

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

Факультет компьютерных наук  
Департамент больших данных и информационного поиска

Утверждаю  
Академический руководитель  
образовательной программы  
по направлению 01.03.02  
«Прикладная математика и информатика»  
А.С.Конущин

«\_\_» \_\_\_\_\_ 2016 г.

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

для образовательной программы 01.03.02  
«Прикладная математика и информатика» подготовки бакалавра

Разработчики программы:

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

[Евстропов Г. О. (gevstropov@hse.ru)]

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

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

Одобрена на заседании

Департамента Больших данных и информационного поиска

«\_\_» \_\_\_\_\_ 201\_ г.

Руководитель департамента БДИП

В.В.Подольский

Утверждена Академическим советом образовательной программы

«\_\_» \_\_\_\_\_ 201\_ г., № протокола \_\_\_\_\_

Менеджер ДБДИП

\_\_\_\_\_ И.И. Алескерова

Москва, 2016

*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 данную дисциплину, учебных ассистентов и студентов направления 01.03.02 «Прикладная математика и информатика» подготовки бакалавра, изучающих дисциплину «Основы и методология программирования».

Программа разработана в соответствии с:

- Образовательным стандартом федерального государственного автономного образовательного учреждения высшего профессионального образования «Национальный исследовательский университет «Высшая школа экономики»;
- Образовательной программой «Прикладная математика и информатика»;

Рабочим учебным планом университета по направлению 01.03.02 «Прикладная математика и информатика» подготовки бакалавра, утвержденным в 2016 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	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции
Способность к анализу и синтезу на основе системного подхода	ОНК-1	Умение анализировать условия задач по программированию	Лекции, практические занятия, домашние задания
Способность перейти от проблемной ситуации к проблемам, задачам и лежащим в их основе противоречиям	ОНК-2	Умение формализовать условие задачи по программированию	Лекции, практические занятия, домашние задания

Национальный исследовательский университет «Высшая школа экономики»  
 Программа дисциплины «Основы и методология программирования»  
 для направления 01.03.02 «Прикладная математика и информатика» подготовки бакалавра

Способность использовать методы критического анализа, развития научных теорий, опровержения и фальсификации, оценить качество исследований в некоторой предметной области	ОНК-3	Умение производить логические рассуждения в задачах по программированию	Лекции, практические занятия, домашние задания
Способность приобретать новые знания с использованием научной методологии и современных образовательных и информационных технологий	ОНК-6	Способность находить и самостоятельно осваивать нужную информацию из документации, литературы, сети Интернет	Лекции, практические занятия, домашние задания
Способность порождать новые идеи (креативность)	ОНК-7	Умение применять язык программирования в новых ситуациях	Лекции, практические занятия, домашние задания
Способность к письменной и устной коммуникации на государственном языке и необходимое знание второго языка, владение иностранным языком на уровне, достаточном для разговорного общения, а также для поиска и анализа иностранных источников информации	ИК-1	Умение логически правильно излагать свои мысли, умение пользоваться документацией на иностранном языке	Лекции, практические занятия, домашние задания
Умение работать на компьютере, навыки использования основных классов прикладного программного обеспечения, работы в компьютерных сетях, составления баз данных	ИК-2	Способность записать и выполнить программу на компьютере на требуемых языках программирования	Лекции, практические занятия, домашние задания
Способность аналитически работать с информацией из различных источников, включая глобальных компьютерных сетях	ИК-4	Умение логически анализировать информацию, умение пользоваться документацией	Лекции, практические занятия, домашние задания
Способность рефлексии и критического переосмысления накопленного опыта	СЛК-2	Умение логически анализировать накопленный опыт	Лекции, практические занятия, домашние задания
Способность понимать сущность и значение информации в развитии современного информационного общества, сознавать опасности и угрозы, возникающие в этом процессе, соблюдать основные требования информационной безопасности, в том числе защиты государственной тайны	ЛК-7	Понимание основ теории кодирования информации	Лекции, практические занятия, домашние задания
Способность демонстрации общенаучных базовых знаний естественных наук, математики и информатики, понимание основных фактов, концепций, принципов теорий, связанных с прикладной математикой и	К-1	Понимание основных результатов информатики, способность демонстрации этого понимания	Лекции, практические занятия, домашние задания

информатикой			
Способность осуществлять целенаправленный многокритериальный поиск информации о новейших научных и технологических достижениях в сети Интернет и из других источников	К-5	I Способность находить и самостоятельно осваивать нужную информацию из документации и сети Интернет	Лекции, практические занятия, домашние задания
Способность решать задачи производственной и технологической деятельности на профессиональном уровне, включая разработку математических моделей, алгоритмических и программных решений	К-8	I Способность создать и выполнить программу, решающую заданную задачу на компьютере	Лекции, практические занятия, домашние задания
Способность применять в профессиональной деятельности современные языки программирования и языки баз данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии и т. п.	ПК-9	Способность создавать программы для языков программирования Python и C++	Лекции, практические занятия, домашние задания

#### 4 Место дисциплины в структуре образовательной программы

Настоящая дисциплина относится к циклу математических и естественнонаучных дисциплин, 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.

