

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ

Декан дорожно-транспортного факультета



Тюнин В.Л. /

2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Разработка расширений для ГИС на языке программирования
Python»

Направление подготовки 05.04.03 Картография и геоинформатика

Программа Геоинформационное моделирование

Квалификация выпускника Магистр

Нормативный период обучения 2 года

Форма обучения очная

Год начала подготовки 2026

Автор программы
Заведующий кафедрой
Базовая кафедра
кибернетики в системах
организационного
управления

В.Е. Белоусов

В.Е. Белоусов

Руководитель ОПОП

Н.И. Самбулов

Воронеж 2025

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины научить обучаемых разработке инструментов для сбора и преобразования необработанных данных, геокодирования и выполнения операций над данными, создания пространственных объектов и т.д. с использованием языка программирования Python.

1.2. Задачи освоения дисциплины

- автоматизация операций, таких как визуализация промежуточных результатов, публикация карт, отправка уведомлений заинтересованным участникам проекта по расписанию.

- проектирование или самостоятельная разработка специализированных решений, применяя в том числе алгоритмы геостатистики.

- разработка алгоритмов и моделей машинного обучения для автоматизации задач, таких как извлечение данных из спутниковых снимков и других источников, анализа, интерполяция, расчёт прогнозных показателей, создание и обучение пространственных моделей, пространственная кластеризация и других.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Разработка расширений для ГИС на языке программирования Python» относится к дисциплинам части, формируемой участниками образовательных отношений (дисциплина по выбору) блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Разработка расширений для ГИС на языке программирования Python» направлен на формирование следующих компетенций:

ПК-1 - Способен разрабатывать технологии создания тематических информационных продуктов и оказания услуг на основе использования геопространственных данных

ПК-3 - Способен выполнять технологическое обеспечение и координацию выполнения комплекса операций по развитию и модернизации существующих разноуровневых геоинформационных систем

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-1	знать принципы и подходы к обработке пространственных данных, а также возможности языка программирования Python и дополнительных библиотек для решения автоматизации обработки географических данных;
	уметь разрабатывать алгоритм автоматизации обработки данных; выбирать оптимальные средства для реализации алгоритма программными средствами

	владеть программными средствами для написания и тестирования скриптов; приемами написания программ и модулей
ПК-3	знать способы разработки алгоритмов, для расчета локальных и глобальных коэффициентов корреляции, и с учетом зон для двух растровых изображений; уметь реализовать алгоритмы, для расчета локальных и глобальных коэффициентов корреляции на языке Python, тестирование разработанных программ владеть способностью тестирования разработанных программ на различных наборах растровых изображений

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Разработка расширений для ГИС на языке программирования Python» составляет 3 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры
		3
Аудиторные занятия (всего)	36	36
В том числе:		
Лекции	18	18
Лабораторные работы (ЛР)	18	18
в том числе в форме практической подготовки	8	8
Самостоятельная работа	72	72
Виды промежуточной аттестации - зачет	+	+
Общая трудоемкость:		
академические часы	108	108
зач.ед.	3	3

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Python для решения профессиональных задач ГИС-специалиста	История и принципы Python. Формирование кода, присваивание значения переменным, вывод их, анализ ошибок. Основные типы данных и их преобразованиями друг в друга. Тип данных — числа. Какие виды чисел используют в языке Python. Арифметические операции и их порядок. Ошибки, которые могут возникать при вычислениях. Как хранятся вещественные числа, стандартные приемы в Python для работы с ними, а также основы работы с библиотекой math.	4	4	12	20
		<i>практическая подготовка обучающихся</i>	-	2	-	2

