствие результатам обучения/индикаторам
Знания	Студент демонстрирует знания и понимание в области изучения на уровне указанных индикаторов и необходимые для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.
Умения	Студент может применять свои знания и понимание в контекстах, представленных в оценочных заданиях, демонстрирует освоение умений на уровне указанных индикаторов и необходимых для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.
Опыт /владение	Студент демонстрирует опыт в области изучения на уровне указанных индикаторов.
Другие результаты	Студент демонстрирует ответственность в освоении результатов обучения на уровне запланированных индикаторов. Студент способен выносить суждения, делать оценки и формулировать выводы в области изучения. Студент может сообщать преподавателю и коллегам своего уровня собственное понимание и умения в области изучения.

4.2 Для оценивания уровня выполнения критериев (уровня достижений обучающихся при проведении контрольно-оценочных мероприятий по дисциплине модуля) используется универсальная шкала (табл. 5).

Таблица 5

##### Шкала оценивания достижения результатов обучения (индикаторов) по уровням

Характеристика уровней достижения результатов обучения (индикаторов)			
№ п/п	Содержание уровня выполнения критерия оценивания результатов обучения (выполненное оценочное задание)	Шкала оценивания	
		Традиционная характеристика уровня	Качественная характеристика уровня

1.	Результаты обучения (индикаторы) достигнуты в полном объеме, замечаний нет	Отлично (80-100 баллов)	Зачтено	Высокий (В)
2.	Результаты обучения (индикаторы) в целом достигнуты, имеются замечания, которые не требуют обязательного устранения	Хорошо (60-79 баллов)		Средний (С)
3.	Результаты обучения (индикаторы) достигнуты не в полной мере, есть замечания	Удовлетворительно (40-59 баллов)		Пороговый (П)
4.	Освоение результатов обучения не соответствует индикаторам, имеются существенные ошибки и замечания, требуется доработка	Неудовлетворительно (менее 40 баллов)	Не зачтено	Недостаточный (Н)
5.	Результат обучения не достигнут, задание не выполнено	Недостаточно свидетельств для оценивания		Нет результата

## 5. СОДЕРЖАНИЕ КОНТРОЛЬНО-ОЦЕНОЧНЫХ МЕРОПРИЯТИЙ ПО ДИСЦИПЛИНЕ МОДУЛЯ

### 5.1. Описание аудиторных контрольно-оценочных мероприятий по дисциплине модуля

#### 5.1.1. Лекции

Самостоятельное изучение теоретического материала по темам/разделам лекций в соответствии с содержанием дисциплины (п. 1.2. РПД)

#### 5.1.2. Лабораторные занятия

Примерный перечень тем

1. Пакеты
  2. Классы
  3. Расширенные возможности классов
  4. Менеджеры пакетов
  5. Работа с консолью и файлами
  6. Библиотека numpy
  7. Библиотека matplotlib
  8. Библиотека pandas
- LMS-платформа – не предусмотрена

### 5.2. Описание внеаудиторных контрольно-оценочных мероприятий и средств текущего контроля по дисциплине модуля

Разноуровневое (дифференцированное) обучение.

#### Базовый

##### 5.2.1. Контрольная работа

## Примерный перечень тем

1. Примитивные типы
2. Строки
3. Список
4. Кортеж
5. Словарь
6. Множество
7. Включения
8. Функция
9. Модуль
10. Класс

### Примерные задания

Операции над примитивными типами. Операция ``type``. Преобразования типов. Атомарные и не атомарные типы.

Строки. Основные методы. Индексация. Извлечение подстрок. Основные функции и методы.

Список. Способы конструирования. Основные функции и методы. Индексация. Списки списков. Модификация списков. Различия между кортежем и списком. Способы копирования.

Кортеж. Способы конструирования. Основные функции. Индексация. Тип данных ``slice``. Распаковка кортежей. Различия между кортежем и списком. Способы копирования.

Словарь. Способы конструирования. Основные функции и методы. Способы поиска и модификации. Применение оператора `in`. Проверка наличия ключа. Итерирование. Способы копирования.

Множество. Способы конструирования. Основные функции и методы. Способы поиска и модификации. Применение оператора `in`. Проверка наличия ключа. Итерирование. Способы копирования. Тип данных ``FrozenSet``.

Включения. Синтаксис. Применения условия. Включения, производящие списки. Включения, производящие генераторы.

