

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)**

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО
ПРОГРАММИРОВАНИЮ НА ЯЗЫКЕ PYTHON**

САМАРА 2017

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)

ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ НА ЯЗЫКЕ PYTHON

Рекомендовано учебно-методическим советом (РИК) института информатики, математики и электроники федерального государственного автономного образовательного учреждения высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева» в качестве методических указаний

САМАРА
2017

УДК 510.2

Составители *Т.П.Рубцова, М.В.Морозова*

Рецензент к.ф.-м.н., доцент Сиников В.М.

Программирование на языке Python: метод. указания / сост. *Т.П.Рубцова, М.В.Морозова* – Самара, 2017. – 48 с.: ил.

Предназначен для проведения лабораторных работ по курсам «Практикум на ЭВМ», «Программирование» и «Учебная практика» для студентов бакалавриата, обучающихся по направлению «Прикладная математика и информатика», «Математическое обеспечение и администрирование информационных систем».

Подготовлено на кафедре информатики и вычислительной математики.

УДК 510.2

Содержание

Требования к выполнению лабораторных работ	4
Лабораторная работа №1. Создание типа данных «класс»	6
Лабораторная работа №2. Наследование и полиморфизм.....	10
Лабораторная работа №3. Работа с базами данных	14
Лабораторная работа №4. Связанные структуры данных	19
Лабораторная работа №5. Работа с графикой	24
Лабораторная работа №6. Сортировки	27
Лабораторная работа №7. Создание многопоточных приложений	30
Лабораторная работа №8. Линейная регрессия.....	31
Лабораторная работа №9. Логистическая регрессия	38
Лабораторная работа №10. Визуализация данных на языке Python с помощью библиотеки matplotlib	45
Список литературы	47

Требования к выполнению лабораторных работ

В ходе выполнения лабораторной работы студент должен написать программный код на языке программирования Python, выполняющий основное задание лабораторной работы, оформить отчет и отчитаться преподавателю по итогам выполненной лабораторной работы. Далее подробно описаны требования по выполнению каждого этапа лабораторной работы.

Программный код

Студент должен программный код, выполняющий основное задание лабораторной работы.

К программному коду предъявляются следующие требования:

- 1) именование переменных должно быть осмысленным;
- 2) в коде программы должны присутствовать комментарии;
- 3) код должен производить форматированный вывод результатов;
- 4) код должен запрашивать пользователя входные данные;
- 5) код должен иметь базовый уровень проверки входных данных.

Отчет

В ходе работы для получения зачета студент обязан написать отчет о выполненной работе. Шаблон отчета предоставляется преподавателем.

Структура отчета приведена ниже:

1. Титульный лист. На титульном листе указываются номер лабораторной работы, фамилия и инициалы (ФИО) студента, номер группы, ФИО преподавателя.
2. Задание. Задание на лабораторную работу копируется из методических указаний согласно варианту.
3. Ход выполнения лабораторной работы. В данной части подробно описывается процесс выполнения лабораторной работы, приводятся листинг, результаты работы программы и их интерпретация.
4. Заключение. В заключении описывается основной результат, полученный в ходе выполнения работы, формируются выводы по проделанной работе.
5. Список литературы. Список литературы должен содержать минимум две ссылки.

К отчету предъявляются следующие требования:

1. структура отчета должна быть строго соблюдена;
2. отчет должен быть отформатирован;
3. каждая структурная часть должна начинаться с новой страницы;
4. все страницы в отчете, кроме титульной, должны быть пронумерованы;
5. отчет должен содержать описание всех особенностей реализации кода.

Защита перед преподавателем

Для получения зачета студент обязан в срок защитить работу перед преподавателем.

Защита лабораторной работы может состояться только:

- 1) при наличии работающего программного кода, выполняющего основное задание лабораторной работы;
- 2) при наличии отчета по итогам выполненной лабораторной работы.

В ходе защиты лабораторной работы студент обязан продемонстрировать владение материалом лабораторной работы, ответить на дополнительные вопросы преподавателя.

Лабораторная работа №1. Создание типа данных «класс»

1. Создать класс с полями, указанными в индивидуальном задании.

2. Реализовать в классе методы:

✓ конструктор по умолчанию;

✓ деструктор для освобождения памяти (с сообщением об уничтожении объекта);

✓ функции обработки данных, указанные в индивидуальном задании;

✓ функцию формирования строки информации об объекте.

3. Создать проект для демонстрации работы: сформировать объекты со значениями-константами и с введенными с клавиатуры значениями полей объекта.

В основной ветке программы создайте три объекта класса. Вывести результаты работы на экран.

Варианты заданий приведены в таблице 1.

Таблица 1. Варианты заданий к лабораторной работе № 1

№ варианта	Класс-родитель и его поля	Функция-метод 1 обработки данных	Функция-метод 2 обработки данных
1	Дата (три числа): день, месяц, год	Определить, является ли год високосным (кратным 4)	Увеличить дату на 5 дней
2	Работник: фамилия, оклад, год поступления на работу	Вычислить стаж работы работника на данном предприятии	Сколько дней прошло после года поступления на работу
3	Книга: название, количество страниц, цена	Вычислить среднюю стоимость одной страницы	Увеличить цену книги в два раза, если название начинается со слова «Программирование»

№ варианта	Класс-родитель и его поля	Функция-метод 1 обработки данных	Функция-метод 2 обработки данных
4	Время (три числа): часы, минуты, секунды	Вычислить количество полных минут в указанном времени	Уменьшить время на 10 минут
5	Товар: наименование, цена, год выпуска	Определить, сколько лет назад был выпущен товар	Увеличить цену товара на 20%, если в наименовании товара есть слово «TV».
6	Дата (три числа): день, месяц, год	Увеличить год на 1	Уменьшить дату на 2 дня
7	Книга: название, автор, год издания	Вычислить, сколько лет книге	Количество дней, прошедших после года издания книги
8	Работник: фамилия, оклад, дата рождения	Вычислить возраст работника	Сколько календарных дней до исполнения работнику 50 лет
	Время (три числа): часы, минуты, секунды	Вычислить количество секунд в указанном времени	Увеличить время на 5 секунд
10	Четыре целых числа: a, b, c, d	Вычислить среднее арифметическое чисел	Определить максимальное из чисел
11	Работник: фамилия, должность, оклад	Увеличить оклад на 15% (каждому работнику)	Работникам, у которых фамилия начинается с сочетания букв

№ варианта	Класс-родитель и его поля	Функция-метод 1 обработки данных	Функция-метод 2 обработки данных
			«Иван», присвоить должность «инженер».
12	Книга: название, количество страниц, цена	Увеличить количество страниц на 10	Уменьшить цену в два раза, если количество страниц больше 100 (после увеличения)
13	Дата (три числа): день, месяц, год	Определить, совпадают ли номер месяца и число дня	Увеличить дату на один месяц
14	Товар: наименование, цена в рублях, изготовитель	Пересчитать цену товара в евро	Увеличить цену товара в евро, если название товара содержит слово «Samsung».
15	Время (три числа): часы, минуты, секунды	Определить количество минут до полуночи (24:00:00)	Увеличить время 100 минут
16	Правильная дробь: числитель, знаменатель	Выразить значение дроби в процентах	Найти сумму цифр значения знаменателя
17	Комната: длина, ширина, высота (в метрах)	Площадь стен (вместе с окнами и дверьми)	Площадь стен без окна (размер 2×15 м) и двери (размер 2 ×8 м).
18	Комплексное число:	Вычислить модуль	Вычислить аргумент

№ варианта	Класс-родитель и его поля	Функция-метод 1 обработки данных	Функция-метод 2 обработки данных
	действительная (a1) и мнимая (b1) части числа	комплексного числа	комплексного числа в градусах
19	Координаты изображения прямоугольника: x1, y1, x2, y2	Вычислить площадь прямоугольника в пикселях	Вычислить длину диагонали прямоугольника в пикселях
20	Параллелепипед: длины сторон	Вычислить площадь поверхности	Вычислить объем параллелепипеда

Лабораторная работа №2. Наследование и полиморфизм

1. На основании предложенной предметной области спроектировать 3-4 класса, используя механизм наследования. Для каждого класса использовать отдельный модуль.
2. Предусмотреть у класса наличие полей, методов и свойств. Названия членов класса должны быть осмысленны и снабжены комментариями.
3. Один из наследников должен перегружать метод родителя.
4. Один из классов должен содержать виртуальный метод, который переопределяется в одном наследнике и не переопределяется в другом.
5. Продемонстрировать работу всех объявленных методов.
6. Продемонстрировать вызов конструктора родительского класса при наследовании.

Варианты заданий приведены в таблице 2.

Таблица 2. Варианты заданий к лабораторной работе № 2

Вариант 1*	Написать программу, в которой описана иерархия классов: средство передвижения (велосипед, автомобиль, грузовик). Базовый класс должен иметь поля для хранения средней скорости, названия модели, числа пассажиров, а также методы получения потребления топлива для данного расстояния и вычисления времени движения на заданное расстояние. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа объекта для демонстрации.
Вариант 2	Написать программу, в которой описана иерархия классов: человек («дошкольник», «школьник», «студент», «работающий»). Базовый класс должен иметь поля для хранения ФИО, возраста, пола, а также методы получения среднего дохода и среднего расхода в денежном эквиваленте.

	Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа объекта для демонстрации.
Вариант 3	Написать программу, в которой описана иерархия классов: геометрические фигуры (круг, прямоугольник, треугольник). Реализовать методы вычисления площади и периметра фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.
Вариант 4	Написать программу, в которой описана иерархия классов: геометрические фигуры (эллипс, квадрат, трапеция). Реализовать методы вычисления площади и периметра фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.
Вариант 5	Написать программу, в которой описана иерархия классов: геометрические фигуры (ромб, параллелепипед, эллипс). Реализовать методы вычисления площади и периметра фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.
Вариант 6	Написать программу, в которой описана иерархия классов: геометрические фигуры (куб, цилиндр, тетраэдр). Реализовать методы вычисления объема и площади поверхности. Реализовать методы вычисления объема и площади поверхности фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.
Вариант 7	Написать программу, в которой описана иерархия классов:

	<p>геометрические фигуры (куб, конус, тетраэдр). Реализовать методы вычисления объема и площади поверхности фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.</p>
Вариант 8*	<p>Написать программу, в которой описана иерархия классов: геометрические фигуры (ромб, прямоугольник, эллипс). Реализовать методы вычисления площади и периметра фигуры. Продемонстрировать работу всех методов классов, предоставив пользователю выбор типа фигуры для демонстрации.</p>
Вариант 9	<p>Написать программу, в которой описана иерархия классов: функция от одной переменной (синус, косинус, тангенс). Базовый класс должен иметь методы получения значения функции для данного значения переменной, а также создания экземпляра класса, представляющего собой производную текущего экземпляра. Продемонстрировать работу всех методов классов всех классов.</p>
Вариант 10*	<p>Написать программу, в которой описана иерархия классов: функция от одной переменной (секанс, косеканс, котангенс). Базовый класс должен иметь методы получения значения функции для данного значения переменной, а также создания экземпляра класса, представляющего собой производную текущего экземпляра. Продемонстрировать работу всех методов классов всех классов.</p>
Вариант 11*	<p>Написать программу, в которой описана иерархия классов: функция от одной переменной (арксинус, арккосинус, а также класс, необходимый для представления производных). Базовый класс должен иметь методы получения значения</p>

	<p>функции для данного значения переменной, а также создания экземпляра класса, представляющего собой производную текущего экземпляра. Продемонстрировать работу всех методов классов всех классов.</p>
Вариант 12*	<p>Написать программу, в которой описана иерархия классов: функция от одной переменной (арктангенс, арккотангенс, а также класс, необходимый для представления производных). Базовый класс должен иметь методы получения значения функции для данного значения переменной, а также создания экземпляра класса, представляющего собой производную текущего экземпляра. Продемонстрировать работу всех методов классов всех классов.</p>
Вариант 13*	<p>Написать программу, в которой описана иерархия классов: функция от одной переменной (логарифм, натуральный логарифм, а также класс, необходимый для представления производных). Базовый класс должен иметь методы получения значения функции для данного значения переменной, а также создания экземпляра класса, представляющего собой производную текущего экземпляра. Продемонстрировать работу всех методов классов всех классов.</p>
Вариант 14**	<p>Написать программу, в которой описана иерархия классов: функция от одной переменной (экспонента, гиперболический синус, гиперболический косинус). Базовый класс должен иметь методы получения значения функции для данного значения переменной, а также создания экземпляра класса, представляющего собой производную текущего экземпляра. Продемонстрировать работу всех методов классов всех классов.</p>

Лабораторная работа №3. Работа с базами данных

1. Представьте таблицы (согласно вашему варианту) в виде структур языка Python

2. Реализуйте в консоли интерфейс по добавлению, удалению, изменению данных. Имейте в виду, что связанные операции (удаление, добавление, изменение) для связанных таблиц, должны изменять данных во всех связанных структурах.

3. Реализуйте функционал по сохранению данных в файлы формата .csv и считыванию информации из файлов

Варианты задания приведены в таблице 3.

Таблица 3. Варианты заданий к лабораторной работе №3

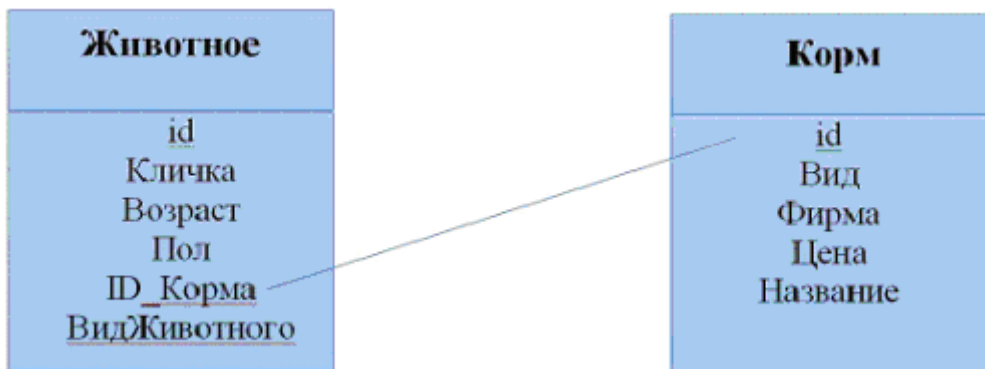
№ варианта	Выведите следующую информацию в консоль построчно:	Посчитайте и выведите результат:
1	Для каждого преподавателя: «ФИО преподавателя», «название кафедры», «должность преподавателя».	Для каждой кафедры: сколько всего преподавателей.
2	Для каждого животного: «Кличка животного», «пол», «возраст», «название корма», «цена корма».	Для каждого корма: сколько животных им питается.
3	Для каждого ребенка: «ФИО ребенка», «ФИО отца», «ФИО матери», «ФИО врача», «возраст ребенка».	Для каждого врача: сколько детей он лечит.
4	Для каждого товара: «No товара», «название товара», «стоимость товара», «цвет	Для каждого цвета: количество товаров.

	товара».	
5	Для каждого контента: «Название контента», «название меню», «ник автора», «аннотация».	Для каждого пользователя: количество контента, которое он добавил.
6	Для каждого человека: «ФИО персоны», его должность, в какой фирме работает.	Для каждой фирмы: сколько в ней сотрудников.
7	Для каждого преподавателя: «ФИО преподавателя», список дисциплин (которые он ведет).	Для каждого преподавателя: сколько дисциплин он ведет.
8	Для каждой дисциплины: «Название дисциплины», в какой день недели ведется, на какой паре.	Для каждого дня недели: сколько всего пар предусмотрено в этот день.
9	Для каждой дисциплины: «Название дисциплины», «количество лекций», «количество практик», «форма контроля», «название кафедры» (которая читает дисциплину).	Для каждой кафедры: количество дисциплин, которое ведется сотрудниками данной кафедры.
10	Для каждой страны: «Название страны», список городов.	Для каждого города: сколько в нем улиц.

