

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО

**Директор физтех-школы физики
и исследований им. Ландау
А.В. Рогачев**

	Рабочая программа дисциплины (модуля)
по дисциплине:	Программирование и обработка данных на языке Python
по направлению:	Прикладные математика и физика
профиль подготовки:	Общая и прикладная физика Физтех-школа физики и исследований им. Ландау кафедра инновационной педагогики
курс:	1
квалификация:	магистр

Семестры, формы промежуточной аттестации:

1 (осенний) - Зачет
2 (весенний) - Экзамен

Аудиторных часов: 120 всего, в том числе:

лекции: 60 час.
семинары: 60 час.
лабораторные занятия: 0 час.

Самостоятельная работа: 75 час.

Подготовка к экзамену: 30 час.

Всего часов: 225, всего зач. ед.: 5

Программу составил: Р.С. Кулиев, старший преподаватель

Программа обсуждена на заседании кафедры инновационной педагогики 04.04.2022

Аннотация

Язык Python является одним из самых простых, и в то же время востребованных на рынке труда языков программирования. Он используется в различных сферах прикладного применения: от научных вычислений и анализа данных до разработки промышленных решений.

Целью учебной дисциплины является формирование компетенций студентов в области решения задач в Python, применения шаблонов проектирования на Python, работы с Python библиотеками.

Программа предусматривает изучение следующих тем:

- Возможности языка Python и его особенности
- Основы работы в Pandas

1. Цели и задачи

Цель дисциплины

Обучить слушателей основам программирования на языке Python, необходимым для анализа данных на практике, и познакомить со связанными технологиями анализа, визуализации и наглядного представления данных.

Задачи дисциплины

- познакомить студентов с основными конструкциями, объектами и процедурами языка Python;
- сформировать навыки написания эффективного, простого, понятного и гибкого кода, оптимального с точки зрения повышения скорости и качества разработки;
- научить методам обработки ошибок и тестирования кода на языке Python;
- обучить обработке данных с помощью языка Python и основам функционального программирования;
- обучить технологиям анализа и визуализации представления данных.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	УК-1.1 Анализирует проблемную ситуацию как систему, выявляя ее составляющие и связи между ними
УК-6 Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки	УК-6.1 Умеет решать задачи собственного личностного и профессионального развития, определять и реализовывать приоритеты совершенствования собственной деятельности
ОПК-1 Владеет системой фундаментальных научных знаний в области физико-математических наук	ОПК-1.1 Знает и способен использовать в профессиональной деятельности фундаментальные научные знания в области физико-математических наук
ОПК-3 Способен выбирать и (или) разрабатывать подходы к решению типовых и новых задач в области профессиональной деятельности, учитывая особенности и ограничения различных методов решения	ОПК-3.1 Способен анализировать задачу, планировать пути решения, предлагать и комбинировать способы решения
	ОПК-3.3 Владеет аналитическими и вычислительными методами решения, понимает и учитывает на практике границы применимости получаемых решений
ОПК-4 Способен успешно реализовывать решение поставленной задачи, провести анализ результата и представить выводы, применяя знания и навыки в области физико-математических наук и информационно-коммуникационных технологий	ОПК-4.1 Способен применять знания и навыки по использованию информационно-коммуникационных технологий для поиска и изучения научной литературы, применения прикладных программных продуктов
	ОПК-4.2 Способен применять знания в области физико-математических наук для решения поставленной задачи, формулирования выводов и оценки полученных результатов

ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
---	--

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- ☐ возможности языка Python и его особенности;
- ☐ основы работы в Pandas;
- ☐ синтаксис Python, особенности программирования на Python;
- ☐ стандартные библиотеки языка программирования;
- ☐ методологии и технологии проектирования и использования баз данных;
- ☐ технологии программирования;
- ☐ особенности выбранной среды программирования и системы управления базами данных;
- ☐ компоненты программно-технических архитектур;
- ☐ существующие приложения и интерфейсы взаимодействия с ними.

