

## Программа учебной дисциплины «Компьютерный практикум»

Утверждена

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

Автор	Зонтов Юрий Владимирович Чеповский Александр Андреевич
Число кредитов	2
Контактная работа (час.)	38
Самостоятельная работа (час.)	38
Курс	2
Формат изучения дисциплины	Очная

### I. ЦЕЛЬ, РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ И ПРЕРЕКВИЗИТЫ

Дисциплина «Компьютерный практикум» призвана обучить студентов основам объектно-ориентированного программирования:

- выработке навыков и умений программирования на языке C и C++;
- изучению основ построения алгоритмов и выработки умений эффективных реализаций различных алгоритмов.
- выработке навыков и умений программирования на языке C++ с использованием библиотеки STL;
- изучению основ объектно-ориентированного программирования;
- получению общего представления о широко используемых библиотеках C++.

Цели освоения дисциплины:

- **Знакомство** слушателей дисциплины с основами языка программирования C++, базовыми приемами работы в среде MS Visual Studio, концепциями, используемыми в популярных библиотеках, базовыми приемами машинного обучения.
- **Формирование практических навыков** объектно-ориентированного программирования, базовых принципов, применяемых при разработке и реализации алгоритмов, разработке и отладке компьютерных программ;
- **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.

Практические занятия по программированию подкрепляются теоретическим материалом курса «Алгоритмизация и программирование».

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:

- 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:

- знаниями основных определений и теорем, перечисленных выше дисциплин;
- навыками решения типовых задач этих дисциплин.

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.	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.	2	2	[1-3]
2.	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.	4	10	[1-3, 8]

3.	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.	4	10	[1-3]
4.	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.	4	10	[1-3]
5.	Шаблоны C++.	Понятие обобщенного программирования. Определение шаблона класса. Создание объектов шаблона класса. Наследование шаблонных классов. Шаблоны функций.	4	10	[1-3]
6.	Создание и работа с элементами пользовательского интерфейса. Знакомство с Qt	Проектирование пользовательского интерфейса в Qt.	4	12	[5, 6]
7	Алгоритмы на графах	Построение пользовательских классов и использование популярных библиотек для работы с графами.	4	10	
8.	Основы машинного обучения на C++.	Задачи обучения с учителем: задачи классификации, регрессии. Задачи обучения без учителя, кластеризации.	4	10	[12-13]
9.	Многопоточное программирование	Создание потоков. Рабочие и интерфэйсные потоки. Синхронизация потоков.	4	6	[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 –  $O_{\text{аудиторная}}$ .

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 –  $O_{\text{сам. работа}}$ .

Накопленная оценка за текущий контроль учитывает результаты студента по текущему контролю за соответствующие модули следующим образом:

$$O_{\text{накопленная}} = 0,5 \cdot O_{\text{аудиторная}} + 0,5 \cdot O_{\text{сам. работа}}$$

Результирующая оценка за текущий контроль в форме экзамена проставляется, по накопленной оценке,:

$$O_{\text{результ}} = O_{\text{накопленная}}$$

Способ округления: арифметический.

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

#### IV. ПРИМЕРЫ ОЦЕНОЧНЫХ СРЕДСТВ

##### Примеры заданий промежуточной аттестации

##### Примерный список вопросов к экзамену

1. Пространства имен. Ввод/вывод в C++.
2. Операции инкремента и декремента. Особенности операторов инкремента и декремента при работе с указателями.
3. Выделение/освобождение памяти в C++.
4. Классы стандартной библиотеки для работы со строками.
5. Классы стандартной библиотеки для работы с контейнерами.
6. Преобразование типов в C++.
7. Принципы наследования и композиции при конструировании новых классов. Преимущества и недостатки наследования по сравнению с композицией.
8. Инициализация данных объекта класса (Конструктор по умолчанию, конструктор с параметрами). Назначение деструктора. Способы передачи объекта в функцию, конструктор копирования.
9. Перегрузка операторов.
10. Статические переменные и методы класса. Константные методы. Перегруженные методы.
11. Виртуальные функции, понятие полиморфизма. Чистые виртуальные функции и абстрактные классы. Виртуальный деструктор.
12. Множественное наследование.
13. Понятие обобщенного программирования. Шаблоны классов. Шаблоны функций.
14. Макросы ASSERT, VERIFY и TRACE. Понятие исключения. Механизм генерации и обработки исключений.
15. Рисование в окне, инструменты GDI.
16. Основные классы архитектуры SDI и принцип их взаимодействия.
17. Основные классы архитектуры MDI и принципы их взаимодействия.
18. Понятие сериализации, сохранение и печать документа.
19. Создание и работа с меню. Использование ресурсов проекта.
20. Создание и работа с панелями инструментов. Использование ресурсов проекта.
21. Основные принципы работы с элементами управления (создание, размещение, получение уведомлений, изменение).
22. Основные классы и методы, используемые в алгоритмах на графах.
23. Основные подходы и методы машинного обучения.

## **V. РЕСУРСЫ**

### **V.1 Основная литература**

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

### **V.2 Программное обеспечение**

Для успешного освоения дисциплины студент использует следующие программные средства: язык программирования C++ стандарта C++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 семинарских занятий необходим компьютерный класс, оснащенный современным программным обеспечением, IDE для разработки программ: Visual Studio 2015 и выше, с установленными библиотеками C++.