



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

Московский институт электроники и математики им. А.Н. Тихонова

Департамент Прикладной математики

Рабочая программа дисциплины Алгоритмизация и программирование

для направления 01.03.04. «Прикладная математика» подготовки бакалавра

Авторы программы:

Зонтов Ю.В.

Чеповский А.А.

Одобрена на заседании Департамента Прикладной математики « ____ » _____ 2017 г.

Руководитель департамента Белов А.В. _____

Рекомендована Академическим советом образовательной программы
« ____ » _____ 2017 г., № протокола _____

Утверждена « ____ » _____ 2017 г.

Академический руководитель образовательной программы
Буровский Е.А. _____

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 данную дисциплину, учебных ассистентов и студентов направления подготовки 01.03.04.«Прикладная математика», изучающих дисциплину «Алгоритмизация и программирование».

Программа разработана в соответствии с:

- Федеральным Государственным Образовательным Стандартом (ФГОС) по направлению подготовки бакалавра 01.03.04.«Прикладная математика», уровень подготовки – бакалавр;
- Образовательной программой 01.03.04.«Прикладная математика» подготовки бакалавра.
- Рабочим учебным планом университета по направлению 01.03.04.«Прикладная математика» подготовки бакалавра, утвержденным в 2017 г.

2 Цели освоения дисциплины

Дисциплина «Алгоритмизация и программирование» призвана обучить студентов принципам функционирования информационных технологий, базовым концепциям технологий программирования и эффективным реализаций алгоритмов.

Цели освоения дисциплины:

- **Знакомство** слушателей дисциплины с основными парадигмами и теоретическими основами программирования, терминологией объектно-ориентированного программирования.
- **Формирование** представления о возможностях и особенностях объектно-ориентированных языков программирования при проектировании, разработке и отладке компьютерных программ;
- **Изучение** теоретических основ программирования, методов реализации алгоритмов различного типа, базовых принципов объектно-ориентированного программирования, основных подходов машинного обучения;
- **Выработка навыков** и умений программирования на языке C++ с использованием широко используемых библиотек;
- **Получение опыта** работы с механизмами статического полиморфизма, такими как шаблоны функций и классов; поиска эффективных реализаций различных алгоритмов, практических навыков разработки прикладных программ, в том числе для методов анализа данных.



Теоретический материал курса подкрепляется практическими занятиями по программированию (курс «Компьютерный практикум») в целях реализации предлагаемых заданий в рамках изучаемой тематики.

3 Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:

- **Знать** парадигмы и методологии программирования, особенности наиболее распространенных объектно-ориентированных языков программирования, основные принципы автоматизированных систем сбора, анализа, обработки, передачи и хранения информации;
- **Уметь** применять в профессиональной деятельности современные объектно-ориентированные языки программирования; профессионально решать задачи производственной и технологической деятельности с учетом современных достижений науки и техники, включая: разработку алгоритмических и программных решений в области системного и прикладного программирования, разработку математических, информационных и имитационных моделей по тематике выполняемых исследований;
- **Владеть** базовыми математическими знаниями и информационными технологиями, эффективно применять их для решения научно-технических и прикладных задач, связанных с развитием и использованием информационных технологий, математическими алгоритмами и методами автоматизации сбора и анализа данных при построении систем машинного обучения;
- **Иметь навыки (приобрести опыт)** постановки математических и информационных задач, разработки технического задания по проектированию программного обеспечения, описания алгоритмов решения поставленной задачи и разработки программного кода на языке C++ тестирования программного обеспечения, использования методов машинного обучения.

Уровни формирования компетенций:

РБ – ресурсная база, в основном теоретические и предметные основы (знания, умения);

СД – способы деятельности, составляющие практическое ядро данной компетенции;

МЦ – мотивационно-ценностная составляющая, отражает степень осознания ценности компетенции человеком и готовность ее использовать.