уметь:

- ☐ работать с данными и таблицами на Python;
- ☐ работать с данными в Pandas;
- ☐ применять Python для написания программного кода;
- ☐ использовать выбранную среду программирования и средства системы управления базами данных; использовать возможности имеющейся технической и/или программной архитектуры.

владеть:

- ☐ технологиями программирования на Python.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Установка Python. Знакомство со средой разработки	2	2		2
2	Базовые типы. Конструкции управления потоком: условия. Модули и пакеты	2	4		2
3	Конструкции управления потоком: циклы	4	4		2
4	Знакомство с коллекциями: множества, строки, списки, кортежи	2	2		3
5	Методы списков и строк. Библиотека Numpy	4	4		3
6	Работа со словарями. Списочные выражения	4	2		3
7	Функции: возвращение значения из функции, области видимости переменных, передача параметра. Рекурсия. Функции с переменным числом аргументов	4	2		3

8	Функции как объект. Лямбда-функции	2	4		3
9	Сортировки 1: асимптотическая сложность алгоритмов, сортировка пузырьком, подсчётом	2	2		3
10	Сортировки 2: сортировка слиянием, пирамидальная сортировка	2	2		3
11	Сортировки 3: быстрая сортировка, поразрядная сортировка	2	2		3
12	Библиотеки Python. Встроенные модули: math, random, datetime, pprint	4	2		2
13	Библиотека Open CV	4	2		2
14	Знакомство с библиотекой Numpy	4	2		5
15	Визуализация с matplotlib	2	4		5
16	Знакомство с библиотекой Pandas. Объекты pandas.Series и pandas.DataFrame	2	4		5
17	Анализ взаимосвязей. Описательная статистика. Сводные таблицы	2	2		5
18	Итерирование данных. Работа с несколькими таблицами	2	4		5
19	Работа с файлами	2	2		5
20	Библиотека tkinter для разработки графического интерфейса	4	2		5
21	Извлечение данных из веб-страницы	2	4		3
22	Работа с классами на примере API Yandex	2	2		3
Итого часов		60	60		75
Подготовка к экзамену		30 час.			
Общая трудоёмкость		225 час., 5 зач.ед.			

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 1 (Осенний)

1. Установка Python. Знакомство со средой разработки

Знакомство с языком. Установка интерпретатора Python. Настройка окружения для работы и выбор среды разработки.

2. Базовые типы. Конструкции управления потоком: условия. Модули и пакеты

Знакомство с базовыми типами языка и конструкцией if_else

3. Конструкции управления потоком: циклы

Знакомство с циклическими конструкциями for, while

4. Знакомство с коллекциями: множества, строки, списки, кортежи

Знакомство с новыми типами данных — коллекциями и методами работы с ними. Использование в коде списочных выражений.

5. Методы списков и строк. Библиотека Numpy

Знакомство с библиотекой NumPy, содержащей большое количество полезных инструментов: от быстрых операций с многомерными массивами до реализации различных математических методов.

6. Работа со словарями. Списочные выражения

Знакомство с коллекцией Dict и OrderedDict стандартной библиотеки Python.

7. Функции: возвращение значения из функции, области видимости переменных, передача параметра. Рекурсия. Функции с переменным числом аргументов

Предпосылки появления функций в языках программирования, шаблоны их реализации и использования. Примеры использования параметров, в том числе по умолчанию, возвращаемых значений, лямбда-функций.

8. Функции как объект. Лямбда-функции

Понятие лямбда-функции. Примеры лямбда-функций.

9. Сортировки 1: асимптотическая сложность алгоритмов, сортировка пузырьком, подсчётом

Понятие асимптотической сложности, примеры алгоритмов, оперирующих числами, в том числе вычисляющих простейшие последовательности. Фундаментальные алгоритмы сортировок. Сортировка пузырьком, подсчетом.

10. Сортировки 2: сортировка слиянием, пирамидальная сортировка

Фундаментальные алгоритмы сортировок. Сортировка слиянием, пирамидальная сортировка.

11. Сортировки 3: быстрая сортировка, поразрядная сортировка

Фундаментальные алгоритмы сортировок. Быстрая сортировка, поразрядная сортировка.