Функция. Синтаксис. Определение. Вызов. Значения, возвращаемые функцией. Ключевое слово `return`. Тип данных ``NoneType``.

Модуль. Понятие. Импортирование модуля. Ключевое слово ``import``. Алгоритм работы операции импортирования. Алгоритм поиска модуля. Список ``sys.path``, переменная окружения ``PYTHONPATH``, ``*.py`` файлы. Алгоритм компиляции модуля.

Класс. Определение. Объект класса. Синтаксис класса. Метод класса. Синтаксис метода класса. Ключевое слово ``self``. Конструктор класса. Правила хорошего тона при проектировании конструктора класса. Синтаксис. Поля экземпляра класса. Доступ



### 5.2.2. Домашняя работа

Примерный перечень тем

1. Пакеты
2. Классы
3. Нетривиальные аспекты применения классов
4. Менеджеры пакетов
5. Консоль. Файлы
6. Библиотека `numpy`
7. Библиотека `matplotlib`
8. Библиотека `pandas`

Примерные задания

1. Разработайте функции для работы с длинами в английской системе мер (только футы и дюймы): сумма, разность, умножение на число, деление на равные части. Поместите функции в модуль `imperial\_distance/math/core.py`.
  2. Разработайте функции `test` для тестирования функциональности модуля `.../core.py`. Тесты поместите в модуль `imperial\_distance/math/test.py`.
  3. Разработайте функции по записи значения английской системы мер в строку. Поместите функции в `imperial\_distance/io/to\_str.py`.
  4. Разработайте функции по конвертированию длин из английской системы мер в метрическую и обратно. Поместите функции в `imperial\_distance/io/to\_metric.py`
  5. Из всех модулей соберите пакет.
  6. Продемонстрируйте работу собранного пакета из скриптов при помощи `PYTHONPATH` и импорта из текущей директории.
- 
1. Создайте класс, который называется `Thing`, не имеющий содержимого, и выведите его на экран. Затем создайте объект `example` этого класса и также выведите его. Совпадают ли выведенные значения?
  2. Создайте класс `Element`, имеющий атрибуты объекта `name`, `symbol` и `number`. Создайте объект этого класса со значениями `'Hydrogen'`, `'H'` и `1`.
  3. Для класса `Element` определите метод с именем `dump()`, который выводит на экран значения атрибутов объекта (`name`, `symbol` и `number`). Создайте объект `hydrogen` из этого нового определения и используйте метод `dump()`, чтобы вывести на экран его атрибуты.
  4. На основе пакета математики в английской системе мер, создайте класс `distance` для работы с расстояниями в английской системе мер (футы и дюймы). Класс должен обладать инкапсуляцией, иметь операции сложения, вычитания расстояний и вывода строкового представления расстояния.
  5. Создайте класс, представляющий собой размеры комнаты в английской системе мер. Комната имеет три измерения: длина, ширина, высота. Класс должен рассчитывать площадь пола, стен, потолка, а также объем комнаты.
  6. Создайте иерархию классов:
    - Транспортное средство (ТС): масса, мощность, количество колес.
    - Легковой автомобиль: базовый класс - ТС, количество мест для сидения.
    - Автобус: базовый класс - ТС, количество мест для сидения, количество стоячих мест, маршрут следования.

- Грузовой автомобиль: базовый класс - ТС, масса перевозимого груза.

Каждый класс должен иметь метод, выводящий отформатированную информацию о этом транспортном средстве. Создайте list различных транспортных средств и в цикле for выведите из все на консоль.

1. Для иерархии классов транспортных средства из предыдущего задания сделайте класс транспортного средства абстрактным.

2. Перепишите иерархию классов транспортных средств выделив протокол, так чтобы каждый класс конкретного транспортного средства реализовывал его.

3. Для класса дробей реализуйте перегруженные операции '+' и '\*'.

4. Реализуйте операцию деления длины в английской системе мер на части.

Выбрасывайте исключение, при попытке деления на ноль.

5. Реализуйте операцию чтения значения английской системы мер из строки.

Выбрасывайте исключение, если входная строка не соответствует формату.

6. Реализуйте консольный калькулятор, позволяющий выполнять операции с длинами. Обрабатывайте исключение несоответствия входной строки, но не исключение деления на 0.

7. Используя менеджер контекста добейтесь вывода на консоль строки 'Завершение работы' при завершении работы программы даже при наличии необработанного исключения.