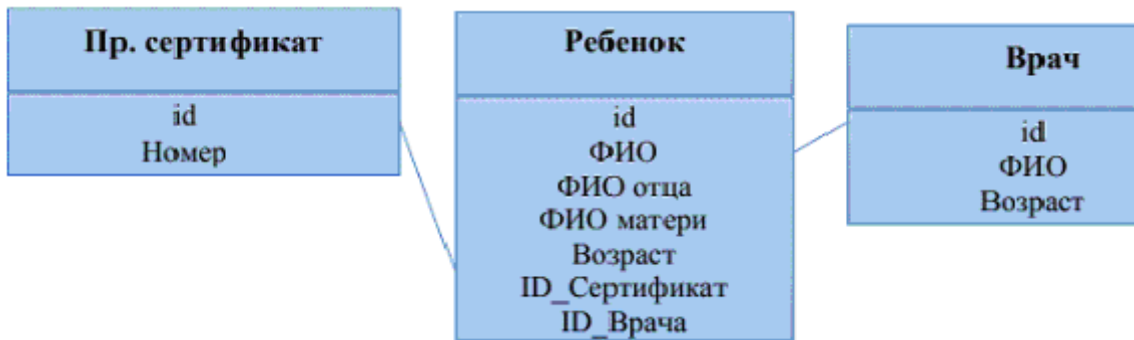
Вариант №1



Вариант №2



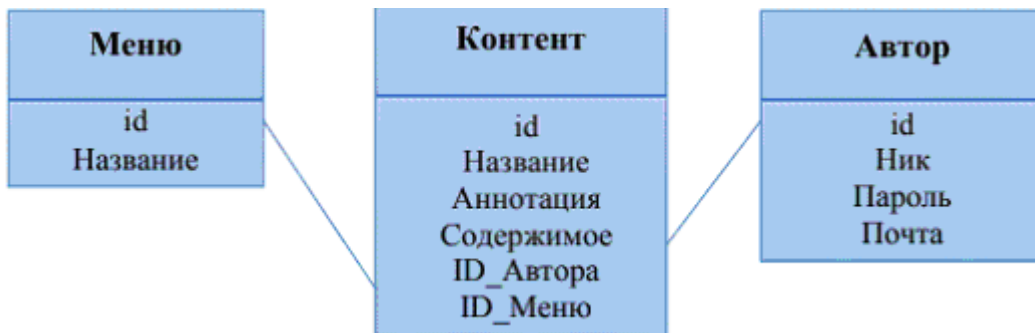
Вариант №3



Вариант №4



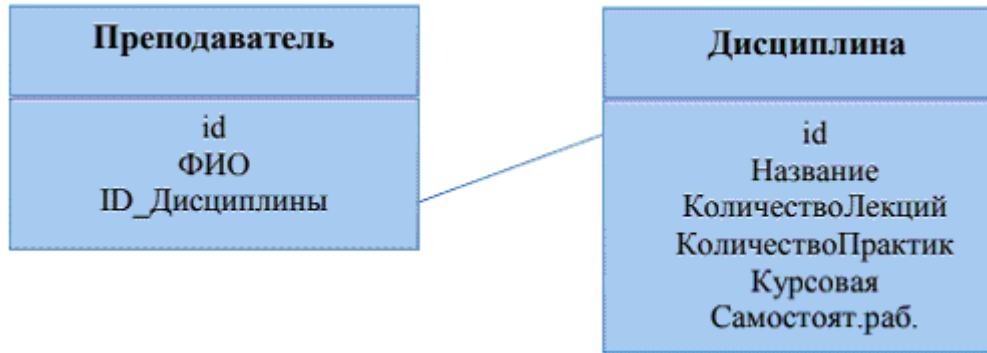
Вариант №5



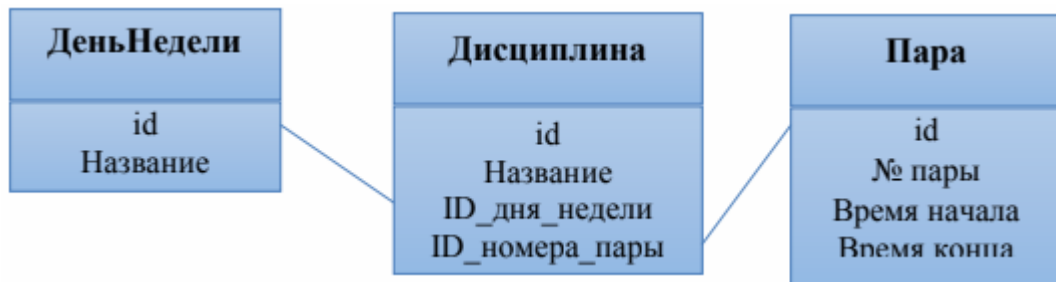
Вариант №6



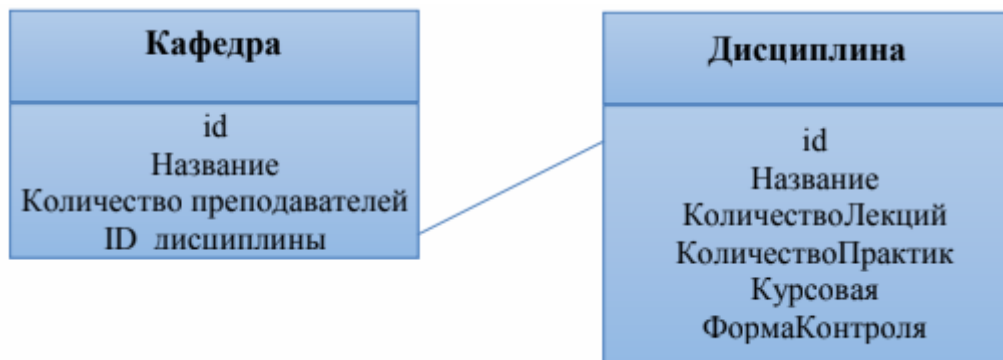
Вариант №7



Вариант №8



Вариант №9



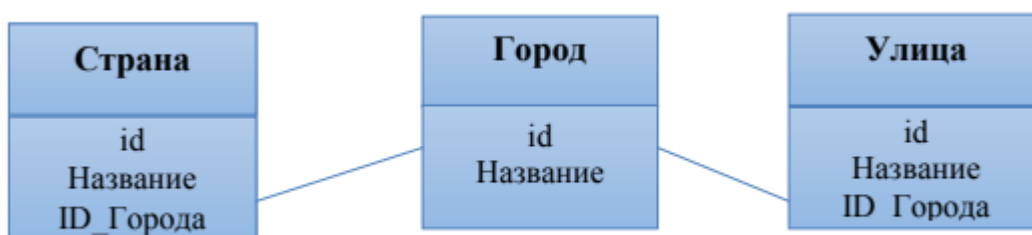
Вариант №10



Лабораторная работа №4. Связанные структуры данных

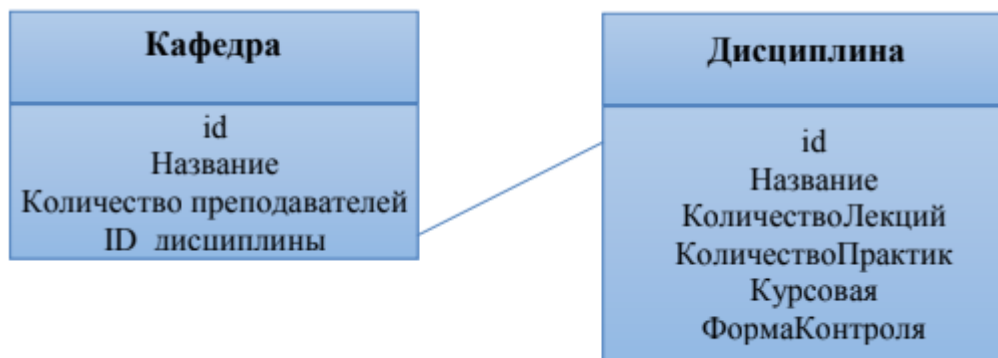
Вариант №1

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



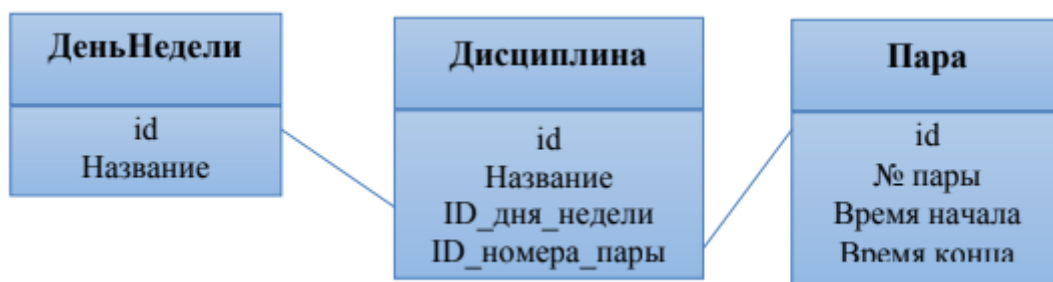
Вариант №2

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



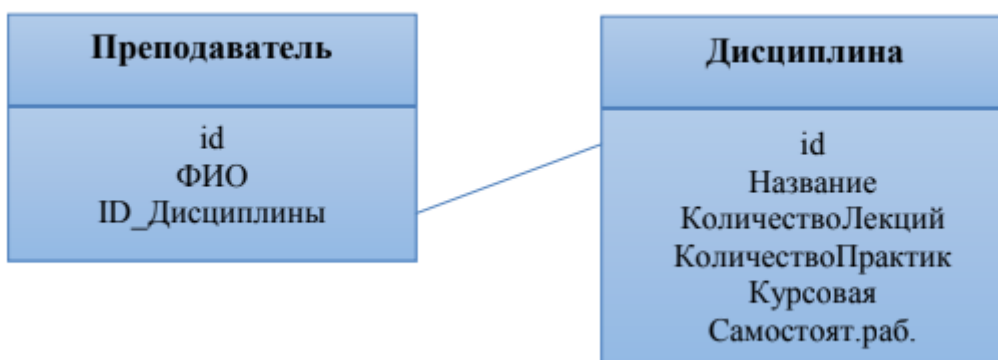
Вариант №3

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



Вариант №4

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

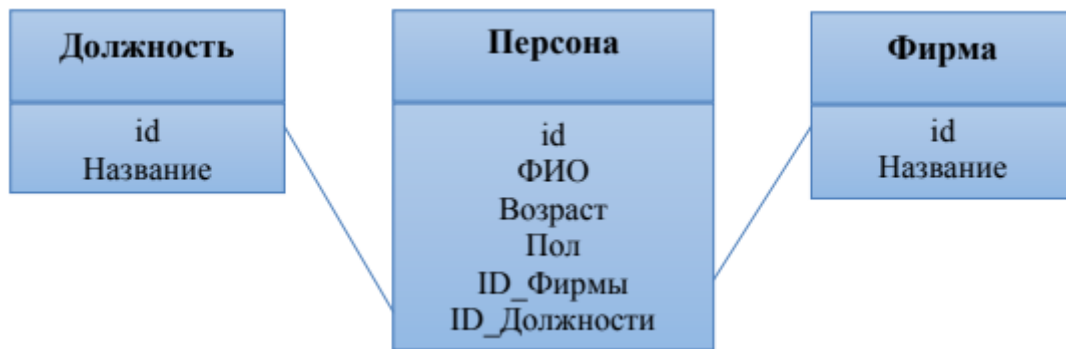


Вариант №5

1. Пусть дана база данных (приведена ниже). Используйте нужные структуры данных для ее хранения. Заполните БД. Выведите список всех должностей для каждой фирмы.

2. Для БД из задания 1 выведите фирму, в которой работает больше всего сотрудников в возрасте от 20 до 30 лет.

3. Для БД из задания 1 выведите список всех сотрудников в должности «Директор» на фирме «НосковИко».

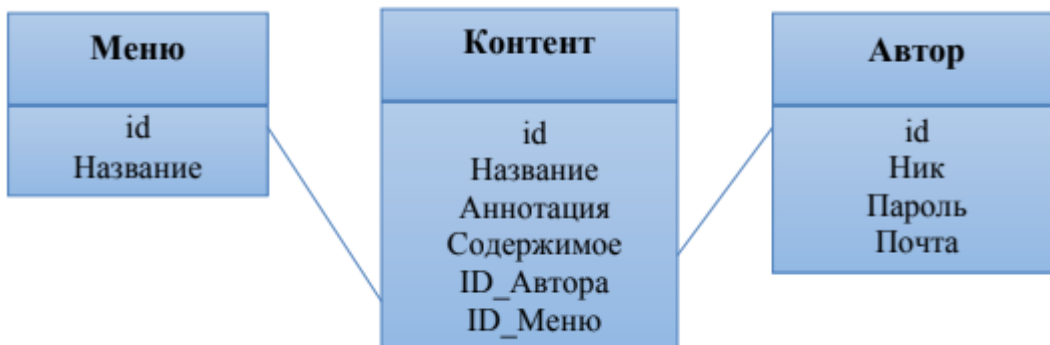


Вариант №6

1. Пусть дана база данных (приведена ниже). Используйте нужные структуры данных для ее хранения. Заполните БД. Выведите список всех меню.

2. Для БД из задания 1 выведите информацию об авторах, кто написал более 2 статей.

3. Для БД из задания 1 выведите количество написанных статей для каждого автора.



Вариант №7

1. Пусть дана база данных (приведена ниже). Используйте нужные структуры данных для ее хранения. Заполните БД. Выведите список товаров красного цвета, которые есть в наличии в магазине.

2. Для БД из задания 1 выведите хотя бы 2 товара одинакового вида, но разного цвета.

3. Для БД из задания 1 выведите виды товаров, для которых имеется товар разных цветов.