Семестр: 2 (Весенний)

12. Библиотеки Python. Встроенные модули: math, random, datetime, pprint

Библиотеки Python. Встроенные модули: math, random, datetime, pprint. Инструменты работы с библиотеками Python.

13. Библиотека Open CV

Особенности работы с библиотекой Open CV.

14. Знакомство с библиотекой NumPy

Знакомство с элементами линейной алгебры — векторами и матрицами, как основным математическим аппаратом для работы с данными. Использование функций модуля numpy для выполнения операций с матрицами и векторами.

15. Визуализация с matplotlib

Рассмотрение объектного подхода к построению диаграмм и графиков различных типов.

16. Знакомство с библиотекой Pandas. Объекты pandas.Series и pandas.DataFrame

Использование средств модуля pandas для загрузки данных и ознакомления с ними. Знакомство с основными функциями для работы с сериями и датафреймами в pandas.

17. Анализ взаимосвязей. Описательная статистика. Сводные таблицы

Корреляционный анализ данных и оценка силы и направления взаимосвязи. Меры средней тенденции, меры разброса. Механизм GroupBy, типы группировок. Агрегирование, преобразование, исключение групп. Подготовка данных для создания сводных таблиц. Представление сводных таблиц.

18. Итерирование данных. Работа с несколькими таблицами

Слияние и конкатенация датафреймов, форматы "Long" и "Wide".

19. Работа с файлами

Методы работы с файлами, в том числе запись видео.

20. Библиотека tkinter для разработки графического интерфейса

Использование библиотеки tkinter для создания графического интерфейса.

21. Извлечение данных из веб-страницы

Извлечение данных из веб-страницы. Использование модулей и сервисов, реализующих веб-запросы и облачную обработку.

22. Работа с классами на примере API Yandex

Работа с классами на примере API Yandex.

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Лекционная аудитория, оснащенная мультимедиапроектором.

6.Перечень рекомендуемой литературы

Основная литература

1. Automate the Boring Stuff with Python [Электронный ресурс] – Режим доступа - <https://automatetheboringstuff.com/>
2. Dive Into Python 3 [Электронный ресурс] – Режим доступа - <http://diveintopython3.problemsolving.io/>
3. Problem Solving with Algorithms and Data Structures using Python [Электронный ресурс] – Режим доступа - <https://runestone.academy/runestone/static/pythonds/index.html>
4. Swaroop Chitlur. A Byte of Python [Электронный ресурс] – Режим доступа - <https://wombat.org.ua/AByteOfPython/AByteofPythonRussian-2.02.pdf> – 2020.
5. Федоров Д. Основы программирования на примере языка Python [Текст] : учебное пособие / Д. Федоров. - 2018.
6. Свейгарт Э. Автоматизация рутинных задач с помощью Python [Текст] : практическое руководство для начинающих / Эл Свейгарт. – 2017.

Дополнительная литература

1. Бэрри П. Изучаем программирование на Python [Текст] / П. Бэрри.– 2017.
2. Савельев В. Статистика и коттики [Текст] / Владимир Савельев. – 2018.
3. Бослав С. Статистика для всех [Текст] / Сара Бослав.– 2015.
4. Хамидуллин Р. Я. Теория вероятностей и математическая статистика [Текст] / Р.Я. Хамидуллин.– 2020.
5. Справочник DAX функций для Power BI и PowerPivot [Электронный ресурс] / Антон Будув – Режим доступа - <https://biprosto.ru/dax/functions.html>
6. Талер Р. Nudge. Архитектура выбора [Текст] / Ричард Талер. – 2017.

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Не используются

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

Документация Postgres про сравнение строк

<https://postgrespro.ru/docs/postgrespro/9.5/functions-matching>

Документация Postgres про другие функции работы со строками

<https://postgrespro.ru/docs/postgrespro/9.5/functions-string>

Тестер регулярных выражений <https://www.regextester.com>

Интерактивный учебник по SQL <http://www.sql-tutorial.ru/ru/content.html>

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Студент, изучающий дисциплину, должен, с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике.

В результате изучения дисциплины студент должен знать основные определения и понятия, уметь применять полученные знания для решения различных задач.

