



**Федеральное государственное автономное образовательное учреждение  
«Национальный исследовательский университет  
«Высшая школа экономики»**

Факультет компьютерных наук  
Департамент больших данных и информационного поиска

**Рабочая программа дисциплины  
«Основы и методология программирования»**

для образовательной программы «Прикладная математика и информатика»  
направления подготовки 01.03.02 Прикладная математика и информатика  
уровень - бакалавр

Разработчики программы:

Густокашин М. С. ([mgustokashin@hse.ru](mailto:mgustokashin@hse.ru))

Евстропов Г. О. ([gevstropov@hse.ru](mailto:gevstropov@hse.ru))

Зобнин А. И., кандидат физ.-мат. наук ([azobnin@hse.ru](mailto:azobnin@hse.ru))

Объедков С. А., кандидат техн. наук ([sobiedkov@hse.ru](mailto:sobiedkov@hse.ru))

Одобрена на заседании департамента больших данных и информационного поиска  
«\_\_» \_\_\_\_\_ 2017 г.

Руководитель департамента

В.В.Подольский \_\_\_\_\_

Утверждена Академическим советом образовательной программы

«\_\_» \_\_\_\_\_ 2017 г., № протокола \_\_\_\_\_

Академический руководитель образовательной программы

А.С. Конушин \_\_\_\_\_

Москва, 2017

*Настоящая программа не может быть использована другими подразделениями университета  
и другими вузами без разрешения подразделения-разработчика программы.*

## 1 Область применения и нормативные ссылки

Настоящая программа учебной дисциплины устанавливает минимальные требования к знаниям и умениям студента и определяет содержание и виды учебных занятий и отчетности. Программа предназначена для преподавателей, ведущих дисциплину «Введение в программирование на Erlang », учебных ассистентов и студентов направления подготовки 01.03.02 «Прикладная математика и информатика», обучающихся по образовательной программе «Прикладная математика и информатика»

Программа учебной дисциплины разработана в соответствии с:

- Образовательным стандартом ФГАОУ ВО НИУ ВШЭ по направлению подготовки 01.03.02 Прикладная математика и информатика <https://www.hse.ru/data/2017/09/04/1321436546/2.03.02%20%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%B0%D1%8F%20%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%20%D0%B8%20%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0.pdf> ;
- Образовательной программой «Прикладная математика и информатика», направление подготовки 01.03.02 «Прикладная математика и информатика»
- Объединенным учебным планом университета по образовательной программе «Прикладная математика и информатика», утвержденным в 2017 г.

## 2 Цели освоения дисциплины

Цели освоения дисциплины «Основы и методология программирования» — обучить студентов навыкам программирования на языках Python и C++, а также базовым алгоритмам, необходимым как в дальнейшем обучении (например, на курсе «Алгоритмы и структуры данных»), так и в работе по специальности.

## 3 Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:

- знать основные конструкции и идиомы языков программирования Python и C++, необходимые для изучения других дисциплин, предусмотренных базовым и рабочим учебными планами, а также для применения в профессиональной деятельности;
- уметь создавать программы, решающие задачи по заданному алгоритму на языках Python и C++, а также пользоваться интерпретатором языка Python и компиляторами языка C++ для их выполнения;
- иметь навыки формализации и решения практических задач по программированию.

В результате освоения дисциплины студент осваивает следующие компетенции:

Компетенция	Код по ФГОС/НИУ	Уровень формирования компетенций	Формы и методы обучения
Способен учиться, приобретать новые знания, умения, в том числе в области, отличной от профессиональной	УК-1	СД, МЦ	Лекции, практические занятия, домашние задания
Способен вести исследовательскую деятельность, включая анализ проблем, постановку целей и задач, выделение объекта и предмета исследования, выбор способа и методов исследования, а также оценку его качества	УК-6	РБ	Лекции, практические занятия, домашние задания
Способен критически оценивать и переосмысливать накопленный опыт	УК-9	МЦ, РБ	Лекции, практические занятия,

