

Программа учебной дисциплины «Алгоритмизация и программирование»

Утверждена

Академическим советом ООП
Протокол № 4 от «24» мая 2016 г.

Автор	Зонтов Юрий Владимирович Чеповский Александр Андреевич
Число кредитов	6
Контактная работа (час.)	102
Самостоятельная работа (час.)	126
Курс	2
Формат изучения дисциплины	Очная

I. ЦЕЛЬ, РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ И ПРЕРЕКВИЗИТЫ

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.

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.

Изучение данной дисциплины базируется на следующих дисциплинах:

- Математический анализ;
- Линейная алгебра и аналитическая геометрия;
- 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 должны быть использованы в дальнейшем при изучении следующих 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);

II. СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

№	Наименование раздела дисциплины	Содержание раздела	Аудиторная работа	Самостоятельная работа	Литература к разделу
1.	Парадигмы программирования. Знакомство с ООП.		2	10	[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.	15	10	[1-3]
3.	Обработка исключений.	Понятие исключения. Механизм генерации и обработки исключе-	11	10	[1-3]

		ний.			
4.	Построение пользовательских типов данных.	Создание пользовательских классов. Принцип композиции при конструировании новых классов. Инициализация данных объекта класса (Конструктор по умолчанию, конструктор с параметрами). Назначение деструктора. Способы передачи объекта в функцию, конструктор копирования. Перегрузка операторов. Статические переменные и методы класса. Константные методы. Перегруженные методы.	20	10	[1-3]
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.	20	10	[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.	19	10	[1-3, 8]
7.	Шаблоны C++.	Понятие обобщенного программирования. Определение шаблона класса. Создание объектов шаблона класса. Наследование шаблонных классов. Шаблоны функций.	15	10	[1-3]
8.	Знакомство с паттернами проектирования.	Понятие паттерна проектирования. Порождающие, структурные, поведенческие паттерны.	14	18	[4]
9.	Многопоточное программирование.	Создание потоков. Рабочие и интерфейсные потоки. Синхронизация потоков. Остановка потоков.	14	10	[1-3, 5, 6]

III. ОЦЕНИВАНИЕ

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 балльной шкале за работу на семинарских занятиях определяется перед промежуточным или итоговым контролем – *О_{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 пересдаче студенту не предоставляется возможность получить дополнительный балл для компенсации оценки за текущий контроль.

IV. ПРИМЕРЫ ОЦЕНОЧНЫХ СРЕДСТВ

Оценочные средства для оценки качества освоения дисциплины в ходе текущего контроля

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

Примеры заданий промежуточной аттестации

Примерный список вопросов к экзамену

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.

V. РЕСУРСЫ

V.1 Основная литература

1. Дейтел Х. М., Как программировать на С++, БИНОМ, 2008

V.2 Программное обеспечение

Язык программирования С++ стандарта С++11, доступный в том числе в Visual Studio 2015.

V.3 Профессиональные базы данных, информационные справочные системы, интернет-ресурсы (электронные образовательные ресурсы)

V.4 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.