Успешное освоение курса требует:

- посещения всех занятий, предусмотренных учебным планом по дисциплине;
- ведения конспекта занятий;
- напряжённой самостоятельной работы студента.

Самостоятельная работа включает в себя:

- чтение рекомендованной литературы;
- проработку учебного материала, подготовку ответов на вопросы, предназначенных для самостоятельного изучения;
- решение задач, предлагаемых студентам на занятиях;
- подготовку к выполнению заданий текущей и промежуточной аттестации.

Показателем владения материалом служит умение без конспекта отвечать на вопросы по темам дисциплины.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов, следует обращаться за консультациями преподавателю.

Возможен промежуточный контроль знаний студентов в виде решения задач в соответствии с тематикой занятий.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Прикладные математика и физика
профиль подготовки:	Общая и прикладная физика Физтех-школа физики и исследований им. Ландау кафедра инновационной педагогики
курс:	1
квалификация:	магистр

Семестры, формы промежуточной аттестации:

- 1 (осенний) - Зачет
- 2 (весенний) - Экзамен

Разработчик: Р.С. Кулиев, старший преподаватель

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	УК-1.1 Анализирует проблемную ситуацию как систему, выявляя ее составляющие и связи между ними
УК-6 Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки	УК-6.1 Умеет решать задачи собственного личностного и профессионального развития, определять и реализовывать приоритеты совершенствования собственной деятельности
ОПК-1 Владеет системой фундаментальных научных знаний в области физико-математических наук	ОПК-1.1 Знает и способен использовать в профессиональной деятельности фундаментальные научные знания в области физико-математических наук
ОПК-3 Способен выбирать и (или) разрабатывать подходы к решению типовых и новых задач в области профессиональной деятельности, учитывая особенности и ограничения различных методов решения	ОПК-3.1 Способен анализировать задачу, планировать пути решения, предлагать и комбинировать способы решения
	ОПК-3.3 Владеет аналитическими и вычислительными методами решения, понимает и учитывает на практике границы применимости получаемых решений
ОПК-4 Способен успешно реализовывать решение поставленной задачи, провести анализ результата и представить выводы, применяя знания и навыки в области физико-математических наук и информационно-коммуникационных технологий	ОПК-4.1 Способен применять знания и навыки по использованию информационно-коммуникационных технологий для поиска и изучения научной литературы, применения прикладных программных продуктов
	ОПК-4.2 Способен применять знания в области физико-математических наук для решения поставленной задачи, формулирования выводов и оценки полученных результатов
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности

2. Показатели оценивания компетенций

В результате изучения дисциплины «Программирование и обработка данных на языке Python» обучающийся должен:

знать:

- ☐ возможности языка Python и его особенности;
- ☐ основы работы в Pandas;
- ☐ синтаксис Python, особенности программирования на Python;
- ☐ стандартные библиотеки языка программирования;
- ☐ методологии и технологии проектирования и использования баз данных;
- ☐ технологии программирования;
- ☐ особенности выбранной среды программирования и системы управления базами данных;
- ☐ компоненты программно-технических архитектур;
- ☐ существующие приложения и интерфейсы взаимодействия с ними.

уметь:

- ☐ работать с данными и таблицами на Python;
- ☐ работать с данными в Pandas;
- ☐ применять Python для написания программного кода;
- ☐ использовать выбранную среду программирования и средства системы управления базами данных; использовать возможности имеющейся технической и/или программной архитектуры.

владеть:

- ☐ технологиями программирования на Python.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

С целью контроля освоения обучающимися учебного материала проводится устный опрос в начале занятия по теме прошлого занятия.

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

Критерии оценивания

Оценка отлично 10 баллов - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично 9 баллов – выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично 8 баллов – выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.

Оценка хорошо 7 баллов – выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты.

Оценка хорошо 6 баллов – выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Оценка хорошо 5 баллов – выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.

Оценка удовлетворительно 4 балла – выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.

Оценка удовлетворительно 3 балла – выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.

Оценка неудовлетворительно 2 балла – выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных принципов и не умеет использовать полученные знания при решении типовых задач.