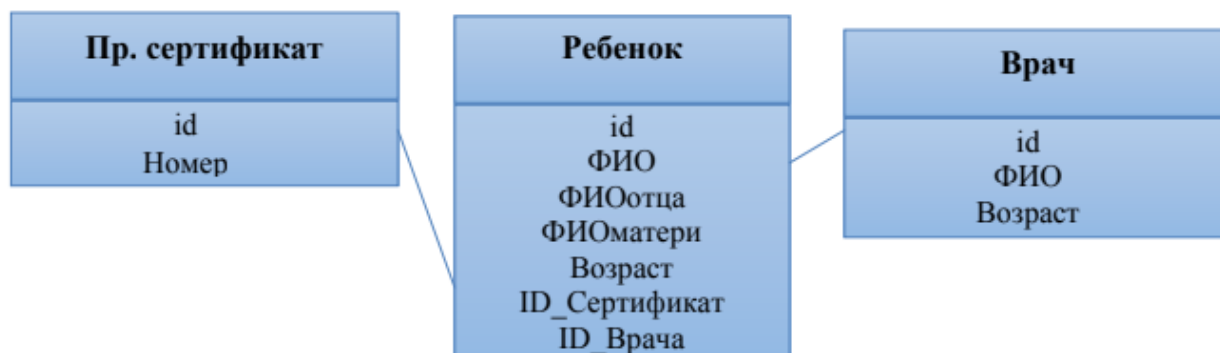


Вариант №8

1. Пусть дана база данных (приведена ниже). Используйте нужные структуры данных для ее хранения. Заполните БД. Выведите список детей в возрасте до 10 лет.

2. Для БД из задания 1 выведите информацию о детях, чьи родители имеют разные фамилии.

3. Для БД из задания 1 выведите список врачей в возрасте от 20 до 60 лет.

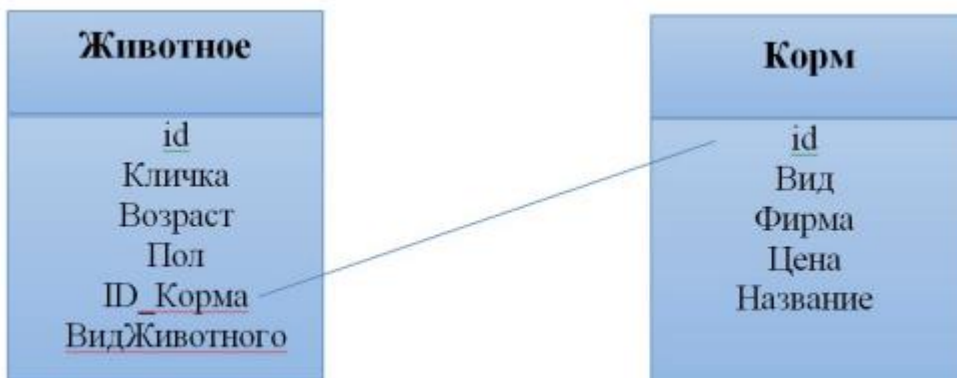


Вариант №9

1. Пусть дана база данных (приведена ниже). Используйте нужные структуры данных для ее хранения. Заполните БД. Выведите животных, чья кличка длиннее 5 символов.

2. Для БД из задания 1 выведите список кормов в порядке убывания их стоимости.

3. Для БД из задания 1 выведите список кормов, которым питается больше всего животных.



Вариант №10

1. Пусть дана база данных (приведена ниже). Используйте нужные структуры данных для ее хранения. Заполните БД. Выведите кафедру, на которой работает больше всего сотрудников.

2. Для БД из задания 1 выведите список кафедр в порядке убывания количества сотрудников.

3. Для БД из задания 1 выведите «самую молодую» кафедру (возраст кафедры = сложить возраст сотрудников и поделить на их количество).



Лабораторная работа №5. Работа с графикой

Разработайте приложение, которое строит график одной из указанных ниже функций. Построение производится в симметричной системе координат (область рисования делится на четыре равных квадранта), маркировка осей производится в зависимости от масштабирования X и Y . График должен полностью помещаться в области рисования. Пользователю можно изменить цвет фона и цвет графика. Коэффициенты и граничные значения x вводятся с клавиатуры.

Указания. Программа должна предусматривать возможность задания значений параметров, диапазона значений по оси OX , при этом диапазон значений по оси OY должен определяться автоматически так, чтобы график полностью помещался в область рисования, также у пользователя должна быть возможность выбора цвета графика, варианта рисования графика: точками или линией (с выбором толщины линии), количества точек, на основании которых должен быть построен график функции (считается, что точки распределяются равномерно по оси OX).

Подсчет значений функции должен быть реализован в виде подпрограммы.

При построении графика необходимо учитывать масштаб, заданный пользователем при построении системы координат.

Пояснения. При построении графиков функций на экране монитора необходимо преобразовывать расчетные координаты в графические с учетом дискретности растровой сетки монитора. Для этого желательно создать процедуры, обеспечивающие универсальность при выводе графических изображений. Ниже приводится алгоритм построения графика функции $Y=F(X)$ в заданной области экрана с возможностью автоматического масштабирования.

Пусть задана непрерывная функция $F(X)$ в диапазоне изменения аргумента $X=[A..B]$. Требуется построить по N точкам график функции $Y=F(X)$ в прямоугольной области экрана left, up, right, down.

а) Определяем массивы значений аргумента и функции.

б) Определяем наибольшее Y_{\max} и наименьшее Y_{\min} значения функции в заданном интервале изменения аргумента. Эти значения необходимо определить для полного размещения графика в расчетной области.

в) Строим систему координат, предварительно задав начало координат X_0, Y_0 .

г) Определяем коэффициент масштабирования при построении графика в заданной области экрана. Данный коэффициент используется для масштабирования по обеим осям, так как при введении разных масштабов по осям, возможно искажение естественной формы кривой (растяжение или сжатие по одной из осей).

д) Определяем координаты точек для построения графика в системе координат экрана. При этом учитываем необходимость смещения координат точек по соответствующим осям относительно границ области left и down., а также необходимость "переворота" оси Y, которая в координатах монитора направлена сверху вниз.

е) Строим график в виде последовательных отрезков, соединяющих две соседние точки (первый отрезок соединяет 1 и 2 точки, второй отрезок соединяет 2 и 3 точки и т.д., в результате чего получается непрерывная линия)

Варианты заданий приведены в таблице 5.

Таблица 5. Варианты заданий к лабораторной работе № 5

№ варианта	Функция
1	$y = a \sin(bx + c) - d \cos(gx^2 + fx + h)$
2	$y = a \sin(bx)$
3	$y = a \cos(bx^2 + cx + d)$
4	$y = ax^3 + bx^2 + cx + d$
5	$y = ax^5 + bx^4 + cx^3 + dx^2 + fx + g$
6	$y = ax^4 + bx^3 + cx^2 + dx + f$
7	$y = (ax^2 + bx + c) \sin(dx)$
8	$y = (ax^3 + bx^2 + cx + d) \sin(x^2)$
9	$y = (ax^4 + bx^3 + cx^2 + dx + f) \cos(ax^2 + bx + c)$
10	$y = e \sin(ax^2 + bx + c) + d \cos(fx + g)$
11	$y = d \sin(ax^2 + bx + c) \exp(-fx)$
12	$y = a \cos(bx) \exp(-cx)$
13	$y = a \exp(bx) + c$

14	$y = a \ln(bx + c) + d * \lg(fx)$
15	$y = a^{bx+c} + f$

Лабораторная работа №6. Сортировки

Реализовать консольное приложение, предназначенное для сортировки целочисленного вектора стандартными средствами и собственной функцией, реализующей алгоритм сортировки согласно варианту. **Входные данные:** количество элементов, тип входного вектора, вектор. **Выходные данные:** отсортированный массив, время работы библиотечной и реализованной функций сортировок, ускорение библиотечной сортировки.

Особенности:

- ✓ должна присутствовать возможность выбора исходного (сортируемого) вектора: вводимый с клавиатуры или генерируемый согласно варианту (два возможных для выбора типа генерируемых данных);
- ✓ в случае, если число элементов вектора менее некоторого разумного значения, исходный и отсортированные двумя методами векторы должны выводиться на экран для сравнения и проверки правильности результатов сортировки;
- ✓ в случае, если число элементов вектора более некоторого разумного значения, должны выводиться на экран для сравнения и проверки правильности результатов сортировки некоторое количество элементов исходного и отсортированных двумя методами векторов.

Варианты заданий представлены в таблице 6.

Таблица 6. Варианты заданий к лабораторной работе № 6

Вариант 1	Сортировка вставками	Строго случайные данные и «почти отсортированные» случайные данные
Вариант 2	Сортировка выбором	Строго случайные данные и «отсортированные наоборот» случайные данные
Вариант 3	Сортировка пузырьком	Строго случайные данные и случайные данные с малым числом уникальных значений
Вариант 4	Сортировка Шелла	Строго случайные данные и «почти отсортированные» случайные данные
Вариант 5	Сортировка слиянием	Строго случайные данные и «отсортированные наоборот» случайные данные
Вариант 6	Пирамидальная сортировка	Строго случайные данные и случайные данные с малым числом уникальных значений
Вариант 7	Быстрая сортировка	Строго случайные данные и «почти отсортированные» случайные данные
Вариант 8	Быстрая сортировка с разделением на 3 части	Строго случайные данные и «отсортированные наоборот» случайные данные
Вариант 9	Сортировка вставками	Строго случайные данные и случайные данные с малым числом уникальных значений
Вариант 10	Сортировка выбором	Строго случайные данные и «почти отсортированные» случайные данные
Вариант 11	Сортировка	Строго случайные данные и

	пузырьком	«отсортированные наоборот» случайные данные
Вариант 12	Сортировка Шелла	Строго случайные данные и случайные данные с малым числом уникальных значений
Вариант 13	Сортировка слиянием	Строго случайные данные и «почти отсортированные» случайные данные
Вариант 14	Пирамидальная сортировка	Строго случайные данные и «отсортированные наоборот» случайные данные
Вариант 15	Быстрая сортировка	Строго случайные данные и случайные данные с малым числом уникальных значений
Вариант 16	Быстрая сортировка с разделением на 3 части	Строго случайные данные и «почти отсортированные» случайные данные

Лабораторная работа №7. Создание многопоточных приложений

1. Создать массив заполнен 100000 случайными элементами, целые числа отображенного графическим образом. Определите базовый класс, содержащий, в частности, поле – одномерный массив и методы отображения массива на экране, заполнения массива (разными способами) и абстрактный метод сортировки массива. Определите три класса-наследника, в которых реализованы следующие методы сортировки (по одному – в каждом): один из простых методов (вставки, выбора, обмена), быстрая сортировка и метод, описанный в Вашем варианте

2. Создайте приложение, которое нужно разработать в данной лабораторной работе, должно не только демонстрировать все три процесса сортировки в «реальном времени», но и делать это одновременно.