2	Создание программ с условиями при решении задач анализа ГИС	Условные операторы. Булевый тип данных. Сравнение чисел. Логические операции, сложные логические выражения. Формы условных инструкций IF, ELIF и ELSE и применение их на практике. Функции, создание и использование. Строки. Понятие строки и способы ее представления. Отличие строки от символа. Методы и способы форматирования строк. Устройство строк в памяти и кодировки.	4	4	12	20
		<i>практическая подготовка обучающихся</i>	-	2	-	2
3	Пространственный анализ, обработка растровых и векторных данных с использованием средств Python	Циклы и области видимости. Цикл for. Функции range(), enumerate(), итерация и тело цикла. Применение помощников циклов: reversed(), break, continue. Синтаксис цикла while и особенности его применения. Понятие областей видимости и их применение в функциях и циклах. Массивы. Создание и использование списков, кортежи и их применение. Создание матриц и обращение к элементам матриц.	4	4	12	20
		<i>практическая подготовка обучающихся</i>	-	2	-	2
4	Алгоритмизация на Python для решения прикладных задач обработки Big Data в ГИС	Алгоритмы с массивами. Задача сортировки, алгоритмы для ее решения данной задачи и асимптотика каждого алгоритма. Стандартные методы сортировки в Python. Алгоритмы поиска элемента в массиве на языке Python. Скорость и применимость. Динамическое программирование на примере классических задач. Алгоритмы с числами. Понятие делимости. Простые и составные числа. Определения наибольшего общего делителя и наименьшего общего кратного и реализуете алгоритмы их поиска. Алгоритмы поиска простых чисел, наивный подход и решето Эратосфена. Алгоритмы со строками. Задача проверки строки на палиндром. Преобразование строки в словарь и задачи с использованием этого приема. Несколько алгоритмов поиска подстроки. Префикс-функция.	2	2	12	16
		<i>практическая подготовка обучающихся</i>	-	2	-	2
5	Машинное обучение и ИИ в задачах анализа и проектирования ГИС на Python	Специализированные библиотеки для анализа и обработки геоданных (GeoPandas, Shapely, и PySAL). Geopandas – это библиотека, предназначенная для обработки и анализа геоданных с использованием привычных для Pandas структур данных. Geopandas предоставляет функционал для работы с геометрией объектов, таких как точки, линии и полигоны, а также методы для пространственного анализа. Shapely – это библиотека для работы с планарными геометрическими объектами. Она предоставляет функционал для создания, анализа и манипуляции с геометрией объектов. Shapely является основной библиотекой для работы с геометрией в Geopandas. PySAL (Python Spatial Analysis Library) — это открытая библиотека для пространственной науки на Python с акцентом на геопространственный анализ, геокомputation и пространственную эконометрику.	2	2	12	16
6	Обработка изображений в ГИС с использованием прикладных Python программ	Отрисовка карты субъекта России на различных уровнях детальности (субъекты, города, городские районы) и подтягивания к ней данных. Определение форматов файлов для чтения. OpenStreetMap - картографический сервис, наполняемый по принципу Википедии — желающие могут редактировать данные, добавлять недостающую информацию и так далее. Мэтчинг данных с картой. Выстраивание связи между почтовыми кодами (к которым привязаны наши данные) и нужными нам уровнями детализации (город / район и тд). Установка и импорт библиотек. Отрисовка	2	2	12	16

		административных границ разных уровней. Формирование нескольких слоев. Обработка данных. Формирование тепловой карты.				
		Итого	18	18	72	108

Практическая подготовка при освоении дисциплины (модуля) проводится путем непосредственного выполнения обучающимися отдельных элементов работ, связанных с будущей профессиональной деятельностью, способствующих формированию, закреплению и развитию практических навыков и компетенций по профилю соответствующей образовательной программы на практических занятиях и (или) лабораторных работах:

№ п/п	Перечень выполняемых обучающимися отдельных элементов работ, связанных с будущей профессиональной деятельностью	Формируемые профессиональные компетенции
1	Как хранятся вещественные числа, стандартные приемы в Python для работы с ними, а также основы работы с библиотекой math.	ПК-1, ПК-3
2	Функции, создание и использование. Строки. Понятие строки и способы ее представления. Отличие строки от символа. Методы и способы форматирования строк. Устройство строк в памяти и кодировки.	ПК-1, ПК-3
3	Массивы. Создание и использование списков, кортежи и их применение. Создание матриц и обращение к элементам матриц.	ПК-1, ПК-3
4	Алгоритмы поиска простых чисел, наивный подход и решето Эратосфена.	ПК-1, ПК-3
5	Специализированные библиотеки для анализа и обработки геоданных (GeoPandas, Shapely, и PySAL).	ПК-1, ПК-3
6	Отрисовка карты субъекта России на различных уровнях детальности (субъекты, города, городские районы) и подтягивания к ней данных.	ПК-1, ПК-3

5.2 Перечень лабораторных работ

1. Знакомство со средой прикладного программирования Google Colab.
2. Обработка датасетов при решении задач ГИС в Google Colab.
3. Обработка растровых и векторных данных с использованием библиотек Python.
4. Методы визуализации больших данных на Python.
5. Разметка и анализ геоданных с использованием QGIS + Python.
6. Использование класса Geocode с Pandas DataFrame.

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

В соответствии с учебным планом освоение дисциплины не предусматривает выполнение курсового проекта (работы) или контрольной работы.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

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

7.1.1 Этап текущего контроля

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

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ПК-1	знать принципы и подходы к обработке пространственных данных, а также возможности языка программирования Python и дополнительных библиотек для решения автоматизации обработки географических данных;	Полное или частичное посещение лекционных, практических и лабораторных занятий. Выполненные СРС, УО, ПО	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь разрабатывать алгоритм автоматизации обработки данных; выбирать оптимальные средства для реализации алгоритма программными средствами	Полное или частичное посещение лекционных, практических и лабораторных занятий. Выполненные СРС, УО, ПО	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть программными средствами для написания и тестирования скриптов; приемами написания программ и модулей	Полное или частичное посещение лекционных, практических и лабораторных занятий. Выполненные СРС, УО, ПО	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

ПК-3	знать способы разработки алгоритмов, для расчета локальных и глобальных коэффициентов корреляции, и с учетом зон для двух растровых изображений;	Полное или частичное посещение лекционных, практических и лабораторных занятий. Выполненные СРС, УО, ПО	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь реализовать алгоритмы, для расчета локальных и глобальных коэффициентов корреляции на языке Python, тестирование разработанных программ	Полное или частичное посещение лекционных, практических и лабораторных занятий. Выполненные СРС, УО, ПО	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть способностью тестирования разработанных программ на различных наборах растровых изображений	Полное или частичное посещение лекционных, практических и лабораторных занятий. Выполненные СРС, УО, ПО	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 3 семестре для очной формы обучения по двухбалльной системе:

«зачтено»

«не зачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Не зачтено
ПК-1	знать принципы и подходы к обработке пространственных данных, а также возможности языка программирования Python и дополнительных библиотек для решения автоматизации обработки географических данных;	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь разрабатывать алгоритм автоматизации обработки данных; выбирать оптимальные средства для	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

	реализации алгоритма программными средствами			
	владеть программными средствами для написания и тестирования скриптов; приемами написания программ и модулей	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ПК-3	знать способы разработки алгоритмов, для расчета локальных и глобальных коэффициентов корреляции, и с учетом зон для двух растровых изображений;	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь реализовать алгоритмы, для расчета локальных и глобальных коэффициентов корреляции на языке Python, тестирование разработанных программ	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть способностью тестирования разработанных программ на различных наборах растровых изображений	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

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

7.2.1 Примерный перечень заданий для подготовки к тестированию

1. Для каких поколений Python работает The Zen of Python?

- 4
- 5
- 2
- ни для каких, у python нет дзена
- 3

2. Чем полезны комментарии в коде?