Оценка неудовлетворительно 1 балл – выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубейшие ошибки в формулировках базовых понятий дисциплины и вообще не имеет навыков решения типовых практических задач.

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

Осенний семестр. Для получения зачета необходимо выполнить одно из контрольных заданий. Зачет проводится в форме защиты контрольного задания. Опрос обучающегося не должен превышать одного астрономического часа.

Весенний семестр. Экзамен проводится в форме защиты итогового проекта. Опрос обучающегося не должен превышать одного астрономического часа.

Программирование и обработка данных на языке Python

Перечень контрольных вопросов

1. Условный оператор и каскадная условная конструкция в Python;
2. Строки в Python. Методы `find`, `count`, `replace`, `strip`, `upper`, `lower` и другие;
3. Строки в Python. Срезы с двумя и тремя параметрами;
4. Кортежи в Python. Создание и использование. Кортежи переменных;
5. Списки в Python. Методы списков и операции со списками;
6. Срезы списков. Присваивание в срез. Проблема копирования списка;
7. Цикл `for` и его особенности в Python. Функция `range()`;
8. Двумерные массивы (списки списков). Вложенная генерация;
9. Класс и экземпляр класса, методы экземпляра. Понятие атрибута;
10. Классы в Python. Перегрузка операторов;
11. Классы в Python. Конструктор класса;
12. Классы в Python. Наследование;
13. Словарь, список, множество, перебор с помощью `for`;
14. Функции: возвращение значения из функции, области видимости переменных, передача параметра;
15. Рекурсия. Функции с переменным числом аргументов;
16. Работа с файлами. Работа с каталогами;
17. Обработка параметров командной строки;
18. Функция как объект. Лямбда-функции;
19. Понятие сложности алгоритма;
20. Сортировка методом пузырька;
21. Сортировка подсчётом;
22. Поразрядная сортировка;
23. Сортировка слиянием;
24. Пирамидальная сортировка;
25. Быстрая сортировка;
26. Библиотека `Open CV`;
27. Библиотека `Numpy`;
28. Визуализация с библиотекой `matplotlib`;
29. Библиотека `Pandas`;
30. Библиотека `tkinter` для разработки графического интерфейса.

Примеры контрольных заданий

Задание №1 (по теме 1)

Вам дан двумерный массив `data`. В каждой строке записана информация об эмодзи. А именно название, количество (в миллионах) использованных эмодзи данного типа за исследуемый период в трех различных социальных сетях: `EmojiExpress`, `Instagram`, `Twitter`.

In []:

```
data = [  
    ['grinning face', 2.26, 1.02, 87.3],  
    ['winking face', 19.1, 1.69, 150.0],  
    ['star-struck', 233.0, 7.31, 270.0],  
    ['smiling face', 15.2, 2.36, 264.0],  
    ['thinking face', 22.7, 4.26, 565.0],  
    ['neutral face', 6.81, 0.636, 0.0],  
    ['pensive face', 6.0, 0.236, 478.0],  
    ['cold face', 24.7, 1.35, 654.0],  
    ['hot face', 10.0, 5.69, 445.0],
```

```

['worried face', 118.0, 26.0, 180.0],
['fearful face', 23.1, 3.75, 227.0],
['crying face', 1.74, 0.11, 0.0],
['tired face', 4.5, 2.49, 150.0],
['pouting face', 0.0333, 0.056, 932.0]
]

```

Задание

- 1) Вам нужно добавить ещё один столбец - суммарное по трем указанным платформам количество использований эмодзи данного типа.
- 2) Необходимо отсортировать данные по данному признаку, так чтобы в первой строке был самый часто используемый эмоджи, а в последней - наименее популярный.
- 3) Вывести таблицу при помощи форматированного вывода.