1. Развернуть виртуальные окружения ('pipenv' или 'virtualenv') и 'conda'.

2. В каждое виртуальное окружение установить пакеты 'numpy', 'pandas', 'matplotlib'.

3. Продемонстрировать, что вы работаете именно в виртуальном окружении и что установленные пакеты работают.

1. Разработать консольную утилиту пиццерии. Утилита должна обеспечивать ввод данных о тесте (толстое или тонкое), диаметре (один из трех вариантов) и перечне ингредиентов (произвольный список). Утилита должна выводить на консоль информацию о диаметре и типе теста. Ингредиенты должны быть отфильтрованы по перечню существующих ингредиентов. Реализовать задачу методами консольного ввода, библиотеками 'argparse' и 'click'.

2. Изменить программу таким образом, чтобы она считывала ввод в одном формате из файла, а выводила - в другом в файл. Форматы файлов: 'xml', 'json', 'yaml', 'csv', конфигурационный файл. Для работы с форматами файлов использовать библиотеки.

1. Создайте 2D массив numpy 15x15, целочисленного типа в котором единицами напишите псевдографикой первые две буквы своего имени. При выполнении используйте возможности индексации numpy (прихотливую, маскированную, комбинированную) и поэлементного сложения.

2. Программным способом определите:

- среднее количество единиц в столбцах, содержащих хотя бы одну единицу
- среднее количество единиц в строках, содержащих хотя бы одну единицу
- высоту и ширину каждой буквы
- есть ли в вашем массиве повторяющиеся столбцы или строки

- максимальное расстояние по Евклиду между единицами

1. Постройте псевдографическое изображение из задания к предыдущей лекции.
2. Постройте гистограммы, отражающие количество закрашенных пикселей псевдографического изображения по горизонтали и вертикали. Гистограммы расположите с наружной стороны оси рядом с осями.
3. Постройте графики случайных процессов и их гистограммы (на одном рисунке) для генераторов `np.random.rand``, `np.random.randn`` и `np.random.lognormal``.
4. Во всех заданиях подпишите оси и названия графиков.
5. У гистограмм случайных процессов подпишите максимум, если он один.
6. При рисовании используйте как минимум два разных стиля.

1. Скачайте набор данных о чемпионате евро 2012:  
<[https://raw.githubusercontent.com/guipsamora/pandas\\_exercises/master/02\\_Filtering/\\_%26\\_Sorting/Euro12/Euro\\_2012\\_stats\\_TEAM.csv](https://raw.githubusercontent.com/guipsamora/pandas_exercises/master/02_Filtering/_%26_Sorting/Euro12/Euro_2012_stats_TEAM.csv)>
  - Программно отобразите имена всех стран участниц.
  - Команда какой страны забила больше всего мячей? Сколько мячей было забито во время чемпионата?
  - Выведите первые 10 стран из всех, отсортированных по количеству красных и желтых карточек.

2. Скачайте набор данных об употреблении алкоголя в странах по адресу:  
<<https://raw.githubusercontent.com/justmarkham/DAT8/master/data/drinks.csv>>.

Получите ответы на следующие вопросы:

- Какая страна потребляет меньше всего алкоголя?
- Какая страна потребляет спирты (spirits) чаще всего?
- Какой континент потребляет вино реже всего?

3. Скачайте набор данных о компании Apple адресу:  
<[https://raw.githubusercontent.com/guipsamora/pandas\\_exercises/master/09\\_Time\\_Series/Apple\\_Stock/appl\\_1980\\_2014.csv](https://raw.githubusercontent.com/guipsamora/pandas_exercises/master/09_Time_Series/Apple_Stock/appl_1980_2014.csv)>.

Получите ответы на следующие вопросы:

- В какой день цена акций поднялась больше всего? Упала?
- В какой месяц цена акций выросла больше всего?
- В какой год цена акций выросла больше всего?

LMS-платформа – не предусмотрена

### **5.3. Описание контрольно-оценочных мероприятий промежуточного контроля по дисциплине модуля**

#### **5.3.1. Зачет**

Список примерных вопросов

1. Статические поля класса. Методы класса. Декоратор `@classmethod`` Статические методы класса. Декоратор `@staticmethod``.
2. Сравнение классов и словарей. Конструирование анонимных объектов. Применение функции ``type`` для построения объектов.