- они более понятно описывают происходящее в коде
- добавляют красоты в код
- хранят в себе скрытые переменные
- являются пространством для переписок разработчиков

- выделяют важные моменты

3. Почему важно стараться придерживаться стандарта PEP8 при написании кода на Python?

- это модно
- увеличивает зарплату
- шифрует код от чужих глаз
- вы сможете понять свой код спустя время
- другие люди смогут быстро разобраться в вашем коде, что очень важно на совместных проектах

4. Какие функции помогают работать пользователю с вводом и выводом данных?

- import
- print
- enter
- input
- view

5. Вам нужно в программе получить число итераций от пользователя после сообщения. Введите число итераций. Какие из вариантов ниже выполняют данную задачу?

- print('Введите число итераций: ')
- iter = int(input())
- print('Введите число итераций: ')
- iter = input()
- iter = int(input('Введите число итераций: '))
- iter = int(input())

6. Какая из переменных хранит в себе y-координату параболы?

- y = 1
- y = x*x
- y = 15 - 5*x + x*x
- y = 10 - 1
- y = x

7. В каких случаях результат выполнения будет <class 'str'>?

- print(type('my first word'))
- print(type(1984))
- print(type('1984'))
- print(type(None))
- print(type("None"))

8. В каких случаях теряется дробная часть, а не появляется ошибка?

- int(6.75)
- int("19.87613")
- int(13/2)
- int(17.5*18.9)
- int("13/2")

9. Для задачи нужны данные о росте пользователя. Рост может быть дробным числом. Какие из предложенных вариантов подходят для считывания этой информации и дальнейшей работы с ней?

- height = int(input('Введите рост: '))

- height = input('Введите рост: ')
- height = int(input())
- height = input('Введите рост: ')
- height = float(height)
- height = float(input())

10. Что получится в результате выполнения следующего кода?

- a = bool("True")
- b = bool()
- print(f'{a}-{b}')
- True-True
- True-False
- False-True
- False-False

7.2.2 Примерный перечень заданий для решения стандартных задач

1. Какими лексемами зарезервированы значения для булева типа?

- True и False
- 1 и 0
- true и false
- t и f

2. Как строки преобразуются в булевы значения?

- ложна любая непустая строка
- истинна любая строка
- ложна любая строка
- истинна любая непустая строка

3. Что будет выведено, если выполнить код ниже?

```
f = 'False'
t = 'True'
if bool(f) != bool(t):
    print(bool(f))
    print("test")
```

- test
- False
- test
- True
- test
- None

4. Какие операторы сравнения можно использовать в логическом выражении условной инструкции IF?

- все
- только равенства и неравенства
- только больше и меньше
- только больше или равно, или меньше или равно

5. В каком случае выполняется блок кода в условной инструкции IF?

- если условие ложно
- если условие истинно

- если условие истинно и ложно
- если условие истинно или ложно

6. Сколько может быть инструкций IF в коде?

- только одна на программу
- только одна на файл
- сколько угодно
- только одна на функцию

7. Что будет в результате выполнения кода ниже?

```
value = 1
if value > 0:
    print('>0')
elif:
    print('<=0')
```

- >0
- <=0
- Error
- None

8. Как объявить функцию в Python?

- def
- func
- define
- function

9. Как выглядит оператор вызова функции?

- []
- ()
- {}
- call
- call()
- call[]
- Call{}

10. Каков будет результат работы следующей функции, если в качестве аргумента ей передать число 5?

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)
```

- 1
- 2
- 6
- 24
- 120

7.2.3 Примерный перечень заданий для решения прикладных задач

1. Каков будет результат работы следующей функции, если в качестве аргументов ей передать числа 4 и 6?