В результате освоения дисциплины студент осваивает следующие компетенции:

Компетенция	Код по ФГОС/ НИУ	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
Способен решать проблемы в профессиональной деятельности на основе анализа и синтеза	УК-3	СД	Подбор и анализ материалов по теме задания. Самостоятельно пишет программы по заданию преподавателя	Лекции, семинарские занятия, самостоятельное выполнение домашних заданий.	Домашние задания, экзамен.
Способен оценивать потребность в ресурсах и планировать их использование при решении задач в профессиональной	УК-4	РБ	Владеет навыками разработки алгоритмических и программных решений с использованием современных	Лекции, семинарские занятия, самостоятельное выполнение домашних заданий.	Домашние задания, экзамен.



Компетенция	Код по ФГОС/ НИУ	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
деятельности			технологий программирования		
Способен работать с информацией: находить, оценивать и использовать информацию из различных источников, необходимую для решения научных и профессиональных задач (в том числе, на основе системного подхода)	УК 5	РБ	Применяет современные стандартные среды разработки (IDE) при создании и отладке программных продуктов	Лекции, семинарские занятия, самостоятельное выполнение домашних заданий.	Домашние задания, экзамен.
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	Понимает стадии и этапы разработки программного обеспечения. Демонстрирует знание современных языков программирования низкого и высокого уровня	Лекции, семинарские занятия, самостоятельное выполнение домашних заданий.	Домашние задания, экзамен.
Способен проектировать и разрабатывать компоненты программного обеспечения на основе современных парадигм, технологий и языков программирования	ПК-4	СД	Разрабатывает техническое задание на разработку программных компонент автоматизированных систем.	Лекции, семинарские занятия, самостоятельное выполнение домашних заданий.	Домашние задания, экзамен.
Способен применять знания жизненного цикла современных проектов по созданию и эксплуатации программных систем и инструментальные средства управления проектами в области ИТ.	ПК-8	СД	Демонстрирует знание современных языков программирования	Лекции, семинарские занятия, самостоятельное выполнение домашних заданий.	Домашние задания, экзамен.
Способен использовать и развивать методы математического моделирования и применять аналитические и научные пакеты прикладных программ	ПК-11	МЦ	Применяет современные стандартные среды разработки (IDE) при создании и отладке программных продуктов	Лекции, семинарские занятия, самостоятельное выполнение домашних заданий.	Домашние задания, экзамен.
Способен применять математические алгоритмы и методы	ПК-12	МЦ	Разрабатывает алгоритмы на основании базовых подходов и	Лекции, семинарские занятия, самостоятельное выполнение до-	Домашние задания, экзамен.



Компетенция	Код по ФГОС/ НИУ	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
автоматизации сбора и анализа данных при построении систем машинного обучения.			методов машинного обучения.	машинных заданий.	

4 Место дисциплины в структуре образовательной программы

Дисциплина относится к профессиональному учебному циклу (Б.3) и блоку дисциплин, обеспечивающих базовую (общепрофессиональную) подготовку.

Для освоения учебной дисциплины, студенты должны изучить следующие учебные курсы:

- Линейная алгебра и аналитическая геометрия (1 курс);
- Математический анализ (1 курс);
- Алгоритмизация и программирование (1 курс);
- Компьютерный практикум (1 курс);
- Теория вероятностей и математическая статистика (2 курс)

Для освоения учебной дисциплины, студенты должны владеть следующими знаниями и компетенциями:

- Владеть основами программирования на языке C.
- Знать методы и базовые алгоритмы обработки информационных структур, методы анализа сложности алгоритмов.