3. В приложении должна быть возможность изменять приоритеты потоков (чтобы влиять на скорость сортировки). Разумеется, каждый поток будет работать со своей копией массива. Но было бы желательно показать, что произойдет, если все три потока будут работать с одним и тем же массивом (по очереди). Варианты методов сортировки необходимо взять из лабораторной работы №6.

Лабораторная работа №8. Линейная регрессия

Ознакомьтесь с содержимым папки с заданием, которая расположена в локальной сети университета (I:\tutorial\teachers\Tatyana.P.Rubtsova\Phyton\) и включает в себя файлы, представленные ниже.

main_one.py – «основной» модуль, необходимый для выполнения первой части задания, который поможет выполнить его поэтапно. Настоящий программный код не требует какой-либо коррекции!

main_multi.py – «основной» модуль, необходимый для выполнения второй части задания, который поможет выполнить его поэтапно. Настоящий программный код не требует какой-либо коррекции!

data1.txt – база данных для выполнения первой части задания.

data2.txt – база данных для выполнения второй части задания. **plotData.py** – модуль, содержащий функцию `plotData`, которая необходима для визуализации данных.

computeCost.py – модуль, содержащий функцию `computeCost`, которая необходима для вычисления значения стоимостной функции линейной регрессии.

gradientDescent.py – модуль, содержащий функцию `gradientDescent`, которая необходима для выполнения градиентного спуска с целью поиска параметров модели линейной регрессии.

featureNormalize.py – модуль, содержащий функцию `featureNormalize`, которая необходима для нормализации признаков.

normalEqn.py – модуль, содержащий функцию `normalEqn`, которая необходима для поиска параметров модели линейной регрессии с использованием нормальных уравнений.

При выполнении данного задания требуется заполнить пустые места программного кода в блоках с комментарием «Ваш код здесь». Данную процедуру необходимо выполнить для следующих функций: `plotData`, `computeCost`, `gradientDescent`.

При решении любой задачи с использованием инструментов машинного обучения важным является понимание структуры анализируемых данных и их визуализация в случае возможности. В первой части задания предлагается использовать базу данных из файла **data1.txt**. Данные представляют собой множество объектов, описываемых одним признаком (численность населения в некотором городе) и меткой (прибыль, которую можно получить при продаже определенного товара в городе с соответствующей численностью населения). Завершите программный код в модуле **plotData.py**, который позволит выполнять визуализацию данных. Завершение модуля подразумевает под собой написание строчек программного кода, которые позволят вызвать функцию из соответствующего модуля в файле **main_one.py**, позволяя решить определенный кусок настоящего задания. Например, в данном случае заверченный программный код будет выглядеть так, как представлено на рис. 1.

После завершения каждого блока кода интерпретируйте файл **main_one.py** с целью проверки правильности работы соответствующей части задания. Результат визуализации данных с использованием функции `plotData` представлен на рис. 2. В случае успешной интерпретации программного кода разрешается перейти к следующему пункту задания.

Завершите программный код в модуле **computeCost.py**, который позволит вычислить значение стоимостной функции для линейной регрессии. При выполнении данной части задания могут понадобиться функции из библиотеки **NumPy**, представленные ниже.

`dot` – позволяет вычислить матричное произведение для двумерных массивов и скалярное произведение для одномерных массивов (без комплексного сопряжения).

`sum` – позволяет вычислить сумму элементов вдоль определенной размерности двумерного массива и сумму всех элементов для одномерного массива.

```

import matplotlib.pyplot as plt
import numpy as np

def plotData(data):
    """
    Функция позволяет выполнить визуализацию данных в декартовой
    системе координат с подписанным осями (численность населения
    и прибыль)
    """

    # ===== Ваш код здесь =====
    # Инструкция: визуализируйте данные с использованием функций
    # figure и plot. Подпишите оси с использованием функций xlabel
    # и ylabel, предполагая, что аргументами этих функций являются
    # численность населения по x и прибыль по y

    plt.figure()
    plt.plot(data[:, 0], data[:, 1], 'rx', markersize = 5, label = 'Тренировочные данные')
    plt.legend(loc = 'upper right', shadow = True, fontsize = 12, numpoints = 1)
    plt.xlabel('Численность населения в 10,000')
    plt.ylabel('Прибыль в $10,000')
    plt.grid()

    # =====

```

Рис. 1. Завершенный программный код для функции plotData

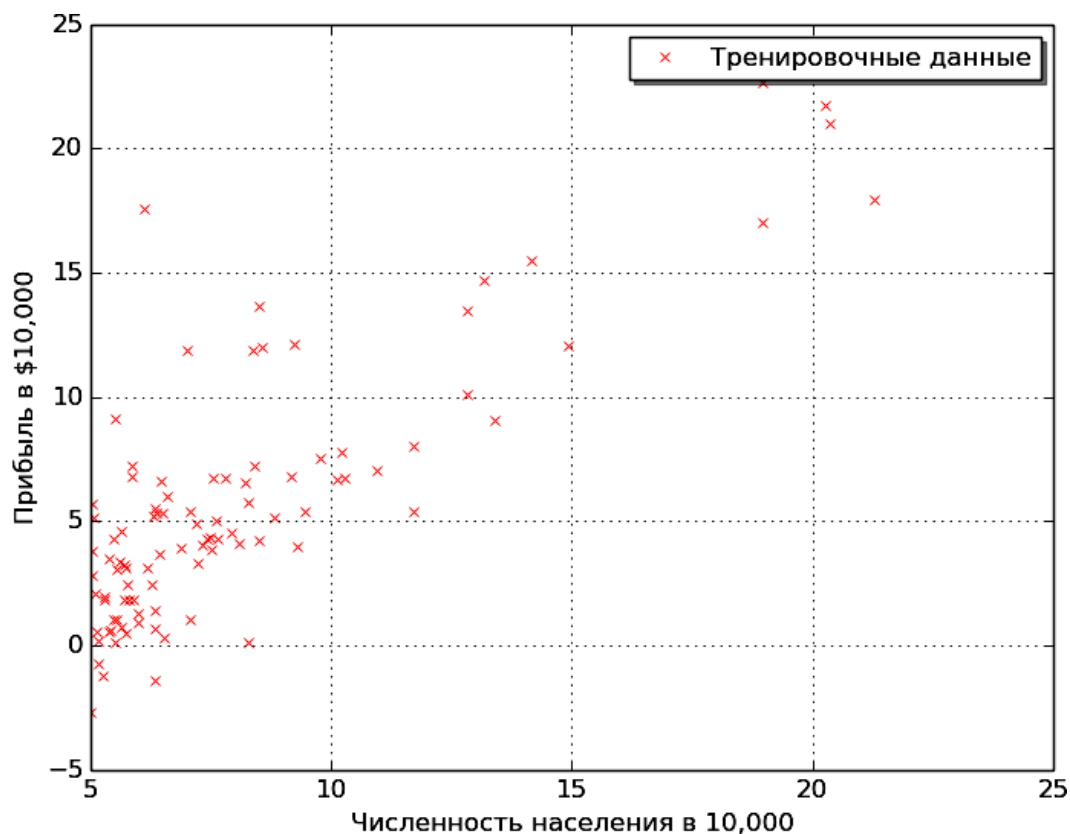


Рис. 2. Результат визуализации тренировочных данных

Завершите программный код в модуле **gradientDescent.py**, который позволит выполнить алгоритм градиентного спуска с целью обучения параметров модели линейной регрессии. При выполнении данной части задания могут понадобиться следующие функции из библиотеки **NumPy**: `dot` и `transpose`.

`transpose` – позволяет выполнить транспонирование массива. Для одномерного массива данная функция не оказывает никакого действия, а для двумерного массива использование функции соответствует обычному матричному транспонированию.

После обучения параметров модели линейной регрессии с одной переменной с настройками градиентного спуска, заданными по умолчанию в файле `main_one.py` должен получиться результат, представленный на рис. 3.