```
def lcm(x, y):
    if x > y:
```

```

greater = x
else:
greater = y
while(True):
if((greater % x == 0) and (greater % y == 0)):
lcm = greater
break
greater += 1
return lcm
- 0
- 1
- 12
- 4

```

2. Что является результатом выражения "Python is fun".split() в Python?

- ["Python", "is", "fun"].
- "Python is fun"
- ["Pythonisfun"].
- ["Python", "isfun"].

3. Какой список сформируется, если выполнить команду list(range(1, 5))?

- [1, 2, 3]
- [0, 1, 2, 3, 4, 5]
- [1, 2, 3, 4]
- [0, 1, 2, 5]
- [5, 4, 3, 2, 1]

4. Сколько раз программа ниже напечатает фразу «hello world»?

```

t = 3
while t <= 8:
print("hello world")
t += 1
- 6
- Бесконечность
- 3
- 2

```

5. Сколько раз программа ниже напечатает фразу «hello world»?

```

t = 3
while t <= 8:
print("hello world")
t -= 1
- 6
- Бесконечность
- 3
- 2

```

6. Какое значение примет переменная "a" после выполнения программы?

```

a = 3
while a <= 48:
a *= 2
- 48
- 24
- 96
- 192

```

7. Каков результата выполнения кода?

```
number = 1234567890
count = 0
while number > 0:
    last_digit = number % 10
    if last_digit < 3:
        count = count + 1
    number = number // 10
print(count)
```

- 4
- 5
- 3
- 0

8. Каким будет результат выполнения следующего кода?

```
first_list = [1, 2, 3]
second_list = first_list
first_list.pop(0)
second_list.append(4)
print(first_list)
```

- [0, 2, 3, 4]
- [1, 2, 3, 4]
- [0, 1, 2, 3, 4]
- [2, 3, 4]

9. Что будет выведено на экран в результате выполнения следующего кода?

```
lst1 = [0, 1, 2]
lst2 = lst1.append([3])
print(len(lst2))
```

- Error
- [0, 1, 2, 3]
- [0, 1, 2, [3]]
- None

10. Укажите результат работы следующей программы (см. код ниже).