Основные положения дисциплины и освоенные компетенции (УК-3, УК-4, УК-5, УК-6, ПК-4, ПК-8, ПК-11, ПК-12) должны быть использованы в дальнейшем при изучении дисциплин:

- Компьютерный практикум (3 курс)
- Операционные системы (3 курс);
- Базы данных (3 курс);
- Численные методы (3-4 курсы);
- Междисциплинарная курсовая работа (3 курс);
- Производственная практика (3 курс);

5 Тематический план учебной дисциплины

№	Название раздела	Всего часов	Аудиторные часы			Самостоятельная работа
			Лекции	Семинары	Практические занятия	
1	Поколения языков программирования. Знакомство с ООП.	5	2	1		2
2	Введение в программирование на C++.	7	4	1		2



3	Обработка исключений.	8	2	2		4
4	Построение пользовательских типов данных.	28	4	8		16
5	Наследование, виртуальные функции, полиморфизм.	28	8	6		14
6	Знакомство с библиотекой STL.	24	4	6		14
7	Шаблоны в C++.	24	4	6		14
8	Знакомство с паттернами проектирования.	22	8	6		8
9	Семейство паттернов MVC	8	2	2		4
10	Элементы пользовательского интерфейса.	10	2	2		6
11	Архитектура документ-представление.	10	4	2		4
12	Алгоритмы на графах.	20	4	6		10
13	Основы машинного обучения на C++.	20	4	6		10
14	Многопоточное программирование.	14	4	4		6
	Итого	228	56	58		114

6 Формы контроля знаний студентов

Тип контроля	Форма контроля	1 год				Параметры
		1	2	3	4	
Промежуточный	Экзамен		*			Создание программы в IDE и защита ее текста, 80 минут
Итоговый	Экзамен				*	Создание программы в IDE и защита ее текста, 80 минут

6.1. Критерии оценки знаний, навыков

Существуют следующие формы контроля:

- Выполнение домашних заданий;
- Текущий контроль на лекциях и семинарских занятиях;
- Экзамены в конце модулей: 2(промежуточный) и 4(итоговый).

Промежуточная оценка (за 1-2 модули) состоит из:

- Экзамена в конце 2 модуля (50%);
- Выполнения домашних заданий, текущего контроля 1-го, 2-го модулей (50%).

Накопленная итоговая оценка состоит из:

- Финального экзамена (50%);
- Домашних заданий и текущего контроля 3-го, 4-го модулей (40%).
- Промежуточной оценки (10%);

6.2. Порядок формирования оценок по дисциплине

Преподаватель оценивает работу студентов на семинарах: оценивается активность студента в дискуссиях, скорость и правильность решения задач. Оценки за работу на семинарских занятиях преподаватель выставляет в рабочую ведомость. Накопленная оценка по 10-ти балль-



ной шкале за работу на семинарских занятиях определяется перед промежуточным или итоговым контролем – $O_{аудиторная}$.

Преподаватель оценивает самостоятельную работу студентов: оценивается правильность и количество реализованных заданий. Оценки за самостоятельную работу студента преподаватель выставляет в рабочую ведомость. Накопленная оценка по 10-ти балльной шкале за самостоятельную работу определяется перед промежуточным или итоговым контролем – $O_{сам. работа}$.

Накопленная оценка за текущий контроль как 1-2 модулей, так и 3-4 модулей учитывает результаты студента по текущему контролю за соответствующие модули следующим образом:

$$O_{накопленная} = 0,4 \cdot O_{аудиторная} + 0,6 \cdot O_{сам. работа}$$

Результирующая оценка за текущий контроль в форме экзамена по итогам 2 модуля рассчитывается следующим образом:

$$O_{промеж} = 0,5 \cdot O_{накопленная} + 0,5 \cdot O_{экзамен}$$

Результирующая оценка за итоговый контроль в форме экзамен по итогам 4 модуля выставляется по его результатам – $O_{результ}$.

$$O_{результ} = 0,5 \cdot O_{экзамен} + 0,4 \cdot O_{накопленная} + 0,1 \cdot O_{промеж}$$

Способ округления во всех случаях: арифметический.

На передаче студенту не предоставляется возможность получить дополнительный балл для компенсации оценки за текущий контроль.