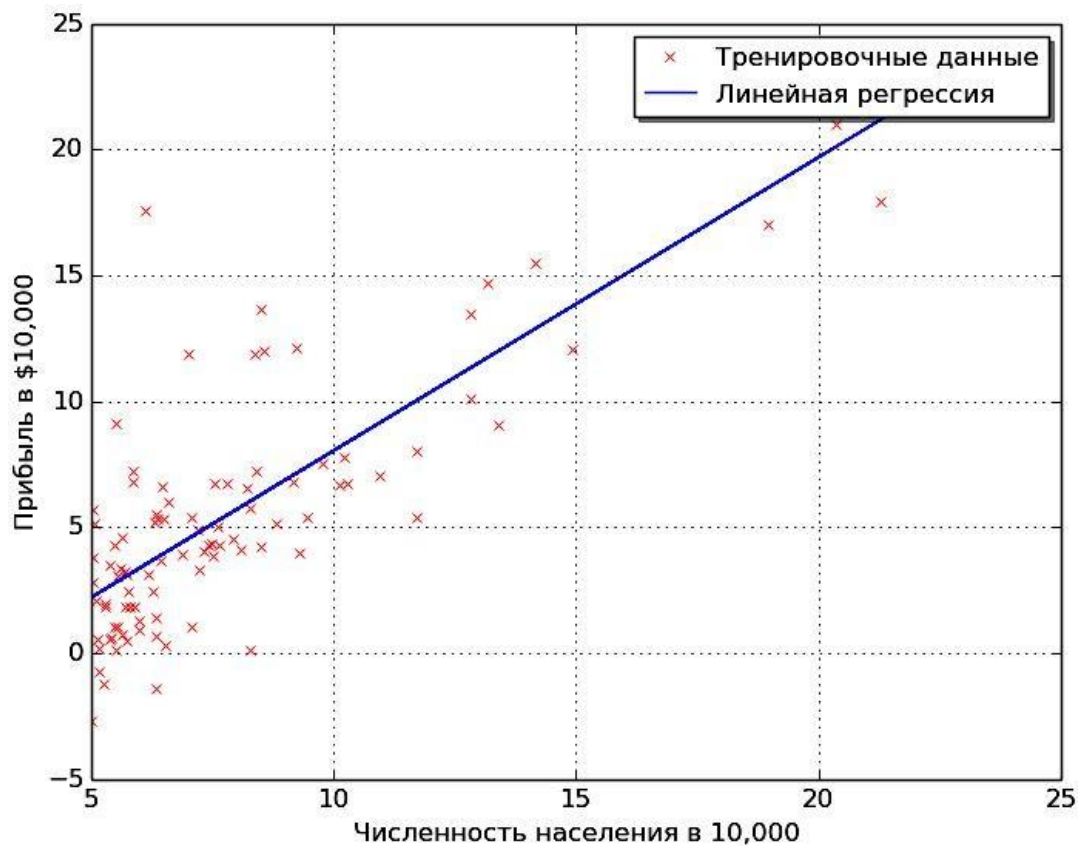


Рис. 3. Результат визуализации тренировочных данных и линии регрессии (гипотезы для линейной регрессии с одной переменной)

После завершения предыдущих пунктов выполните предсказание прибыли от продажи товара в городах с численностью населения 35,000 и 70,000. При выполнении задания обратите внимание на то, что в матрице объекты-признаки, сформированной в файле `main_one.py` после загрузки базы данных из `data1.txt`, единственный признак объекта, описывающий численность населения в городе, является нормированным на значение 10,000.

Линейная регрессия со множеством переменных

При выполнении данного задания требуется заполнить пустые места программного кода в блоках с комментарием «Ваш код здесь». Данную процедуру необходимо выполнить для следующих функций: `featureNormalize`, `normalEqn`.

1. Во второй части задания предлагается использовать базу данных из файла **data2.txt**. В этом случае данные представляют собой множество объектов, описываемых двумя признаками (площадь помещения в квадратных футах и число комнат в нем) и меткой (стоимость жилья для заданной площади и числа комнат). Завершите программный код в модуле **featureNormalize.py**, который позволит выполнить нормализацию признаков на их математическое ожидание и среднеквадратическое отклонение. При выполнении данной части задания могут понадобиться функции из библиотеки **NumPy**, представленные ниже.

`mean` – позволяет вычислить арифметическое среднее вдоль определенной размерности.

`std` – позволяет вычислить среднеквадратическое отклонение вдоль определенной размерности. При вызове функции в настоящем задании формальному параметру `ddof` следует присвоить значение 1. Последнее требуется для получения несмещенной оценки среднеквадратического отклонения.

`divide` – позволяет выполнить поэлементное деление одного массива на другой.

`repeat` – выполняет повторение массивов размерности 0, 1 и 2 вдоль определенной размерности.

2. С использованием ранее завершенных функций `computeCost`, `gradientDescent` выполните обучение параметров модели линейной регрессии со множеством переменных. Проведите небольшое исследование влияния параметра сходимости и числа итераций на качество сходимости градиентного спуска. Исследование можно выполнить, используя визуализацию изменения значения стоимостной функции в зависимости от числа итераций при фиксированном параметре сходимости (рис. 4).

3. После завершения предыдущих пунктов выполните предсказание стоимости жилья для площади 1650 квадратных футов и числа комнат 3.

4. Обратите внимание на то, что перед выполнением процедуры предсказания

требуется провести нормализацию признаков на соответствующие им математическое ожидание и среднее квадратическое отклонение.

5. Завершите программный код в модуле **normalEqn.py**, который позволит выполнить поиск параметров модели линейной регрессии с использованием нормальных уравнений. При выполнении данной части задания могут понадобиться следующие функции из библиотеки **NumPy**: `dot`, `transpose` и `inv`.

`inv` – позволяет выполнить вычисление обратной матрицы.

Используя файл **main_one.py** ответьте на следующие вопросы по первой части практического задания.

1. Чему равно значение стоимостной функции для случая, когда все параметры модели равны нулю?

2. Чему равны значения параметров обученной модели линейной регрессии с одной переменной для случая, когда параметр сходимости равен 0.01, а число итераций градиентного спуска равно 1500 ?

3. Чему равна прибыль от продажи товара в городах с численностью населения 35,000 и 70,000 для обученной в вопросе 2 модели?

Используя файл **main_multi.py** ответьте на следующие вопросы по первой части практического задания.

1. Чему равны значения параметров обученной модели линейной регрессии со множеством переменных для случая, когда параметр сходимости равен 0.01, а число итераций градиентного спуска равно 400?

2. Чему равна стоимость жилья для площади 1650 квадратных футов и числа комнат 3 для обученной в вопросе 1 модели?

3. Чему равны значения параметров обученной модели линейной регрессии со множеством переменных с использованием нормальных уравнений?

4. Чему равна стоимость жилья для площади 1650 квадратных футов и числа комнат 3 для обученной в вопросе 3 модели?

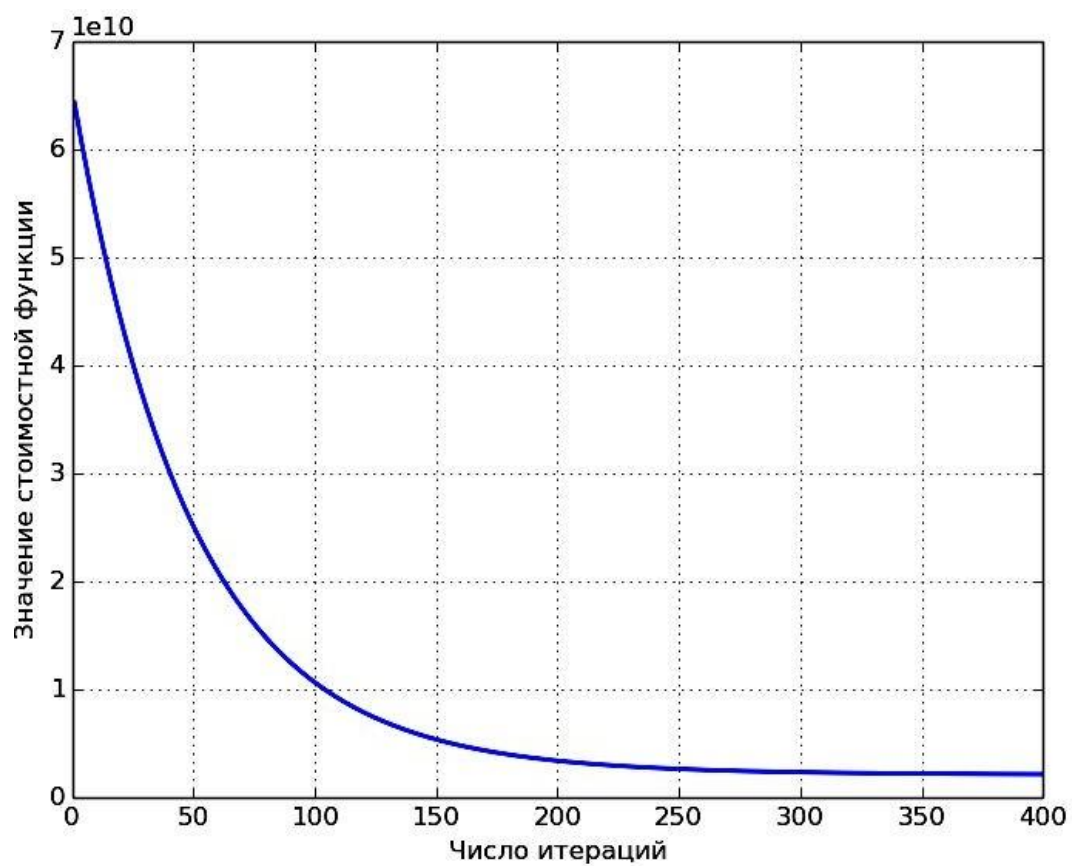


Рис. 4. Пример сходимости градиентного спуска для удачно подобранной скорости сходимости

Лабораторная работа №9. Логистическая регрессия

Ознакомиться с содержимым папки с заданием, которая расположена в локальной сети университета (I:\tutorial\teachers\Tatyana.P.Rubtsova\Phyton\)) и включает в себя файлы, представленные ниже.

main.py – «основной» модуль, необходимый для выполнения задания, который поможет выполнить его поэтапно. Настоящий программный код не требует какой-либо коррекции!

data.txt – база данных для выполнения задания.

plotData.py – модуль, содержащий функцию `plotData`, которая необходима для визуализации данных. **plotDecisionBoundary.py** – модуль, содержащий функцию `plotDecisionBoundary`, которая необходима для визуализации данных с границей решения для заданного множества параметров модели логистической регрессии. Данный модуль не требует коррекции!

computeCost.py – модуль, содержащий функцию `computeCost`, которая необходима для вычисления значения стоимостной функции логистической регрессии.

gradientDescent.py – модуль, содержащий функцию `gradientDescent`, которая необходима для выполнения градиентного спуска с целью поиска параметров модели логистической регрессии.

featureNormalize.py – модуль, содержащий функцию `featureNormalize`, которая необходима для нормализации признаков. Данный модуль не требует коррекции!

sigmoid.py – модуль, содержащий функцию `sigmoid`, которая позволяет вычислить значение сигмоидной функции. **predict.py** – модуль, содержащий функцию `predict`, которая необходима для предсказания метки класса (0 или 1).

