

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Забайкальский государственный университет»  
(ФГБОУ ВО «ЗабГУ»)

Горный факультет  
Кафедра открытых горных работ

**УЧЕБНЫЕ МАТЕРИАЛЫ**  
**для студентов заочной формы обучения**  
*(с полным и ускоренным сроком обучения)*

по дисциплине «Цифровое моделирование горных работ»

для специальности 21.05.04 Горное дело направленность «Открытые горные работы»/ «Обогащение полезных ископаемых»/ «Подземная разработка рудных месторождений»/ «Маркшейдерское дело».

Общая трудоемкость дисциплины (модуля) – 4 зачётные единицы.

Формы текущего контроля в семестре – контрольная работа.

Курсовая работа (курсовой проект) (КР, КП) – нет.

Форма промежуточного контроля в семестре – экзамен.

## **Краткое содержание курса**

1. Информация и информационные процессы (разделы «Цифровая грамотность» и «Теоретические основы информатики») Техника безопасности и правила работы на компьютере. Информация и информационные процессы. Виды информации. Хранение информации. Устройства для работы с информацией. Устройство компьютера. Кодирование информации. Код. Процессы кодирования и декодирования. Единицы измерения информации. Файловая система. Одноуровневая и многоуровневая файловые структуры. Путь к файлу. Операции с файлами.

2. Основы языка программирования Python (раздел «Алгоритмы и программирование») Современные языки программирования. Алгоритм. Язык программирования. Программа. Среда разработки IDE. Интерфейс Sculpt. Виды алгоритмов: линейный, разветвляющийся. Переменные. Правила образования имён переменных. Типы данных: целое число, строка. Функция. Виды функций. Функция: print(), input(), int(). Ветвление в Python. Оператор if-else. Вложенное ветвление. Множественное ветвление. Оператор if-elif-else. Проект «Калькулятор».

3. Циклы в языке программирования Python (раздел «Алгоритмы и программирование») Логическое выражение. Простые и сложные логические выражения. Результат вычисления логического выражения. Условие. Операции сравнения в Python. Логические операторы в Python: and, or и not. Операторы целочисленного деления и деления с остатком на Python. Цикл с предусловием. Цикл с параметром. Проект «Оптимизируем расчеты».

## **Форма текущего контроля**

В межсессионный период студенты изучают теоретический материал по учебному курсу и изучают основы программирования на Python по предложенным ниже учебным онлайн ресурсам. По результатам необходимо

выполнить самостоятельные задания. На сессии преподаватель проверяет написанный студентом код и выполняется работа над ошибками в вычислительном классе горного факультета.

### **Методические рекомендации к выполнению практической работы**

Прежде чем приступить к выполнению практической работы, студент должен внимательно ознакомиться с вопросом, уяснить его объем и содержание, изучить теоретический материал по указанным источникам.

При подготовке работы необходимо соблюдать следующие требования:

- программный код должен быть написан в соответствии с рекомендациями учебных курсов и иметь оптимальную структуру;

- выполненные задания должны быть представлены преподавателю в виде электронного файла с расширением \*.ру.

- при затруднении в выполнении работы, дописать код можно будет в период сессии (желательно иметь с собой ноутбук в связи с многочисленностью учебных групп).

Перечень выполняемых заданий и краткие теоретические сведения к выполнению приведены ниже.

#### **Перечень заданий на самостоятельную работу**

##### **Задание 1. Ввод и вывод данных**

Для вывода данных на экран используется функция **print()**.

То что мы хотим вывести на экран пишется в **круглых скобках**. При этом текст указывается внутри **кавычек**. Кавычки могут быть одинарными и двойными.

**Пример:** напишем программу, которая выводит на экран фразу «Мы изучаем цифровое моделирование горных работ»

```
>>> print('Мы изучаем цифровое моделирование горных работ')
```

```
>>> print("Мы изучаем цифровое моделирование горных работ")
```

```
<<< Мы изучаем цифровое моделирование горных работ
```

*! Предыдущие две строки делают одно и то же!*

Команда `print()` позволяет выводить несколько аргументов, в таком случае их **надо отделять запятыми**. Тогда аргументы будут выведены ровно через один пробел.

**Пример:** напишем программу, которая выводит на экран фразу «Мы изучаем цифровое моделирование горных работ», но по-другому.

```
>>> print('Мы', 'изучаем', 'цифровое', 'моделирование', 'горных', 'работ')
<<< Мы изучаем цифровое моделирование горных работ
```

**Пример:** напишем программу, которая выводит числа от 1 до 5.

```
>>> print('1', '2', '3', '4', '5')
<<< 1 2 3 4 5
```

Также можно выводить текст с новой строки.

**Пример:** напишем программу, которая выводит этот текст в две строки.

```
>>> print('Мы изучаем')
print('цифровое моделирование горных работ')
<<<Мы изучаем
цифровое моделирование горных работ
```

Это можно сделать не прописывая функцию `print()` несколько раз, используя параметр `sep`.