(собственный и чужой), рефлексировать профессиональную и социальную деятельность			домашние задания
Способен формализовать и алгоритмизировать поставленную задачу.	ПК-4	РБ, СД	Лекции, практические занятия, домашние задания, текущие и итоговые тестирования
Способен писать, оформлять, отлаживать и оптимизировать программный код.	ПК-5	РБ, СД	Лекции, практические занятия, домашние задания, текущие и итоговые тестирования
Способен вести письменную и устную коммуникацию на русском (государственном) языке в рамках профессионального и научного общения, как межличностного, так и группового	ПК-10	СД	Лекции, практические занятия, домашние задания
Способен строить профессиональную деятельность на основе правовых, профессиональных и этических норм и обязанностей, выполнять технологические требования и нормативы	ПК-16	МЦ, РБ, СД	Лекции, практические занятия, домашние задания

### **Виды и задачи профессиональной деятельности**

<b>Научно-исследовательские</b>	
Исследование и разработка математических моделей и методов, алгоритмов и программного обеспечения по тематике проводимых научно-исследовательских проектов;	НИД-3
<b>Проектные и производственно-технологические</b>	
Разработка архитектуры, алгоритмических и программных решений системного и прикладного программного обеспечения	ПД-3
Разработка программного и информационного обеспечения компьютерных систем, автоматизированных систем вычислительных комплексов, сервисов, операционных систем и распределенных баз данных	ПД-4
Изучение и использование различных языков программирования, алгоритмов, библиотек и пакетов программ при разработке программного обеспечения	ПД-5

## **4 Место дисциплины в структуре образовательной программы**

Настоящая дисциплина относится к профессиональному циклу, является базовой для подготовки бакалавра по направлению 01.03.02 «Прикладная математика и информатика».

Для освоения учебной дисциплины, студенты должны владеть знаниями и навыками в объеме программы средней школы по математике и информатике и ИКТ.

Основные положения дисциплины должны быть использованы в дальнейшем при изучении следующих дисциплин:

- Алгоритмы и структуры данных
- Алгоритмы и структуры данных 2
- Архитектура компьютера и операционные системы
- Программный проект
- Машинное обучение
- Машинное обучение 2
- Машинное обучение на больших данных

- Прикладной статистический анализ
- Теория вычислений
- Сложность вычислений и логика в теоретической информатике
- Коды с исправлением ошибок
- Машинное обучение и майнинг данных
- Параллельные и распределенные системы
- Комбинаторика, графы и вычислительная логика
- Теория информации
- Компьютерное зрение
- Функциональное программирование

## 5 Тематический план учебной дисциплины

### Пилотный поток

Название раздела	подразд.	всего часов	аудиторные часы		самост. работа
			лекции	практич. занятия	
Язык программирования C++	ДБДИИ		22	34	86
Алгоритмы	ДБДИИ		66	86	200
<b>Итого:</b>		<b>494</b>	<b>88</b>	<b>120</b>	<b>286</b>

### Основной поток

Название раздела	подразд.	всего часов	аудиторные часы		самост. работа
			лекции	практич. занятия	
Язык программирования Python	ДБДИИ		22	34	80
Язык программирования C++	ДБДИИ		44	66	158
Алгоритмы	ДБДИИ	40	20	20	48
<b>Итого:</b>		<b>494</b>	<b>88</b>	<b>120</b>	<b>286</b>

## 6 Формы контроля знаний студентов

### Пилотный поток

Тип контроля	Форма контроля	1 год			Параметры
		1-й мод.	2-й мод.	3-й мод.	
Текущий (неделя)	Контр. работа	2, 4, 6 недели	2, 4, 6 недели	2, 4, 6, 8 недели	Решение задач в системе Яндекс.Contest и/или письменная работа по теории
	Домашнее задание	5-я неделя	5-я неделя	5-я неделя	Решение задач в системе Яндекс.Contest, по 7-14 дней
Промежуточный	Экзамен	*			Устный экзамен и/или решение

					задач в системе Яндекс.Contest, от 2 до 4 часов.
Итоговый	Экзамен			*	Устный экзамен и/или решение задач в системе Яндекс.Contest, от 2 до 4 часов.