```
a = frozenset("set")
b = {}
for elem in a:
    b[elem] = 1
print(b["s"])
```

- 1
- 0
- set()
- frozenset()
- True
- False

7.2.4 Примерный перечень вопросов для подготовки к зачету

1. В чём разница между списками и кортежами? Когда стоит применять именно кортежи?
2. В чём разница между многопроцессорностью и многопоточностью?
3. В чем разница между модулем, пакетом и библиотекой?
4. В чём заключается проблема с многопоточностью в python?

5. Что такое декораторы? Можете ли вы описать ситуацию, в которой стоит использовать декораторы?
6. Как правильно записать данные в файл? Что может пойти не так в ином случае?
7. Аргументы функции передаются по ссылке или по значению?
8. Как изменить способ вывода объектов?
9. Напишите функцию, которая вычисляет факториал целого числа n.
10. В чем разница между операторами `is` и `==`?
11. Когда вам не следует использовать оператор `assert`?
12. Что такое генератор в Python?
13. В чем разница между методом класса и статическим методом?
14. Приведите пример того, как вы используете функции `zip` и `enumerate`.
15. Как бы вы использовали `*args` и `**kwargs`?
16. Приведите пример функционального программирования с использованием функции `map`.
17. В чем разница между операторами `continue` и `break`?
18. Как предотвратить вызов функции ненужное количество раз?
19. Дайте несколько рекомендаций по PEP8.
20. Как прочитать файл объемом 8 ГБ на Python с помощью компьютера с 2 ГБ ОЗУ?
21. Что такое глобальные, защищенные и приватные атрибуты в Python?
22. Как используется `self` в Python?
23. Что такое модульные тесты в Python?
24. В чем разница между массивами и списками в Python?
25. Что такое comprehensions Dict и List?
26. Что такое срезы в Python?
27. Как соединить список строк в одну. Как разбить строку на список строк.
28. Как сделать список уникальным (без повторяющихся элементов).
29. Какая разница между одинарным и двойным подчеркиванием.
30. Что делают StringIO и BytesIO.

7.2.5 Примерный перечень заданий для подготовки к экзамену

Не предусмотрено учебным планом

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Зачет проводится по тест-билетам, каждый из которых содержит 10 вопросов и задачу. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом, задача оценивается в 10 баллов (5 баллов верное решение и 5 баллов за верный ответ). Максимальное количество набранных баллов – 20.

1. Оценка «Незачтено» ставится в случае, если студент набрал менее 6 баллов.

2. Оценка «Зачтено» ставится, если студент набрал от 16 до 20 баллов.)

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Python для решения профессиональных задач ГИС-специалиста	ПК-1, ПК-3	Тест, защита лабораторных работ, зачет
2	Создание программ с условиями при решении задач анализа ГИС	ПК-1, ПК-3	Тест, защита лабораторных работ, зачет
3	Пространственный анализ, обработка растровых и векторных данных с использованием средств Python	ПК-1, ПК-3	Тест, защита лабораторных работ, зачет
4	Алгоритмизация на Python для решения прикладных задач обработки Big Data в ГИС	ПК-1, ПК-3	Тест, защита лабораторных работ, зачет
5	Машинное обучение и ИИ в задачах анализа и проектирования ГИС на Python	ПК-1, ПК-3	Тест, защита лабораторных работ, зачет
6	Обработка изображений в ГИС с использованием прикладных Python программ	ПК-1, ПК-3	Тест, защита лабораторных работ, зачет

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

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

1. Гегечкори, Е. Т. Программирование на языке Python : учебное пособие / Е. Т. Гегечкори. — Омск : ОмГТУ, 2023. — 172 с. — ISBN 978-5-8149-3617-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/421673>

2. Кочетыгов, А. А. Основы программирования на языке Python : учебное пособие / А. А. Кочетыгов. — Тула : ТулГУ, 2024. — 272 с. — ISBN 978-5-7679-5380-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/427316>

3. ДеМерс М. Географические информационные системы. Основы. Пер. с англ. М.; Дата+, 1999. 471 с.

4. Демьянов Г.В. Местные системы координат, существующие проблемы и возможные пути их решения / Г.В. Демьянов, Н.Н. Майоров, Г.Г. Побединский // Геопрофи. 2009. № 2. С. 52–57.

5. Добавление местной координатной системы в ГИС //Gis-Lab: географические информационные системы и дистанционное зондирование – [электронный ресурс]. Режим доступа: <http://gis-lab.info/qa/local-cs.html>

6. Дубинин М. Общее описание ASTER GDEM // Gis-Lab: географические информационные системы и дистанционное зондирование. Режим доступа: <http://gis-lab.info/qa/aster-gdem.html>

7. Дубинин М. Описание и получение данных SRTM // Gis-Lab: географические информационные системы и дистанционное зондирование – [электронный ресурс]. Режим доступа: <http://gis-lab.info/qa/srtm.html>