Пример оформления:

<https://yadi.sk/i/SmReFZ9drKgCcQ>

Пример выполненного задания

In [18]:

```

data= [
['grinning face', 2.26, 1.02, 87.3],
['winking face', 19.1, 1.69, 150.0],
['star-struck', 233.0, 7.31, 270.0],
['smiling face', 15.2, 2.36, 264.0],
['thinking face', 22.7, 4.26, 565.0],
['neutral face', 6.81, 0.636, 0.0],
['pensive face', 6.0, 0.236, 478.0],
['cold face', 24.7, 1.35, 654.0],
['hot face', 10.0, 5.69, 445.0],
['worried face', 118.0, 26.0, 180.0],
['fearful face', 23.1, 3.75, 227.0],
['crying face', 1.74, 0.11, 0.0],
['tired face', 4.5, 2.49, 150.0],
['pouting face', 0.0333, 0.056, 932.0]
]

```

```

for i in range(len(data)):

```

```

data[i].append(sum(data[i][1:]))

data.sort(key=lambda row: row[4], reverse=True)

headings = ['Emoji short name', 'EmojiXpress', 'Instagram', 'Twitter', 'Total use']

print(*headings, sep=' | ')

print('-'* (sum(len(h) for h in headings) + 3* (len(headings) - 1)))

for row in data:

print('{: <16} | {: >11.2f} | {: >9.2f} | {: >7.2f} | {: >9.2f}'.format(*row))

Emoji short name | EmojiXpress | Instagram | Twitter | Total use
-----
pouting face | 0.03 | 0.06 | 932.00 | 932.09
cold face | 24.70 | 1.35 | 654.00 | 680.05
thinking face | 22.70 | 4.26 | 565.00 | 591.96
star-struck | 233.00 | 7.31 | 270.00 | 510.31
pensive face | 6.00 | 0.24 | 478.00 | 484.24
hot face | 10.00 | 5.69 | 445.00 | 460.69
worried face | 118.00 | 26.00 | 180.00 | 324.00
smiling face | 15.20 | 2.36 | 264.00 | 281.56
fearful face | 23.10 | 3.75 | 227.00 | 253.85
winking face | 19.10 | 1.69 | 150.00 | 170.79
tired face | 4.50 | 2.49 | 150.00 | 156.99
grinning face | 2.26 | 1.02 | 87.30 | 90.58
neutral face | 6.81 | 0.64 | 0.00 | 7.45
crying face | 1.74 | 0.11 | 0.00 | 1.85

```

Задание №2 (по теме 2)

В данном задании Вам предстоит работать с данными при помощи библиотеки Pandas. Последовательно выполните следующее:

- 1) прочитайте данные из файла titanic.csv и сохраните в переменную типа DataFrame;
- 2) выведите последние 10 строк таблицы, а также список с названиями всех столбцов;
- 3) переведите названия всех столбцов в нижний регистр;

- 4) выведите имя, пол и возраст для клиентов с индексами от 10 до 17;
- 5) выведите возраст первых 5 человек. Тип выводимого значения должен быть целочисленным;
- 6) выведите имя, пол и возраст первых 10 человек со значением Class равным 3;

Для выполнения задания используйте библиотеку matplotlib.

1. На основании данных о возрасте клиентов составьте гистограмму распределения возрастов.
2. На основании данных о Class составьте круговую диаграмму.

Пример выполненного задания

In [2]:

```
import pandas as pd
```

```
% matplotlib inline
```

In [11]:

```
data = pd.read_csv('titanic.csv')
```

Вывести последние 10 строк таблицы

In [13]:

```
data.tail(10)
```

Out[13]:

	ClientId	Survived	Class	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
881	882	0	3	Markun, Mr. Johann	male	33.0	0	0	349257	7.8958	NaN	S
882	883	0	3	Dahlberg, Miss. Gerda Ulrika	female	22.0	0	0	7552	10.5167	NaN	S
883	884	0	2	Banfield, Mr. Frederick James	male	28.0	0	0	C.A./SOTON 34068	10.5000	NaN	S
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTON/OQ 392076	7.0500	NaN	S
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.1250	NaN	Q
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S

	ClientId	Survived	Class	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
887	888	1	1	Graham, Miss. MargaretEdith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. KarlHowell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

Вывести названия колонок

In [30]:

```
data.columns
```

Out[30]:

```
Index(['ClientId', 'Survived', 'Class', 'Name', 'Sex', 'Age', 'SibSp',  
      'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],  
      dtype='object')
```