### **Основной поток**

Тип контроля	Форма контроля	1 год			Параметры
		1-й мод.	2-й мод.	3-й мод.	
Текущий (неделя)	Контроль работ	2, 4, 6 недели	2, 4, 6 недели	2, 4, 6, 8 недели	Решение задач в системе Яндекс.Contest и/или письменная работа по теории, проводится в несколько этапов по 80 мин (защита домашних задания)
	Домашнее задание	5-я неделя	5-я неделя	5-я неделя	Теоретические задания и/или решение задач в системе Яндекс.Contest, по 7-14 дней
Промежуточный	Экзамен	*			Письменный экзамен, от 2 до 4 часов.
Итоговый	Экзамен			*	Письменный экзамен, от 2 до 4 часов.

### **7 Критерии оценки знаний, навыков**

Для прохождения контроля студент должен продемонстрировать понимание основных конструкций языков программирования Python и C++, знание основных функций и классов стандартных библиотек этих языков, умение составить работающую программу на этих языках, решающую поставленную задачу.

Оценки по всем формам текущего контроля выставляются по 10-балльной шкале.

К особенностям проведения контроля относится использование системы проверки решений Яндекс.Contest (<http://contest.yandex.ru/>). С помощью этой системы студенты автоматически проверяют свои программы на заранее подготовленном наборе тестов, а преподаватели могут посмотреть и прокомментировать решения студентов. Кроме того, возможно использование системы Anytask (<http://anytask.org/>) для организации проверки решений.

### **8 Содержание дисциплины**

#### **Пилотный поток**

## **I. Язык программирования C++.**

1. Встроенные типы данных. Переменные. Константность. Области видимости переменных.
2. Инструкции (if, while, do, for, switch, goto).
3. Ввод-вывод (потоки).
4. Выражения. Обзор операторов языка C++. Приоритет операторов.
5. Последовательные контейнеры стандартной библиотеки C++. Цикл for по коллекции. Итераторы.
6. Ассоциативные контейнеры стандартной библиотеки C++.
7. Функции. Перегрузка функций. Шаблоны функций. Лямбда-функции.
8. Обзор алгоритмов стандартной библиотеки.
9. Классы. Уровни доступа к именам. Конструкторы. Деструктор. Перегрузка операторов.
10. Шаблоны классов.
11. Стек и динамическая память. Жизненный цикл объекта. Семантика перемещения.
12. Исключения. Идиома RAII. Умные указатели как иллюстрация идиомы RAII.

## **II. Алгоритмы**

1. Введение в теорию вероятностей.
2. Методы анализа алгоритмов. Схемы доказательства корректности на примере квадратичных сортировок. Тривиальные способы оценки времени работы.
3. Сортировка слиянием. Быстрая сортировка. Пример доказательства рекуррентных соотношений по индукции. Оценка снизу на сложность сортировки сравнениями. Сортировка подсчётом.
4. Цифровая сортировка, корзина сортировка. Сортировка случайных данных. Нахождение порядковой статистики за линейное время. Метод бинарного поиска.
5. Линейные структуры данных: односвязный и двусвязный списки, стек, очередь, дек. Двоичные и  $k$ -ичные пирамиды.
6. Амортизационный анализ. Метод кредитов и метод потенциалов. Реализация очереди на двух стеках и дека на трёх стеках.
7. Биномиальные и Фибоначчиевы пирамиды.
8. Бинарные деревья поиска. Сбалансированные деревья. Декартово дерево.
9. Хеширование. Постановка задачи, парадокс дней рождений. Полиномиальный хеш для строк и таблиц.
10. Хеширование других объектов. Хеш-таблицы с открытой и закрытой адресацией. Стратегии удаления и масштабирования открытых таблиц.
11. Частичные суммы, разреженные таблицы, деревья отрезков.
12. Многомерные задачи.  $K$ - $d$ -дерево.
13. Задачи LCA и LA.
14. Персистентность и персистентные структуры данных.
15. Простой перебор комбинаторных объектов. Динамическое программирование. Основные идеи и примеры. Приёмы параметризации. Генерация комбинаторных объектов.
16. Динамическое программирование на ациклических графах, подотрезках, поддеревьях и подмножествах.
17. Различные приёмы экономии памяти и восстановления ответа в задачах динамического программирования.
18. Введение в теоретико-числовые алгоритмы. Быстрое возведение в степень, расширенный алгоритм Евклида, решето Эратосфена и линейное решето, факторизация за  $O(n)$ .
19. Тест на простоту Миллера-Рабина и шифрование RSA.
20. Алгоритм Карацубы. Метод деревьев рекурсии и основная теорема.