8. Дубинин М. Разница в определении координат в WGS-1984 и СК-1942 // Gis-Lab: географические информационные системы и дистанционное зондирование – [электронный ресурс]. Режим доступа: <http://gis-lab.info/qa/wgs-pul-compare.html>

9. <https://inspire-geoportal.ec.europa.eu/srv/rus/catalog.search#/home> - Интеграция ИПТ стран Евросоюза.

10. <https://www.google.com/> (Публичный портал электронных карт)

11. Интеграция знаний в цифровых инфраструктурах пространственных данных : монография / С. А. Ямашкин, А. А. Ямашкин, Е. О. Ямашкина, В. В. Занозин. — Саранск : МГУ им. Н.П. Огарева, 2021. — 216 с. — ISBN 978-5-7103-4242-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/311555>

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

Лицензионное программное обеспечение

1. WIN HOME 10 32-bit/64-bit All Lng PK Lic Online DwnLd NR
2. nanoCAD

Свободное ПО

1. 7zip
2. Adobe Acrobat Reader

3. Google Chrome
4. HeidiSQL
5. HK-Software IBExpert Personal Edition
6. LibreOffice
7. Moodle
8. QGIS
9. SQLite
10. STDU Viewer
11. WinDjView

Информационные справочные системы

1. Образовательный портал ВГТУ
<https://old.education.cchgeu.ru/>
2. КонсультантПлюс правовая поддержка
<http://www.consultant.ru/>
3. Электронно-библиотечная система Лань
<https://e.lanbook.com/>
4. База данных «Цифровая библиотека IPRsmart (IPRsmart ONE)»
<http://www.iprbookshop.ru/>
5. Natural Earth Data:

Предлагает векторные и растровые картографические данные в различных масштабах, идеально подходящие для исторических и политических карт.

<https://www.naturalearthdata.com/downloads/>

6. USGS Earth Explorer:

Предоставляет доступ к спутниковым снимкам, аэрофотосъемке и наборам данных о земле.

<https://earthexplorer.usgs.gov/>

7. Esri Open Data Hub:

Платформа для доступа к широкому спектру географических данных.

<https://hub.arcgis.com/search>

8. OpenStreetMap:

Совместный проект по созданию бесплатной редактируемой карты мира.

<https://gisgeography.com/openstreetmap-download-osm-data/>

9. Центр социально-экономических данных и приложений НАСА (SEDAC):

Сосредоточен на взаимодействии человека с окружающей средой.

<https://earthdata.nasa.gov/centers/sedac-daac>

10. Открытая топография:

Специализируется на наборах данных высокого разрешения о земной поверхности, в основном на топографических данных.

<https://opentopography.org/>

11. UNEP Environmental Data Explorer:

Содержит наборы данных, относящихся к экологическим исследованиям, от Программы ООН по окружающей среде.

<https://www.unep.org/publications-data>

12. ArcGIS Living Atlas of the World:

Это крупнейшая коллекция географической информации со всего мира. Он включает карты, приложения, слои данных и многое другое.

<https://livingatlas.arcgis.com/en/home/>

13. Terra Populus:

Интегрирует данные о населении и окружающей среде.

<https://terra.ipums.org/>

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Реализация дисциплины «Разработка расширений для ГИС на языке программирования Python» требует наличия учебной аудитории для проведения учебных занятий

Оборудование учебной аудитории: комплект учебной мебели:

- рабочее место преподавателя (стол, стул);
- рабочие места обучающихся (столы, стулья);

Учебная аудитория для проведения практических работ Лаборатория "Компьютерный класс"/ Лаборатория "Математической обработки результатов геодезических измерений, информационного обеспечения кадастра"

Оборудование учебного кабинета: комплект учебной мебели:

- рабочее место преподавателя (стол, стул);
- рабочие места обучающихся (столы, стулья);

Технические средства обучения:

- Персональные компьютеры с установленным ПО, подключенные к сети Интернет – 14 шт.

Помещение для самостоятельной работы «Аудитория для групповых и индивидуальных консультаций/ Аудитория для самостоятельной работы».

Оборудование кабинета: комплект учебной мебели:

- рабочее место преподавателя (стол, стул);
- рабочие места обучающихся (столы, стулья)

Технические средства обучения:

- интерактивная доска Trace Board TS6080B;

персональный компьютер с возможностью подключения к сети "Интернет" и обеспечением доступа к электронной информационно-образовательной среде вуза

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Разработка расширений для ГИС на языке программирования Python» читаются лекции, проводятся лабораторные работы.

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

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоения учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед зачетом три дня эффективнее всего использовать для повторения и систематизации материала.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Перечень вносимых изменений	Дата внесения изменений	Подпись заведующего кафедрой, ответственной за реализацию ОПОП
----------	-----------------------------	----------------------------	--