**Пример:** используем параметр `sep` для вывода слов с новой строки.

```
>>> print('Мы', 'изучаем', 'цифровое', 'моделирование', 'горных', 'работ', sep='\n')
<<< Мы
изучаем
цифровое
моделирование
горных
работ
```

*!Разделители между объектами могут быть разными. Например, так можно вывести фразы через запятую!*

Все примеры программ до этого момента выводили на экран текст уже написанный внутри. Чтобы работать с данными, введенными в момент выполнения программы используется функция **input()**.

**Пример:** напишем программу, которая будет выводить на экран сообщение о вашем занятии в настоящее время.

```
>>>print ('Что вы сейчас делаете?')
para = input()
print('Я сейчас', para)
<<< Я сейчас изучаю цифровое моделирование горных работ
```

Функция `input()` работает так: когда программа доходит до места, где есть `input()`, она ждет, пока пользователь введёт текст с клавиатуры (ввод завершается нажатием клавиши Enter). Введенная строка подставляется на место `input()`

В нашем случае функция получает от пользователя какие-то данные и записывает их в качестве значения переменной `para`.

Функция `input()` при наличии аргументов внутри неё может работать как вывод текста, а потом ввод текста (именно в этом порядке).

**Пример:** напишем программу, которая будет выводить на экран сообщение о вашем занятии в настоящее время.

```
>>>para = input('Что вы сейчас делаете?')
print('Я сейчас', para)
<<< Я сейчас изучаю цифровое моделирование горных работ
```

## **Задание 2. Работа с текстом.**

2.1. Написать программу, которая выводит на экран текст из таблицы.

2.2. Написать программу, которая выводит на экран текст из таблицы, где вместо пробелов будет «-». Воспользуйтесь параметром `sep`.

2.3. Написать программу, которая выводит на экран текст из таблицы двумя строками, причем первое слово – первая строка, а остальной текст – вторая.

Вариант	Текст
1	<i>Открытые горные работы</i> – совокупность работ, проводимых с земной поверхности с целью добывания разнообразных горных пород и создания различных выемок и котлованов.
2	<i>Технология разработки месторождения</i> – это совокупность способов и приемов механизированного осуществления взаимосвязанных процессов горных работ, основанная на фундаментальных знаниях закономерностей разработки и возможностей технических средств.
3	<i>Вертикальными выработками</i> является большая часть шахтных стволов, шурфы, а также определенное количество выработок, называемых восстающими.
4	<i>Подземные горные выработки</i> имеют различную форму поперечного сечения – прямоугольную, квадратную, трапециевидную, круглую и комбинированную.
5	Для поддержания подземных горных выработок в рабочем состоянии и предотвращения обрушения пород в неустойчивых породах применяют <i>различные виды крепи</i> – дерево, металл, бетон, железобетон и другие крепежные материалы.
6	При выполнении горной графической документации систему подземных горных выработок показывают как вместе с массивом вмещающих ее горных пород, так и отдельно, без изображения горных пород, в которых расположены выработки.
7	Последний вариант дает возможность более наглядно представить взаимное расположение горных выработок в пространстве и показать транспортные, вентиляционные и другие связи между ними.
8	Открытая разработка многих коренных и осадочных месторождений полезных ископаемых ведется карьерами.
9	Карьер – это горное предприятие, представляющее собой совокупность горных выработок, которое ведет разработку месторождения полезного ископаемого открытым способом.
10	Уступ представляет собой часть борта карьера в форме ступени.

### Задание 3. Вывод на экран

Вариант	Задание
1	Написать программу, которая выводит на экран текст «плотность породы =>» и введенное с клавиатуры значение этого параметра.
2	Написать программу, которая считывает три строки, а затем выводит их на экран в обратной последовательности и каждую с новой строки.
3	Написать программу, которая выводит на экран текст «относительный показатель трудности бурения =>» и введенное с клавиатуры значение этого параметра.
4	Написать программу, которая считывает три строки, а затем выводит их на экран, каждую с новой строки.
5	Написать программу, которая выводит на экран текст «По трудности бурения породы относятся к ... классу» и введенное с клавиатуры значение этого параметра вместо «...».
6	Написать программу, которая считывает три строки, а затем выводит их на экран через пробел.
7	Написать программу, которая выводит на экран текст «показатель трудности разрушения породы =>» и введенное с клавиатуры значение этого параметра.
8	Написать программу, которая выводит на экран текст «Породы по трудности бурения относятся к ... классу» и введенное с клавиатуры значение этого параметра вместо «...».
9	Написать программу, которая выводит на экран текст «Диаметр буровой скважины принимается ... мм» и введенное с клавиатуры значение этого параметра вместо «...».
10	Написать программу, которая выводит на экран текст «техническая скорость бурения =>» и введенное с клавиатуры значение этого параметра.

#### Задание 4.

Что будет выведено на экран в результате выполнения кода из таблицы?