21. Основные термины теории графов. Некоторые интересные множества на графах. Обход в глубину и обход в ширину. Топологическая сортировка и компоненты сильной связности.
22. Мосты и точки сочленения. Компоненты рёберной и вершинной двусвязности.
23. Задача о кратчайших путях. Алгоритмы Форда-Беллмана, Флойда-Уоршелла и Дейкстры.
24. Задача о минимальном остове. Лемма о безопасном ребре. Алгоритмы Прима, Борувки и Краскала.
25. Система непересекающихся множеств. Критерии оптимальности остовного дерева. Простейшие алгоритмы кластеризации на основе остовных деревьев.
26. Задача поиска шаблона в тексте. Префикс-функция и z-функция. Автомат префикс-функции.
27. Бор-дерево и сжатое бор-дерево. Алгоритм Ахо-Корасик.
28. Суффиксный массив и lcp
29. Алгоритм Укконена.

## **Основной поток**

### **I. Язык программирования Python.**

1. Целые числа и ввод-вывод.
2. Условный оператор
3. Цикл while
4. вещественные числа и строки
5. Функции и рекурсия
6. Цикл for
7. Списки
8. Линейный поиск и сортировка
9. Множества
10. Словари
11. Функциональное программирование
12. Классы

### **II. Язык программирования C++.**

1. C++: ввод и вывод, условный оператор, циклы, оператор switch.
2. Векторы и строки. Структуры.
3. Константность. Ссылки. Функции.
4. Контейнеры map и unordered\_map.
5. Контейнеры list и deque. Алгоритмы стандартной библиотеки.
6. Адаптеры stack, queue и priority\_queue.
7. Битовые операции. Выражения, операторы. Перегрузка операторов.
8. Класс Date. Шаблонный класс Matrix.
9. Жизненный цикл объекта. Конструкторы, деструктор. Семантика перемещения.
10. Исключения.
11. Указатели. Динамическая память. Динамические массивы. Null-terminated strings.
12. Идиома RAII.
13. Умные указатели
14. Одиночное наследование. Полиморфизм. Виртуальные функции.
15. Пример: реализация шаблонного класса vector.

### **III. Алгоритмы.**

1. Оценка сложности алгоритмов по времени и памяти.  $O$ -,  $o$ -,  $\Omega$ -,  $\omega$ -,  $\Theta$ -обозначения. Время работы в худшем, лучшем и среднем случаях. Линейный и двоичный поиск. Сортировка вставкой.

2. Сортировка слиянием. Оценка времени работы алгоритма при помощи рекуррентного соотношения. Примеры решения рекуррентных соотношений: решение с использованием дерева рекурсии и методом подстановки. Основная теорема.
3. Быстрая сортировка. Оптимальность сортировки слиянием.
4. Разделяй и властвуй: быстрое возведение в степень по модулю, выбор порядковой статистики за время  $O(n)$  — рандомизированный и детерминированный алгоритмы.
5. Амортизационный анализ: групповой анализ, банковский метод и метод потенциалов (на примере двоичного счетчика).
6. Линейные структуры данных: список, очередь, стек, дек, динамический массив.
7. Приоритетная очередь, ее реализация при помощи двоичной кучи.
8. Деревья поиска. Алгоритм построения двоичного дерева поиска. Сложность поиска в двоичном дереве поиска. Сортировка при помощи двоичного дерева поиска и ее связь с быстрой сортировкой. Сбалансированные деревья поиска.
9. Красно-черные деревья.
10. Деревья отрезков. Динамические порядковые статистики.

## 9 Примеры заданий контрольных работ

### Язык программирования C++

#### Задача 1 ("Общий префикс").

На входе даны две текстовых строки, состоящие из маленьких латинских букв алфавита. Длина строк не превосходит 1000 символов. Напечатайте длину наибольшего общего префикса этих строк.

#### Задача 2 ("Палиндромы").

Вам нужно проверить, является ли данная строка палиндромом, если учитывать в ней только латинские буквы и считать соответствующие заглавные и строчные буквы равными. На вход подаётся одна строка, состоящая из символов ASCII с кодами от 32 до 127. Длина строки не превосходит 65000 символов. Выведите YES, если строка является палиндромом в указанном смысле. Иначе выведите NO.