7 Содержание дисциплины

№	Наименование раздела дисциплины	Содержание раздела	Аудиторная работа	Самостоятельная работа	Литература к разделу
1.	Парадигмы программирования. Знакомство с ООП.		7	2	[1, 7]
2.	Введение в программирование на C++.	Пространства имен. Ввод/вывод в C++. Операции инкремента и декремента. Особенности операторов инкремента и декремента при работе с указателями. Ссылки. Выделение/освобождение памяти в C++. Константы и макроопределения.	11	2	[1-3]
3.	Обработка исключений.	Понятие исключения. Механизм генерации и обработки исключений.	10	4	[1-3]
4.	Построение пользовательских типов данных.	Создание пользовательских классов. Принцип композиции при конструировании новых классов. Инициализация данных объекта класса (Конструктор по умолчанию, кон-	32	16	[1-3]



		структор с параметрами). Назначение деструктора. Способы передачи объекта в функцию, конструктор копирования. Перегрузка операторов. Статические переменные и методы класса. Константные методы. Перегруженные методы.			
5.	Наследование, виртуальные функции, полиморфизм.	Спецификаторы доступа public, protected, private. Понятие и механизм наследования классов. Преимущества и недостатки наследования по сравнению с композицией. Виртуальные функции, понятие полиморфизма. Чистые виртуальные функции и абстрактные классы. Доступ к объекту через указатель, преобразование типов. Непрямые базовые классы. Виртуальный деструктор. Множественное наследование.	36	14	[1-3]
6.	Знакомство с библиотекой STL.	Понятие объекта и класса. Классы стандартной библиотеки для работы со строками. Классы стандартной библиотеки для работы с контейнерами. Операции с объектами стандартной библиотеки. Преобразование типов.	28	14	[1-3, 8]
7.	Шаблоны C++.	Понятие обобщенного программирования. Определение шаблона класса. Создание объектов шаблона класса. Наследование шаблонов классов. Шаблоны функций.	28	14	[1-3]
8.	Знакомство с паттернами проектирования.	Понятие паттерна проектирования. Порождающие, структурные, поведенческие паттерны.	30	8	[4]
9.	Семейство паттернов MVC	Паттерн Модель – Представление – Контроллер (MVC). Вариации MVC: MVP, MVVM	10	4	[4]
10.	Элементы пользовательского интерфейса.	Проектирование пользовательского интерфейса.	12	6	[4]
11.	Архитектура документ-представление.	Архитектура SDI приложений. Архитектура MDI приложений. Различные представления одного документа. Уведомление представлений об изменении документа. Понятие сериализации, сохранение и печать документа.	14	4	[4]
12.	Алгоритмы на графах.	Построение пользовательских классов и использование популярных библиотек для работы с графами.	24	10	[11]
13.	Основы машинного обучения на C++.	Задачи обучения с учителем: задачи классификации, регрессии. Задачи обучения без учителя, кластеризации.	24	10	[12-13]
14.	Многопоточное программирование.	Создание потоков. Рабочие и интерфейсные потоки. Синхронизация потоков. Остановка потоков..	18	6	[1-3, 5, 6]

8 Образовательные технологии

Занятия по курсу проходят в форме лекций и семинарских занятий с регулярными домашними заданиями (составление и защита программного решения для выданной проблемы). Лекции должны проводиться в классах, обеспеченных компьютером и проекционным оборудованием. Студенты реализуют на семинарах и при выполнении домашних заданий с помощью



разобранных на лекциях и семинарах методов и подходов различные задачи по программированию на языке C++. Для достижения хороших результатов при изучении дисциплины студентам необходимо самостоятельно дома изучать рассмотренные на лекциях приемы программирования, решать задания, выданные преподавателем, а также разбирать материалы по соответствующим темам в рекомендованных учебниках и электронных ресурсах.