Поменять регистр всех букв в названии колонок на нижний

In [14]:

```
data = data.rename(str.lower, axis="columns")
```

Вывести имена, пол и возраст клиентов 10-17

In [15]:

```
data[['name', 'sex', 'age']][10:18]
```

Out[15]:

	name	sex	age
10	Sandstrom, Miss. MargueriteRut	female	4.0
11	Bonnell, Miss. Elizabeth	female	58.0

	name	sex	age
12	Saunderscock, Mr. WilliamHenry	male	20.0
13	Andersson, Mr. AndersJohan	male	39.0
14	Vestrom, Miss. Hulda Amanda Adolfina	female	14.0
15	Hewlett, Mrs. (Mary D Kingcome)	female	55.0
16	Rice, Master. Eugene	male	2.0
17	Williams, Mr. CharlesEugene	male	NaN

Вывести 5 целых чисел (типа int32) - возраст первых 5 человек

In [46]:

```
data['age'].head().astype(int)
```

Out[46]:

```
0    22
1    38
2    26
3    35
4    35
```

Name: age, dtype: int32

Вывести имя, пол и возраст первых 10 человек, со значением Class равным 3

In [50]:

```
data[data['pclass'] == 3].iloc[:10, 2:6]
```

Out[50]:

	Class	name	sex	age
0	3	Braund, Mr. OwenHarris	male	22.0

	Class		name	sex	age
2	3		Heikkinen, Miss. Laina	female	26.0
4	3		Allen, Mr. WilliamHenry	male	35.0
5	3		Moran, Mr. James	male	NaN
7	3		Palsson, Master. GostaLeonard	male	2.0
8	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)		female	27.0
10	3		Sandstrom, Miss. MargueriteRut	female	4.0
12	3		Saunderscock, Mr. WilliamHenry	male	20.0
13	3		Andersson, Mr. AndersJohan	male	39.0
14	3		Vestrom, Miss. Hulda Amanda Adolfina	female	14.0