#### Задача 3 ("Предметный указатель").

Профессор написал научную книгу и составил для неё предметный указатель. Это список ключевых слов, для каждого из которых указана страница, на которой это слово встречается. Теперь профессор хочет для каждой страницы выписать в алфавитном порядке все ключевые слова, которые на эту страницу попали (если такие вообще есть). Помогите профессору решить эту задачу.

Формат ввода. Сначала задано натуральное число  $n$ , не превосходящее 1000 — количество слов, которое требуется обработать. Далее идут  $n$  строк. В каждой строке сначала записано ключевое слово. Затем идёт натуральное число, также не превосходящее 1000, — номер страницы. Ключевые слова состоят из латинских букв, не бывают пустым и по длине не превосходят 16 символов. Слова в списке, конечно, могут повторяться.

Формат вывода. Выпишите в порядке возрастания все страницы, на которых присутствуют ключевые слова. После каждого номера страницы через пробел выпишите в алфавитном порядке сами эти слова. Если на какой-то странице слово встретилось несколько раз, то повторять его не нужно. Завершающего пробела в конце строк быть не должно.

#### Задача 4 ("Сортировка посередине").

Вам дан набор целых чисел. Необходимо отсортировать указанный промежуток в середине этого набора и вывести получившиеся числа.

Формат ввода. Сначала задано натуральное число  $n$ , не превосходящее 1000 — количество чисел. Далее идут  $n$  целых чисел, по модулю не превосходящих 1000. В конце заданы два натуральных числа  $i$  и  $j$ , не превосходящие  $n$ . Вам надо будет отсортировать числа исходного набора в диапазоне от  $i$ -го до  $j$ -го включительно, где нумерация начинается с единицы.



Формат вывода. Отсортируйте указанный диапазон исходного набора чисел и напечатайте все числа по одному на строке.

#### Задача 5 ("MathVector").

Вам дан шаблонный класс `MathVector<T>`, представляющий из себя математический вектор с элементами типа `T` (не путать с `std::vector!`). Вам требуется исправить ошибки в коде этого класса и дописать унарный минус, бинарный минус и оператор `==`. Считайте, что все операторы всегда будет применяться только для векторов одинакового размера.

#### Задача 6 ("Объём фигур").

Вам дана структура `Point3D` и абстрактный класс `Figure3D`. Требуется реализовать классы-наследники `Ball` (шар) и `Cuboid` (прямоугольный параллелепипед), переопределив в них нужным образом виртуальные функции базового класса `Figure3D`. Конструктор класса `Ball` должен принимать на вход `const Point3D&` (центр шара) и `double r` (радиус). Конструктор класса `Cuboid` должен принимать на вход `const Point3D&` (центр) и три числа типа `double` — длины сторон параллелепипеда. (Можно считать, что стороны параллелепипеда будут параллельны осям координат, поэтому этими параметрами параллелепипед задаётся однозначно.) Оба класса в случае отрицательного радиуса или длины стороны должны генерировать в конструкторе исключение `std::domain_error` с текстом "Length must be non-negative".

## **10 Образовательные технологии**

Чтение лекций и проведение практических занятий. На семинарах (практических занятиях) разбираются прошлые домашние задачи, решаются текущие задачи, выдается новое домашнее задание.

## **11 Оценочные средства для текущего контроля и аттестации студента**

### **10.1 Оценочные средства для оценки качества освоения дисциплины в ходе текущего контроля**

Примерные задания для домашней работы:

1. Напишите функции, переводящие целое число в строку и наоборот.
2. Реализуйте шаблонный класс «Многочлен от одной переменной».
3. Создайте свою реализацию шаблонного адаптера `stack`.
4. Напишите собственную реализацию шаблона `vector`.
5. Напишите собственную версию «умного» указателя `shared_ptr`.
6. Напишите свои версии следующих утилит Unix: `wc`, `rev`, `cat`, `sort`, `uniq`.
7. Отрецензируйте данный код, укажите в нём все ошибки.
8. Оцените сложность данного алгоритма относительно размера входа.
9. Приведите рекуррентное соотношение, описывающее сложность данного алгоритма и перепишите это соотношение в замкнутой форме.