9 Оценочные средства для текущего контроля и аттестации студента

9.1. Вопросы для оценки качества освоения дисциплины

Примерный перечень вопросов к зачету (экзамену) по всему курсу или к каждому промежуточному и итоговому контролю для самопроверки студентов.

1. Классификация языков программирования. Парадигмы программирования.
2. Объектно-ориентированный подход. Основные понятия.
3. Принципы наследования и композиции при конструировании новых классов. Преимущества и недостатки наследования по сравнению с композицией.
4. Инициализация данных объекта класса (Конструктор по умолчанию, конструктор с параметрами). Назначение деструктора. Способы передачи объекта в функцию, конструктор копирования.
5. Перегрузка операторов.
6. Статические переменные и методы класса. Константные методы. Перегруженные методы.
7. Виртуальные функции, понятие полиморфизма. Чистые виртуальные функции и абстрактные классы. Виртуальный деструктор.
8. Множественное наследование.
9. Понятие обобщенного программирования. Шаблоны классов. Шаблоны функций.
10. Понятие исключения. Механизм генерации и обработки исключений.
11. Понятие паттернов проектирования. Структурные паттерны.
12. Понятие паттернов проектирования. Порождающие паттерны.
13. Понятие паттернов проектирования. Поведенческие паттерны.
14. Понятие паттернов проектирования. Семейство Модель-Представление-Контроллер.
15. Основные классы и методы, используемые в алгоритмах на графах.
16. Основные подходы и методы машинного обучения.

10 Учебно-методическое и информационное обеспечение дисциплины

10.1 Базовый учебник

1. Г. Шильдт. C++: руководство для начинающих, 2-е издание. : Пер. с англ. – М. : Издательский дом «Вильямс», 2005. – 672 с.

10.2 Основная литература

2. Б. Страуструп. Программирование и практика с использованием C++, 2-е издание. : Пер. с англ. – М. : ООО «И.Д. Вильямс», 2016. – 1328 с.
3. Б. Страуструп. Язык программирования C++. Специальное издание. Пер. с англ. – М.:Издательство Бином, 2011 г. – 1136 с.
4. Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влссидес. Приемы объектно-ориентированного проектирования. Паттерны проектирования – СПб: «Питер», 2007. – 366 с.



10.3 Дополнительная литература

5. Qt Documentation. [Электронный ресурс]. URL: <http://doc.qt.io/>
6. Русское Qt-сообщество. Программирование Qt. [Электронный ресурс]. URL: <http://qt-doc.ru/>
7. Л.В. Городняя. Парадигмы программирования: Курс лекций. – Новосибирск: НГУ, 2007.
8. М. Шлее, Qt 4.5. Профессиональное программирование на C++ – БХВ-Петербург, 2009, 896
9. Херн Д., Бейкер М., Компьютерная графика и стандарт OpenGL, Вильямс, 1168 с
10. С. Майерс. Эффективное использование STL. — СПб: Питер, 2002. — 224 с.
11. Boost Documentation. [Электронный ресурс]. <http://www.boost.org/doc/>
12. Dlib Documentation. [Электронный ресурс]. <http://dlib.net/>
13. Mpack Documentation. [Электронный ресурс]. <http://www.mpack.org/>

10.4 Справочники, словари, энциклопедии

Не предусмотрены.

10.5 Программные средства

Для успешного освоения дисциплины, студент использует следующие программные средства: язык программирования C++ стандарта C++11, доступный в том числе в Visual Studio 2015.

10.6 Дистанционная поддержка дисциплины

Не предусмотрена.

11 . Материально-техническое обеспечение дисциплины

Для проведения лекций необходима аудитория, оснащенная компьютером и проекционным оборудованием.

Для проведения семинарских занятий необходим компьютерный класс, оснащенный современным программным обеспечением, IDE для разработки программ: Visual Studio 2015 и выше, с установленными библиотеками C++.