3. Способы обеспечения инкапсуляции в `Python`. Одиночное подчеркивание и двойное подчеркивание. Поведение интерпретатора `Python` для таких полей. Определение свойств экземпляра класса используя функцию `property` и декоратор `@property`. Свойства только для чтения. Вычисляемые свойства.
4. Наследование. Концепция. Синтаксис. Порядок вызова конструктора базового класса. Явный вызов конструктора базового класса. Функция `super`. Переопределение метода в подклассе. Место класса `Object` в иерархии наследования. Порядок разрешения методов. Атрибут `\_\_mro\_\_`.
5. Полиморфизм. Классический полиморфизм. Утиный полиморфизм. Реализация в `Python`. Агрегирование. Концепция. Пример.
6. Абстрактный класс. Модуль `abc`. Перечисление. Модуль `enum`. Протокол. Тип `typing.Protocol`.
7. Магические методы. Методы жизненного цикла `\_\_new\_\_` и `\_\_init\_\_`. Применение. Математические магические методы. Математические методы контейнеров. Определение класса-контейнера и класса-итератора.
8. Магические методы для работы с атрибутами `\_\_getattr\_\_`, `\_\_setattr\_\_`, `\_\_delattr\_\_`, `\_\_getattribute\_\_`. Разница `\_\_getattr\_\_` и `\_\_getattribute\_\_`. Метод `\_\_call\_\_`. Вызов объекта класса как функции.
9. Исключения. Обработчик исключений по умолчанию. Синтаксис обработки исключений. Ключевые слова `try`, `except`, `finally`, `else`. Возбуждение исключений. Ключевое слово `raise`. Повторное возбуждение последнего исключения. Алгоритм обработки исключений. Конструкция `raise from`. Конструкция `assert`. Иерархия классов исключений. Привести примеры трех классов исключения и их использования. Пользовательские классы исключений.
10. Менеджер контекста. Синтаксис. Ключевое слово `with`. Протокол управления контекстом. Магические методы для реализации класса менеджера контекста. Множественные менеджеры контекста.
11. Пакет `wheel`. Формат пакета. Применение `setuptools` для создания и развертывания пакетов. Менеджер пакетов `pip`. Команды. Пакетный менеджер `conda`. Команды.
12. Виртуальное окружение. `virtualenv` и `pipenv`. Архитектура, принцип работы, основные команды.
13. Функции `input` и `print` для создания консольных утилит. Модуль `pprint`. Библиотеки богатого консольного интерфейса `blessings`. Библиотека обработки параметров командной строки `argparse`. Функция `add\_argument`. Пакет `click`. Основные аннотации.
14. Работа с файлами. Функция `open`. Параметры и режимы открытия файлов. Работа с файлом как с итератором. Гарантирование закрытия файла. Модули `csv`, `xml`, `json`, `yaml`, `configparser`, `xlrd`.
15. Сериализация и десериализация объектов. Модуль `pickle`.37. Библиотека `numpy`. Достоинства недостатки. Основные типы данных: `ndarray`, `dtype`, `nditer`, `MaskedArray`. Способы создания массивов. Стандартные типы данных. Атрибуты массивов `ndim`, `shape`, `size`, `itemsize`, `nbytes`.
16. Индексация массива `numpy`. Индексация многомерного массива. Срезы массивов. Доступ к строкам и столбцам. Запись в срезы. Объект `Ellipsis`. Флаги массива

``C_CONTIGUOUS`, `F_CONTIGUOUS`, `OWNDATA`, `WRITEABLE``. Создание копии массива. Метод ``reshape``. Пустые измерения. ``numpy.newaxis``.

17. Объединение массивов ``numpy``. ``concatenate`, `hstack`, `vstack`, `dstack``. Достоинства и ограничения. Разбиение массивов. Методы ``split`, `hsplit`, `vsplit`, `dsplit``.

18. Векторизованные функции ``numpy``. Универсальные функции. Функции для агрегирования ``sum`, `prod`, `min`, `max``. Транслирование размеров. Правила транслирования.

19. Маскирование в ``numpy``. Операторы сравнения. Агрегирующие битовые функции ``any`, `all``. Булевы операторы. Операция наложения маски.

LMS-платформа – не предусмотрена

#### **5.4 Содержание контрольно-оценочных мероприятий по направлениям воспитательной деятельности**

Направления воспитательной деятельности сопрягаются со всеми результатами обучения компетенций по образовательной программе, их освоение обеспечивается содержанием всех дисциплин модулей.