Вариант	Код
1	<pre>print(' Общий', ' объем', ' горной', sep='*') print('массы', 'в', ' контурах', sep='-', end='\n') print('карьера', 'является', 'важнейшим', sep='**', end="")</pre>

2	<pre>t = 'Общий' t = 'объем горной массы' print(t)</pre>
3	<pre>print('Борт', 'представленный', 'рабочими', sep='+', end='%') print('уступами', 'называют', 'рабочим', sep='&amp;', end='\n') print('бортом', 'карьера', sep='/', end='!')</pre>
4	<pre>opr = 'Угол между линией, нормальной к простиранию борта и соединяющей верхний и нижний контуры, и горизонталью называется' opr = 'Борт представленный рабочими уступами называют' print(opr, 'рабочим бортом карьера')</pre>
5	<pre># print('Совокупность') # print('уступов') # print('находящихся') print('Мы изучаем', end='+') # print('в одновременной') # print('разработке') print('цифровое', end='=') # print('называется рабочей зоной') print('моделирование горных работ') # карьера</pre>
6	<pre>print('Общий', 'показатель', 'трудности', sep='1', end='%') print('разрушения', 'породы', sep='%') print('Пр', '= 13,6', sep='/', end='!')</pre>
7	<pre>s1 = '1 классу' s2 = '2 классу' s3 = '3 классу' s3 = s2 s1 = s3 print('Данная порода относится к', s3, 'по буримости')</pre>
8	<p>g = 'есть комплексная характеристика породы, определяемая целым рядом ее физико-механических свойств, оказывающих влияние на процесс ее разрушения при бурении. Крепость горной породы есть величина постоянная, не зависящая от способа бурения.'</p>



	<pre>g = 'это величина углубки скважины за единицу времени чистого бурения (механическая скорость бурения). Она оценивается в м/ч, см/мин, мм/мин.' print('Буримость породы' , g)</pre>
9	<pre>s1 = 'Оптимальная' s2 = 'скорость вращения' s3 = 'долота' s3 = s2 s1 = s3 print( 's3')</pre>
10	<pre># print('Продолжительность') # print('смены') # print('Тсм') # print(',') # print('!') print('Кт', end='=') # print('15') print('1,04') # finish</pre>

## Форма промежуточного контроля

Перечень примерных вопросов для подготовки к экзамену.

1. Роль функций в языке Python. Как они объявляются. Какие следует указывать имена функций.
2. Какую роль в Python играет имя функции. Оператор вызова функций.
3. Оператор return в функциях. Особенность(и) его работы.
4. Способы передачи аргументов функции. Фактические и формальные параметры.
5. Способы объявлений функций с произвольным числом фактических и формальных параметров. Возможные их комбинации.
6. Оператор упаковки и распаковки коллекций. Примеры использования.
7. Рекурсивные функции. Преимущества и недостатки. Ограничения их использования.
8. Области видимости переменных. Ключевые слова global и nonlocal.

9. Анонимные (лямбда) функции. Их отличия от обычных функций языка Python. Области применения лямбда-функций.
10. Замыкание функций. Области применения замыканий функций.
11. Декораторы функций. Способы декорирования через оператор @ и напрямую (без оператора @). Отличия декорирования функции через оператор @ и без него.
12. Декораторы с параметрами. Примеры декорирования функции такими декораторами.
13. Модуль языка Python. Отличие модулей от пакетов. Способы импорта модулей.
14. Пакеты в языке Python. Структура пакетов. Инициализатор пакета \_\_init\_\_.py. Способы импорта пакетов в основной модуль программы.
15. Рекомендации способов импорта по стандарту PEP8.
16. Команды для установки пакетов. Пакетная установка (сразу нескольких модулей или пакетов).
17. Функция для работы с файлами open. Текстовый и бинарный режимы доступа к файлу.
18. Функции записи данных в файл в текстовом режиме доступа.
19. Функции чтения данных из файла в текстовом режиме доступа.
20. Функции чтения данных из файла в бинарном режиме доступа. Модуль
21. Pickle.
22. Обработка ошибок (исключений) блоками try/except. Отличия применения блоков try/except от условных операторов if/elif/else.
23. Выражение-генератор. Как задается, для чего нужен. Перебор значений выражения-генератора.
24. Функция-генератор. Оператор yield. Отличия функции-генератора от выражения-генератора. Перебор значений функции-генератора.
25. Функции map и filter. Пример их совместной работы.
26. Функции all и any.
27. Функция sorted.
28. Функция isinstance.
29. Функция zip.
30. Итератор. Перебор коллекций с помощью итератора функцией next и оператором for. Исключение StopIteration.

Для изучения курса рекомендуются следующие источники.

1. <https://docs.anaconda.com/anaconda/install/windows/> - анаконда  
(можно использовать миниконда) для работе с Jupyter Notebook
2. <https://www.jetbrains.com/pycharm/> - PyCharm
3. <https://www.jetbrains.com/pycharm-edu/> - PyCharm для образования
4. <https://trinket.io/> - онлайн платформа
5. <https://www.python.org/> - IDLE
6. Google Colaboratory – совместная работа с документом
7. <https://pythonworld.ru/osnovy/pep-8-rukovodstvo-po-napisaniyu-koda-na-python.html> - руководство по написанию кода на Python

Ведущий преподаватель

А.Е. Фомина