При выполнении задания требуется заполнить пустые места программного кода в блоках с комментарием «Ваш код здесь». Данную процедуру необходимо выполнить для следующих функций: `plotData`, `computeCost`, `gradientDescent`, `sigmoid`, `predict`.

1. При решении любой задачи с использованием инструментов машинного обучения важным является понимание структуры анализируемых данных и их визуализация в случае возможности.

В настоящем задании предлагается использовать базу данных из файла **data.txt**. Данные представляют собой множество объектов, описываемых двумя признаками (оценка студента за первый экзамен и оценка студента за второй экзамен) и меткой (аттестован или не аттестован студент по итогам двух экзаменов). Необходимо обратить внимание на то, что база данных в настоящем задании размечена, а метка принимает дискретный набор из двух значений (0 – не аттестован, 1 – аттестован). Поэтому в рамках настоящего задания рассматривается решение задачи бинарной классификации, а не регрессии, как это было

Завершите программный код в модуле **plotData.py**, который позволит выполнять визуализацию данных.

Завершение модуля подразумевает под собой написание строчек программного кода, которые позволят вызвать функцию из соответствующего модуля в файле **main.py**, позволяя решить определенный кусок настоящего задания. Например, в данном случае заверченный программный код будет выглядеть так, как представлено на рис. 5.

```
import matplotlib.pyplot as plt
import numpy as np

def plotData(X, y):
    """
    Функция позволяет выполнить визуализацию данных с маркером
    + для положительных примеров и маркером o для отрицательных
    примеров. X - матрица объекты-признаки размера mx2,
    а y - вектор меток размера mx1, где m - размер базы данных
    """

    # ===== Ваш код здесь =====
    # Инструкция: визуализируйте положительные и отрицательные
    # примеры на двумерной плоскости, используя маркер + для
    # обозначения положительных примеров и маркер o для обозначения
    # отрицательных примеров

    plt.figure()
    pos = np.where(y == 1)[0]
    neg = np.where(y == 0)[0]

    plt.plot(X[pos, 0], X[pos, 1], '+', markersize = 7, markeredgecolor = 'black', markeredgewidth = 2)
    plt.plot(X[neg, 0], X[neg, 1], 'o', markersize = 7, markeredgecolor = 'black', markerfacecolor = 'yellow')
    plt.grid()

    # создание нового окна
    # извлечение индексов положительных примеров
    # извлечение индексов отрицательных примеров
    # построение положительных примеров
    # построение отрицательных примеров
    # создание координатной сетки
```

Рис. 5. Заверщенный программный код для функции plotData

После завершения каждого блока кода интерпретируйте файл **main.py** с целью проверки правильности работы соответствующей части задания. Результат визуализации данных с использованием функции `plotData` представлен на рис. 6. В случае успешной интерпретации программного кода разрешается перейти к следующему пункту задания.

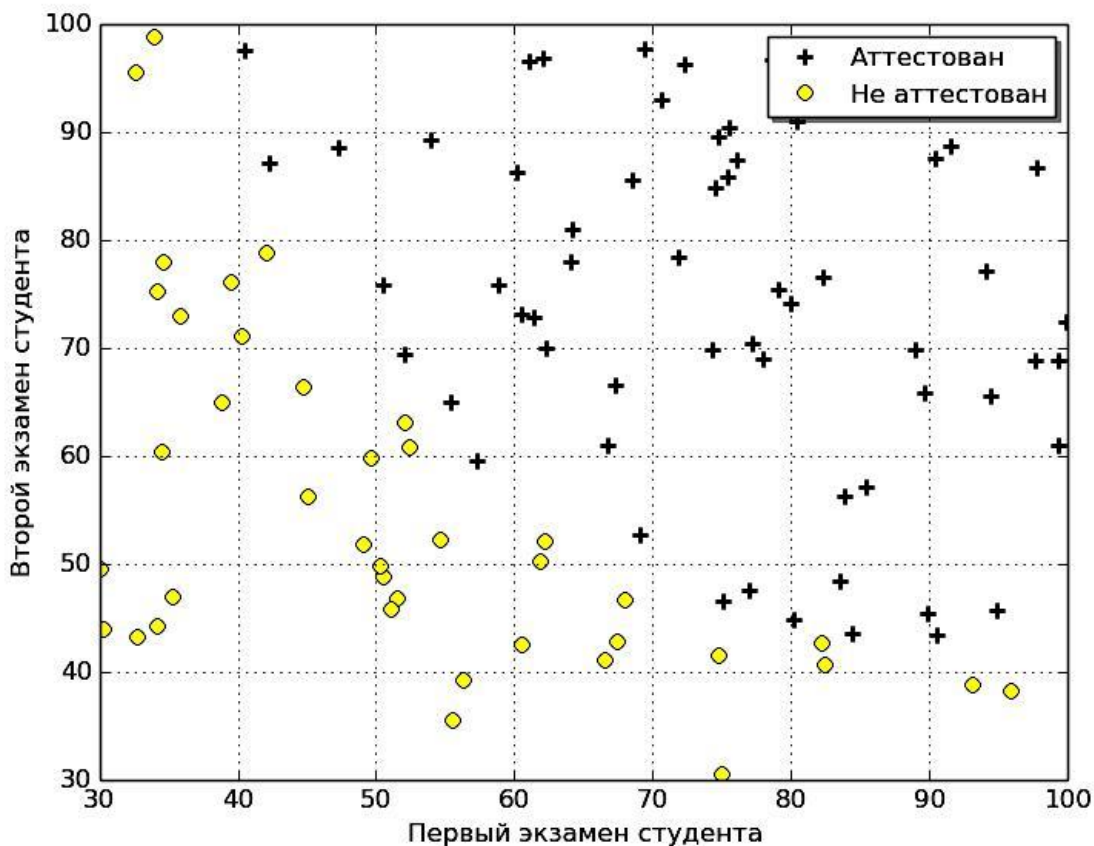


Рис. 6. Результат визуализации тренировочных данных

2. Завершите программный код в модуле **computeCost.py**, который позволит вычислить значение стоимостной функции для логистической регрессии. При выполнении данной части задания могут понадобиться функции из библиотеки **NumPy**, представленные ниже.

`dot` – позволяет вычислить матричное произведение для двумерных массивов и скалярное произведение для одномерных массивов (без комплексного сопряжения).

`sum` – позволяет вычислить сумму элементов вдоль определенной размерности двумерного массива и сумму всех элементов для одномерного массива.

`log` – позволяет вычислить натуральный логарифм от элементов массива.

При заполнении программного кода в модуле **computeCost.py** потребуется вычисление значений сигмоидной функции. Реализуйте ее вычисление в модуле **sigmoid.py**. При реализации сигмоидной функции может понадобиться функция `exp` из библиотеки **NumPy**, которая позволяет вычислить значения экспоненциальной функции от элементов массива.

3. Завершите программный код в модуле **gradientDescent.py**, который позволит выполнить алгоритм градиентного спуска с целью обучения параметров модели логистической регрессии. При выполнении данной части задания могут понадобиться следующие функции из библиотеки **NumPy**: `dot` и `transpose`.

`transpose` – позволяет выполнить транспонирование массива. Для одномерного массива данная функция не оказывает никакого действия, а для двумерного массива использование функции соответствует обычному матричному транспонированию.

Так же совершенно будет необходима функция `sigmoid`, реализованная в модуле **sigmoid.py**.

Обратите внимание на то, что при обучении параметров модели логистической регрессии в файле **main.py** используется нормализация признаков, позволяющая выполнить качественную сходимость градиентного спуска к единственному в данном случае минимуму стоимостной функции. Нормализация признаков полностью реализована в модуле **featureNormalize.py**, который завершать не требуется.

После обучения параметров модели логистической регрессии настройками градиентного спуска, заданными по умолчанию в файле **main.py**, должен получиться результат, представленный на рис. 3, на котором помимо тренировочных данных изображена найденная граница решения для модели на основе логистической регрессии. Построение границы решения полностью реализовано в функции `plotDecisionBoundary` модуля **plotDecisionBoundary.py**.

Объекты, которые описываются точками на рис. 7, лежащими выше границы решения, будут отнесены алгоритмом к классу 1 (аттестован), иначе к классу 0 (не аттестован). Необходимо обратить внимание на то, что в процессе принятия решения алгоритмом на тренировочной базеданных возникают ошибки. Иногда точки, которые относятся к классу 1, классифицируются, как точки класса 0 и наоборот.

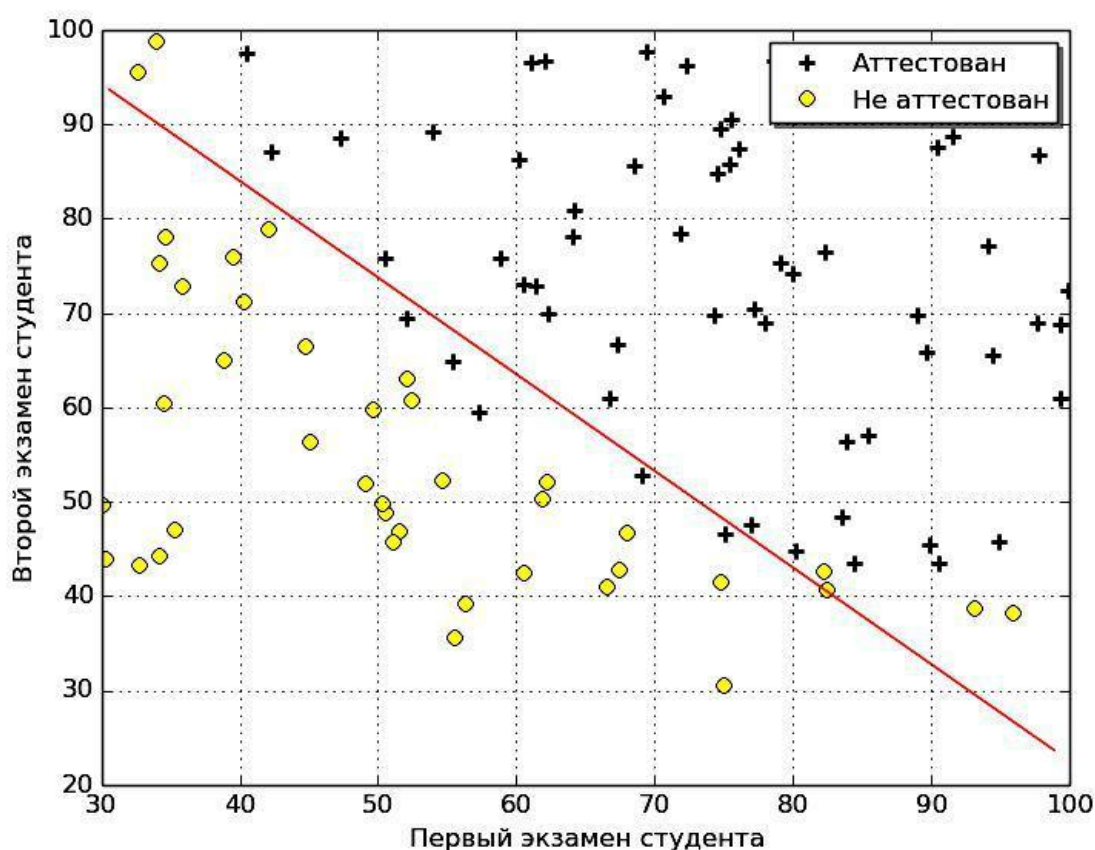


Рис. 7. Результат визуализации тренировочных данных с границей решения для логистической регрессии