### **10.2 Примеры заданий промежуточной и итоговой аттестации**

Примерные задания для экзаменационной работы:

1. Реализуйте алгоритм быстрого возведения в степень.
2. Напишите функцию, определяющую, является ли данная строка палиндромом.
3. Напишите функцию, проверяющую, является ли массив отсортированным.
4. Напишите программу, которая для каждого слова из текстового файла подсчитывала бы количество его вхождений.
5. Напишите функцию, которая бы выводила на экран битовое представление числа.
6. Напишите функцию, которая проверяла бы, что строка, состоящая из открывающих и закрывающих скобок, корректно сформирована.
7. Напишите собственную реализацию шаблонных функций `copy_if`, `rotate` и `unique`.

8. Реализуйте класс «Рациональное число».

## 11 Порядок формирования оценок по дисциплине

Преподаватель оценивает работу студентов на семинарских и практических занятиях (активность студентов, правильность решения задач и т. д.). Оценки за работу на семинарских и практических занятиях преподаватель выставляет в рабочую ведомость. Для округления оценок применяются стандартные математические правила.

### Формулы оценок для пилотного потока

#### Промежуточный контроль (I модуль)

$$O_{\text{накопленная } 1} = 0,4 \cdot O_{\text{дз}} + 0,6 \cdot O_{\text{контр. раб.}}$$

$$O_{\text{промежуточная } 1} = 0,6 \cdot O_{\text{накопленная } 1} + 0,4 \cdot$$

$$O_{\text{экз. } 1}$$

#### Итоговый контроль (III модуль)

$$O_{\text{накопленная } 2-3} = (0,25 \cdot O_{\text{дз } 2} + 0,3 \cdot O_{\text{контесты}} + 0,15 \cdot O_{\text{теор. контр. раб.}}) / 0,7$$

$$O_{\text{итоговая}} = 0,7 \cdot O_{\text{накопленная } 2-3} + 0,3 \cdot O_{\text{экз. } 3}$$

### Формулы оценок для основного потока

#### Промежуточный контроль (I модуль)

$$O_{\text{накопленная } 1} = 1 \cdot O_{\text{дз}}$$

$$O_{\text{промежуточная } 1} = 0,6 \cdot O_{\text{накопленная } 1} + 0,4 \cdot O_{\text{экз. } 1}$$

Раз в 2 недели проводится аудиторная защита очередной части домашней работы в форме контрольной работы.

#### Итоговый контроль (III

$$\text{модуль)} O_{\text{накопленная } 2-3} = 0,4 \cdot O_{\text{дз}} +$$

$$0,6 \cdot O_{\text{контр. раб.}} O_{\text{итоговая}} = 0,6 \cdot$$

$$O_{\text{накопленная } 2-3} + 0,4 \cdot O_{\text{экз. } 3}$$

## 12 Учебно-методическое и информационное обеспечение дисциплины

### 12.1 Основная литература

Bjarne Stroustrup (2013). A Tour of C++. Addison-Wesley. ISBN 978-0321958310.

Bjarne Stroustrup (2013). The C++ Programming Language (4th edition). Addison-Wesley. ISBN 978-0321563842.

Кормен Т.Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К. Алгоритмы: построение и анализ. — 2-е издание: Пер. с англ. — М.: Вильямс, 2007.

### 12.2 Дополнительная литература

Источники в Интернете:

<http://docs.python.org/> – документация по языку Python 3.

<http://cppreference.com/> – документация по языку C++.

<http://isocpp.github.io/CppCoreGuidelines/> – C++ Core Guidelines.

На вики-странице курса могут выкладываться дополнительные материалы по курсу.

### 12.3 Программные средства

Для успешного освоения дисциплины, студент использует следующие программные средства:

- интерпретатор языка Python 3.x.
- компилятор языка C++14 (g++ 4.8 / clang 4.8 или более новых версий).

#### **12.4 Дистанционная поддержка дисциплины**

Проверка заданий осуществляется с помощью системы Яндекс.Контест, а для рецензирования кода по усмотрению преподавателей может использоваться система Anytask. В частности, с помощью этих систем студенты могут вести диалог с преподавателями.

#### **13 Материально-техническое обеспечение дисциплины**

Практические занятия проводятся в компьютерных классах. На лекциях и практических занятиях используется проектор.