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

- Алгоритмы и структуры данных
- Компьютерные системы
- Технологии программирования
- Машинное обучение
- Вычислительные методы
- Алгоритмы для больших данных
- Теория вычислений
- Алгоритмы и сложность
- Логические методы в информатике
- Комбинаторные методы в информатике
- Модели вычислений
- Теория информации
- Вероятностные алгоритмы и протоколы
- Теоретическая информатика
- Теория информации, кодирования и поиска

- Компьютерная лингвистика
- Введение в компьютерное зрение
- Майнинг данных
- Методы обработки текстов на естественном языке
- Функциональное программирование
- Операционные системы
- Распределенные системы
- Распределенные алгоритмы
- Архитектура компьютера

## 5 Тематический план учебной дисциплины

### Пилотный поток

№	Название раздела	Подразделение	Всего часов	Аудиторные часы				Самостоятельная работа
				Лекции	Семинары	Практические занятия	Другие виды работы	
1	Язык программирования С++	ДБДИИ		22		34		
2	Алгоритмы	ДБДИИ						

### Основной поток

№	Название раздела	Подразделение	Всего часов	Аудиторные часы				Самостоятельная работа
				Лекции	Семинары	Практические занятия	Другие виды работы	
1	Язык программирования Python	ДБДИИ		22		34		
2	Язык программирования С++	ДБДИИ						
3	Алгоритмы	ДАДИИ	40	20		20		

## 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	6-я неделя	6-я неделя	6-я неделя	Решение задач в системе Яндекс.Contest, может проводиться в несколько этапов по 80 мин.

	Домашнее задание	5-я неделя	5-я неделя	5-я неделя	Решение задач в системе Яндекс.Contest, по 7-14 дней
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)	Контр. работа	6-я неделя	6-я неделя	6-я неделя	Решение задач в системе Яндекс.Contest, может проводиться в несколько этапов по 80 мин.
	Домашнее задание	5-я неделя	5-я неделя	5-я неделя	Решение задач в системе Яндекс.Contest, по 7-14 дней
Итоговый	Экзамен	*		*	Устный экзамен и/или решение задач в системе Яндекс.Contest, 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.

## **8 Содержание дисциплины**

### **Пилотный поток**

#### **I. Язык программирования C++.**

1. История и стандарты языка C++. Популярные компиляторы.
  2. Встроенные типы данных. Переменные. Константность. Области видимости переменных.
  3. Консольный и файловый ввод-вывод (потoki).
  4. Функции. Перегрузка функций. Шаблоны функций. Лямбда-функции.
  5. Выражения. Обзор операторов языка C++. Оператор присваивания. Приоритет операторов. Value categories (lvalue, rvalue, xvalue).
  6. Битовые операторы.
  7. Инструкции (if, while, do, for, switch, goto).
  9. Указатели. Арифметика указателей. Оператор взятия адреса. Динамическая память и стек.
  11. Ссылки. Константность. CV-квалификаторы. Способы передачи аргументов в функцию.
  12. Контейнеры стандартной библиотеки C++: std::vector, std::string, std::list, std::deque.
- Списки инициализации. Цикл for по коллекции. Итераторы.

13. Ассоциативные контейнеры стандартной библиотеки C++ (`std::map` и `std::set`, `std::unordered_map` и `std::unordered_set`).
14. Обзор алгоритмов стандартной библиотеки. Объекты-функции. Обертка `std::function`.
15. Структуры и классы. Уровни доступа к именам в классе. Конструкторы. Деструктор. Перегрузка операторов.
16. Шаблоны классов. Шаблонные функции внутри класса. Шаблонные параметры шаблонов.
17. Одиночное наследование. Полиморфизм. Виртуальные функции. Работа с объектами производных классов через указатели (ссылки) на базовый класс. Операторы преобразования типов.
19. Идиома RAII. Генерация и обработка исключений. Гарантии безопасности исключений.
20. Умные указатели как иллюстрация идиомы RAII.

## II. Алгоритмы

### Основной поток

#### I. Язык программирования Python.

#### II. Язык программирования C++.

1. История C++. Использование языка. Мультипарадигменность. Стандарты языка. Популярные компиляторы.
2. Типы данных и переменные. Области видимости переменных. Встроенные типы данных. Перечисления.
3. Выражения. Обзор операторов языка C++. Оператор присваивания. Таблица приоритета операторов.
4. Работа с консольным и файловым вводом-выводом с помощью потоков.
5. Инструкции (`if`, `while`, `do`, `for`, `switch`, `goto`).
6. Функции. Перегрузка функций. Шаблоны функций.
7. Массивы фиксированного размера. Динамические массивы.
8. Шаблоны `std::vector` и `std::string`. Списки инициализации. Функции-члены стандартных последовательных контейнеров (`push_back`, `size`, `capacity`, `empty`, `clear`, ...). Цикл `for` по коллекции.
9. Указатели. Арифметика указателей. Оператор взятия адреса. Итераторы. Динамическая память и стек.
10. Ссылки. Константность. Способы передачи аргументов в функцию.
11. Последовательные контейнеры стандартной библиотеки C++ (на примере `std::list`, `std::deque`), адаптеры контейнеров (на примере `std::stack` и `std::queue`). Стандартные функции последовательных контейнеров. Категории итераторов.
12. Ассоциативные контейнеры стандартной библиотеки C++ (`std::map` и `std::set`, `std::unordered_map` и `std::unordered_set`). Стандартные функции ассоциативных контейнеров.
13. Обзор алгоритмов стандартной библиотеки. Объекты-функции и лямбда-функции. Обертка `std::function`.
14. Структуры и классы. Уровни доступа к именам в классе. Конструкторы. Деструктор. Оператор присваивания. Функции-члены, генерируемые компилятором по умолчанию.
15. Перегрузка операторов в классе. Пример: класс «Комплексное число».
16. Шаблоны классов. Шаблонные функции внутри класса.