Как и в практическом задании № 1, в данном случае возможно провести исследование сходимости градиентного спуска при различных настройках с использованием зависимости представленной на рис. 8.

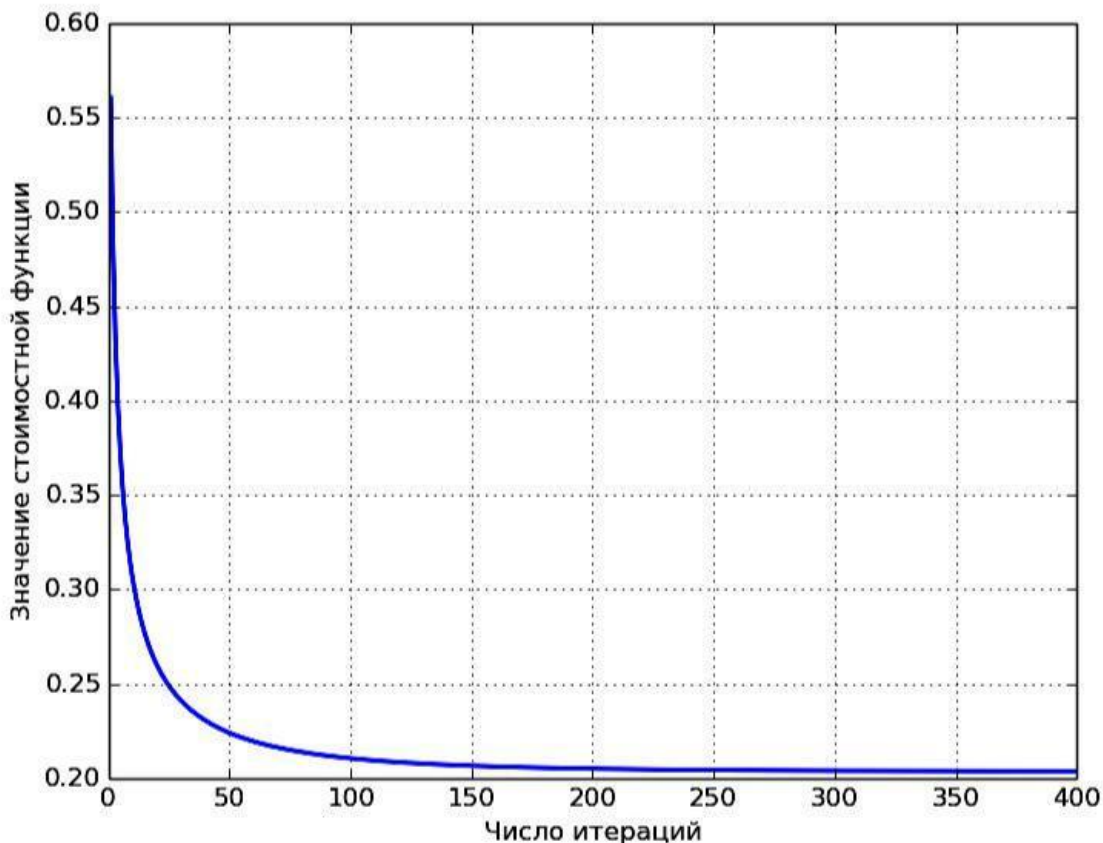


Рис. 8. Пример сходимости градиентного спуска для удачно подобранной скорости сходимости

4. Завершите программный код в модуле **predict.py**, который позволит выполнить предсказание метки класса для обученной модели логистической регрессии. В ходе предсказания порог классификатора необходимо выставить равным значению 0.5. При выполнении данной части задания могут понадобиться следующие функции из библиотеки

NumPy: dot и astype.

astype – позволяет выполнить приведение элементов массива определенному типу данных.

Так же совершенно будет необходима функция sigmoid, реализованная в модуле **sigmoid.py**.

5. После завершения предыдущих пунктов вычислите значение вероятности, с которой студент будет аттестован в случае, если его оценка за первый экзамен равна 45, а оценка за второй экзамен равна 85.

Обратите внимание на то, что перед выполнением процедуры предсказания требуется провести нормализацию признаков на соответствующие им математическое ожидание и среднеквадратическое отклонение.

б. Оцените долю правильных ответов обученной модели логистической регрессии. Необходимо обратить внимание на то, что русскоязычной литературе доля правильных ответов иногда обозначается словом точность (от англ. accuracy). Последний перевод не всегда является удачным, так как существует другая метрика оценки качества работы классификатора – precision, которая на русский язык так же переводится, как точность. Однако смысл accuracy и precision является абсолютно разным. **Accuracy** равна отношению числа ответов, для которых алгоритм выполнил предсказание метки класса верно, числу объектов, для которых алгоритм предсказал метку класса 1.

К общему числу объектов в базе данных. **Precision** равна отношению числа объектов, для которых алгоритм верно предсказал метку класса 1,

Используя файл **main.py** ответьте на следующие вопросы.

1. Чему равно значение стоимостной функции для случая, когда все параметры модели равны нулю?
2. Чему равны значения параметров обученной модели логистической регрессии для случая, когда параметр сходимости равен 1, а число итераций градиентного спуска равно 400?
3. Чему равна вероятность аттестации студента в случае, если его оценка за первый экзамен равна 45, а оценка за второй экзамен равна 85 для обученной в вопросе 2 модели?
4. Чему равна доля правильных ответов обученной в вопросе 2 модели логистической регрессии ?

Лабораторная работа №10. Визуализация данных на языке Python с помощью библиотеки `matplotlib`

1. Создать двухмерный график, применяя функция «`plot(x,y)`» для следующих данных:

```
x = numpy.arange(0.0, 1.0, 0.01)
y_sin = numpy.sin(2*2*numpy.pi*x)
y_cos = numpy.cos(2*numpy.pi*x)
```

2. Создать трехмерный график, применяя функцию «`plot(x,y)`» для данных:

```
theta = numpy.linspace(-4 * numpy.pi, 4 * numpy.pi, 100)
z = numpy.linspace(-2, 2, 100) r = z**2 + 1
x = r * numpy.sin(theta) y = r * numpy.cos(theta)
```

3. Создать трехмерный график, применяя функцию `scatter(x,y,z=0, zdir='y', color='red')` («`color`» может принимать значения 'red', 'blue', 'green', 'yellow' и т.д.) для данных:

```
x1 = numpy.random.sample(20)
y1 = numpy.random.sample(20)
x2 = numpy.random.sample(120)
y2 = numpy.random.sample(120)
x3 = numpy.random.sample(70)
y3 = numpy.random.sample(70)
```

4. Создать трехмерный график, применяя функцию «`plot_surface(x,y,z)`» для данных:

```
u = np.linspace(0, 2 * numpy.pi, 100)
v = np.linspace(0, numpy.pi, 100)
x = 10 * numpy.outer(numpy.cos(u), numpy.sin(v))
y = 10 * numpy.outer(numpy.sin(u), numpy.sin(v))
z = 10*numpy.outer(numpy.ones(numpy.size(u)),
numpy.cos(v))
```

5. Создать трехмерный график, применяя функцию «`imshow(rgb)`» для данных:

```
from matplotlib.colors import LightSource
X,Y=numpy.mgrid[-5:5:0.05,-5:5:0.05]
Z=numpy.sqrt(X**2+Y**2)+numpy.sin(X**2+Y**2)
ls = LightSource(azdeg=0,altdeg=65)
rgb = ls.shade(Z,plt.cm.copper)
```

Список литературы

1. Саммерфилд, М. Программирование на Python 3. Подробное руководство [Текст]/ М.Саммерфилд. – СПб.: Символ – Плюс, 2009. – 608 с.
2. Манакова, И.П. Информационные системы и технологии. Языки программирования высокого уровня. Программирование на языке Python [Электронный ресурс]: учеб.-метод, пособие / И.П. Манакова; М-во образования и науки РФ; ФГАОУ ВО «УрФУ им. первого Президента России Б.Н. Ельцина», Нижнетагил. технол. ин-т (филиал). - Нижний Тагил : НТИ (филиал) УрФУ, 2016. – 37 с.
3. Официальный сайт языка Python. URL: <http://python.org/> (дата обращения: 20.10.2017).
4. Официальный сайт библиотеки matplotlib. URL: <http://matplotlib.sourceforge.net/> (дата обращения: 15.10.2017).
5. Волохов В. Прикладная информатика: машинное обучение. URL: <http://volokhov.blogspot.com/2017/11/2017.html> (дата обращения: 28.11.2017).

Методические материалы

Рубцова Татьяна Павловна, Морозова Марина Валериевна

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО
ПРОГРАММИРОВАНИЮ НА ЯЗЫКЕ PYTHON**

Методические указания