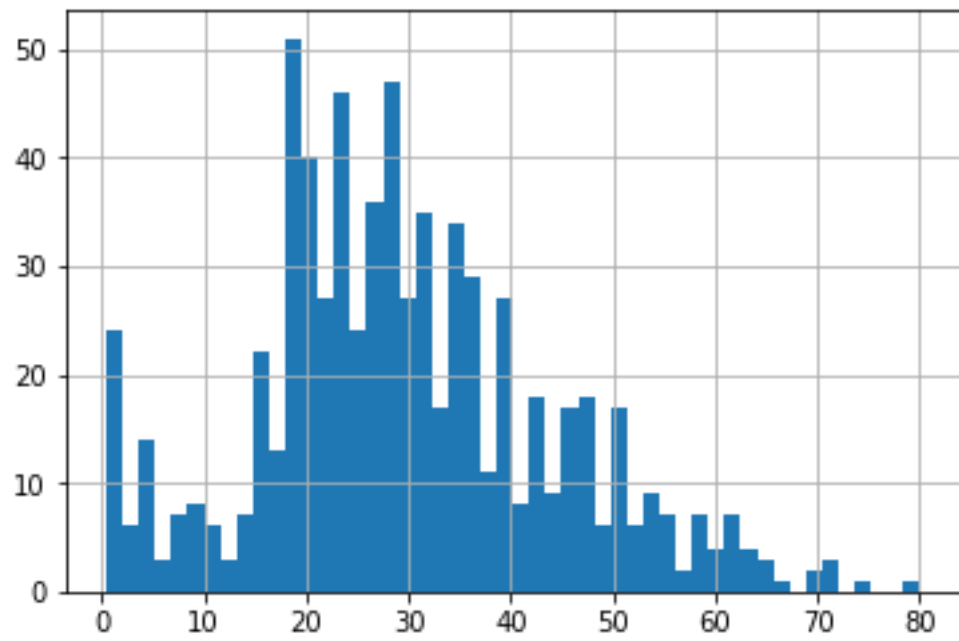
Построить гистограмму по возрасту

In [16]:

```
data.age.hist(bins =50)
```

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xa9e80b8>
```



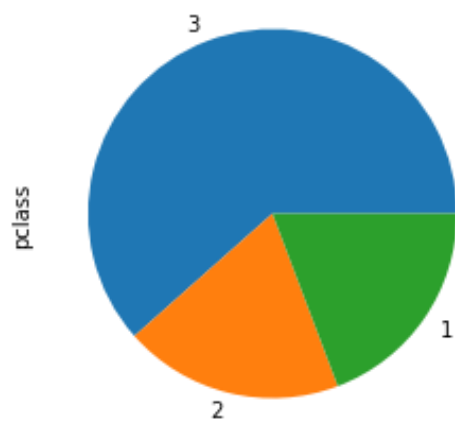
Построить круговую диаграмму по номеру pclass

In [8]:

```
data['pclass'].value_counts().plot(kind='pie')
```

Out[8]:

<matplotlib.axes._subplots.AxesSubplot at 0x953ee80>



Пример итогового проекта

Вам дан двумерный массив `clients_data`. В каждой строке записана информация о клиенте компании. А именно фамилия, год начала пользования услугами компании, рейтинг лояльности компании, возраст, а также список тарифов компании, которыми пользуется клиент

In []:

```
clients_data= [

    ['Ivanov', 2017, 6.914, 23, ['T1', 'T2']],

    ['Petrov', 2016, 6.151, 51, ['T2']],

    ['Sidorov', 2015, 7.489, 29, ['T2', 'T3']],

    ['Kuznetsov', 2013, 7.71, 33, ['T2', 'T4', 'T3']],

    ['Abramov', 2011, 7.942, 69, ['T2', 'T5', 'T6']],

    ['Averin', 2010, 7.977, 18, ['T2', 'T4', 'T3']],

    ['Borisov', 2008, 7.298, 26, ['T7', 'T2', 'T3']],

    ['Baranov', 2007, 7.726, 22, ['T7', 'T2']],

    ['Bogdanov', 2006, 8.456, 51, ['T7', 'T2']],

    ['Vasilev', 2004, 7.896, 18, ['T7', 'T2']],

    ['Voronov', 2002, 7.669, 19, ['T8', 'T6']],

    ['Gurov', 2001, 8.557, 35, ['T2', 'T4', 'T5']],

    ['Gridnev', 2000, 8.585, 55, ['T9', 'T2', 'T10']],

    ['Godunov', 1997, 8.369, 94, ['T2', 'T5']],

    ['Domnin', 1996, 7.849, 55, ['T2', 'T5']],

    ['Dolin', 1995, 8.283, 78, ['T2', 'T4', 'T3']],

    ['Kalinin', 1994, 8.915, 42, ['T2', 'T5']],

    ['Karin', 1993, 8.819, 55, ['T2', 'T4', 'T3']],

    ['Lobanov', 1992, 7.858, 31, ['T2', 'T11']],

    ['Levin', 1990, 8.335, 43, ['T7', 'T12', 'T2']],

    ['Lomonosov', 1990, 8.112, 81, ['T2', 'T10', 'T11']],

    ['Malkin', 1989, 7.645, 99, ['T2']],

    ['Mironov', 1988, 8.25, 33, ['T2']],

]
```

1) Вам нужно реализовать вспомогательную функцию, принимающую на вход Ваш массив и два числа - границы отрезка. Ваша функция должна вернуть подмассив с данными о клиентах, рейтинг лояльности которых принадлежит этому отрезку.

2) Для двух произвольных отрезков вывести подтаблицу, сгенерированную первой функцией при помощи форматированного вывода. Тарифы выводить не нужно. Дополнительно вывести среднее значение возраста клиента и средний рейтинг лояльности для каждой подтаблицы.

3) Для каждого тарифа посчитайте количество клиентов, которые импользуются, и создайте таблицу, состоящую из двух столбцов: "tariff" и "amountofclients". Данные в таблице отсортировать по убыванию количества клиентов.

Пример оформления таблицы:

<https://yadi.sk/i/SmReFZ9drKgCcQ>