17. Одиночное наследование. Полиморфизм. Виртуальные функции. Работа с объектами производных классов через ссылки на базовый класс. Операторы преобразования типов.

19. Единицы трансляции. Заголовочные файлы. Компонировка программы. Правило одного определения. Пространства имен.

20. Идиома RAII. Генерация и обработка исключений. Гарантии безопасности исключений.

### 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 Образовательные технологии

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

## 10 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 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. Оценка по 10-ти балльной шкале за работу на семинарских и практических занятиях определяется перед промежуточным или итоговым контролем —  $O_{аудиторная}$ .

### Формулы оценок для пилотного потока

#### Промежуточный контроль (I модуль)

$$O_{накопленная\ 1} = 0,2 \cdot O_{аудиторная\ 1} + 0,3 \cdot O_{дз\ 1} + 0,5 \cdot O_{контр.\ раб.\ 1}$$

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

Или, более просто, вес аудиторной работы в промежуточной оценке – 12%, домашней работы – 18%, контрольной работы – 30%, а экзамена – 40%.

#### Итоговый контроль (III модуль)

$$O_{накопленная\ 2-3} = 0,2 \cdot O_{аудиторная\ 2-3} + 0,3 \cdot (O_{дз\ 2} + O_{дз\ 3}) + 0,5 \cdot (O_{контр.\ раб.\ 2} + O_{контр.\ раб.\ 3})$$

$$O_{итоговая} = 0,6 \cdot O_{накопленная\ 2-3} + 0,4 \cdot O_{экс.\ 3}$$

Или, более просто, вес аудиторной работы за 2-3 модули в итоговой оценке – 12%, суммарный вес домашних работ – 18%, суммарный вес контрольных – 30%, вес экзамена – 40%.

В диплом выставляется результирующая оценка по учебной дисциплине по формуле

$$O_{результ.} = 1/3 \cdot O_{промежуточная\ 1} + 2/3 \cdot O_{итоговая}$$

### Формулы оценок для основного потока

#### Промежуточный контроль (I модуль)

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

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

Или, более просто, вес аудиторной работы в промежуточной оценке – 12%, домашней работы – 18%, контрольной работы – 30%, а экзамена – 40%.

### **Итоговый контроль (III модуль)**

$$O_{\text{накопленная } 2-3} = 0,4 \cdot (O_{\text{дз } 2} + O_{\text{дз } 3}) + 0,6 \cdot (O_{\text{контр. раб. } 2} + O_{\text{контр. раб. } 3})$$

$$O_{\text{итоговая}} = 0,6 \cdot O_{\text{накопленная } 2-3} + 0,4 \cdot O_{\text{экз. } 3}$$

Или, более просто, суммарный вес домашних работ в итоговой оценке – 24%, суммарный вес контрольных – 36%, вес экзамена – 40%.

В диплом выставляется результирующая оценка по учебной дисциплине по формуле

$$O_{\text{результ.}} = 1/3 \cdot O_{\text{промежуточная } 1} + 2/3 \cdot O_{\text{итоговая}}$$

Способ округления промежуточной, итоговой и результирующей оценок по учебной дисциплине в случае, когда её дробная часть попадает в интервал  $[0,3; 0,7]$  – на усмотрение преподавателя. В остальных случаях применяются математические правила округления.

## **12 Учебно-методическое и информационное обеспечение дисциплины**

### **12.1 Базовый учебник**

[Укажите базовый учебник. Норматив обеспеченности студентов базовым учебником - не менее 50 %. Ридеры указываются при отсутствии базового учебника или неполного покрытия базовым учебником основных тем программы один из предложенных вариантов формирования оценки, которая идет в диплом. Укажите, если доступна электронная версия базового учебника].

### **12.2 Основная литература**

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

Vjarne 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.

### **12.3 Дополнительная литература**

Источники в Интернете:

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

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

<http://www.parashift.com/c++-faq-lite/> – ответы на часто задаваемые вопросы по C++.

### **12.4 Программные средства**

Для успешного освоения дисциплины, студент использует следующие программные средства:

- интерпретатор языка Python 3.x.
- компилятор языка C++11 (g++ 4.8 / clang 4.8).

### **12.5 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.