

ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

КАФЕДРА КИБЕРНЕТИКИ

ОДОБРЕНО

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Направление подготовки
(специальность)

[1] 09.04.04 Программная инженерия

| Семестр | Трудоемкость, кред. | Общий объем курса, час. | Лекции, час. | Практич. занятия, час. | Лаборат. работы, час. | В форме практической подготовки/В СРС, час. | КСР, час. | Форма(ы) контроля, экз./зач./КСР/КП |
|---------|------------------------|----------------------------|--------------|---------------------------|--------------------------|--|-----------|---|
| 2 | 3 | 108 | 15 | 15 | 0 | 42 | 0 | Э |
| Итого | 3 | 108 | 15 | 15 | 0 | 42 | 0 | |

АННОТАЦИЯ

В рамках дисциплины изучаются принципы объектно-ориентированного подхода в разработке программ.

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Целями освоения учебной дисциплины «Объектно-ориентрованное программирование» являются изучение синтаксиса языка C++, понятий объектного и объектно-ориентированного подхода в разработке программ. Используются алгоритмы курса численных методов.

Учебная задача: Получить опыт разработки, отладки конкретных алгоритмов численных методов в их реализации на языке C++ с использованием методологии объектно-ориентированного программирования.

2. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Дисциплина «Объектно-ориентрованное программирование» является обязательной дисциплиной инженерной подготовки студента. Дисциплина не требует специальной начальной подготовки, выходящей за рамки курса математики и информатики программы среднего образования.

В свою очередь, дисциплина является необходимым дополнением следующих курсов, демонстрируя методы и средства выполнения инженерных расчетов, связанных с общими методами математических вычислений, методами оптимизации, методами моделирования и проектирования сложных систем.

- дискретная математика
- логическое и функциональное программирование
- практикум на ЭВМ
- Основы автоматизированных информационных технологий

3. ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Универсальные и(или) общепрофессиональные компетенции:

| | |
|--------------------------------|--|
| Код и наименование компетенции | Код и наименование индикатора достижения компетенции |
|--------------------------------|--|

Профессиональные компетенции в соответствии с задачами и объектами (областями знаний) профессиональной деятельности:

| Задача профессиональной деятельности (ЗПД) | Объект или область знания | Код и наименование профессиональной компетенции; Основание (профессиональный стандарт-ПС, анализ опыта) | Код и наименование индикатора достижения профессиональной компетенции |
|--|---------------------------|--|---|
| | | | |
| проектный | | | |

| | | | |
|---|--|---|---|
| <p>участие в проектировании компонентов программного продукта в объеме, достаточном для их конструирования в рамках поставленного задания, разработка и оформление эскизной, технической и рабочей проектной документации</p> | <p>программный продукт, процессы, методы и инструменты разработки программного продукта</p> | <p>ПК-12 [1] - способен проектировать вспомогательные и специализированные языки программирования и языки представления данных</p> <p><i>Основание:</i> Профессиональный стандарт: 06.028</p> | <p>З-ПК-12[1] - Знать: методы проектирования вспомогательных и специализированных языков программирования и языков представления данных ; У-ПК-12[1] - Уметь: проектировать вспомогательные и специализированные языки программирования и языки представления данных ; В-ПК-12[1] - Владеть: методами проектирования вспомогательных и специализированных языков программирования и языков представления данных</p> |
| <p>производственно-технологический</p> | | | |
| <p>разработка, тестирование и сопровождение программного обеспечения, применение методов и инструментальных средств управления инженерной деятельностью и процессами жизненного цикла программного обеспечения, взаимодействие с заказчиком в процессе выполнения</p> | <p>инструменты разработки программного продукта, процессы жизненного цикла программного продукта</p> | <p>ПК-19 [1] - способен применять навыки создания систем обработки текстов</p> <p><i>Основание:</i> Профессиональный стандарт: 06.028</p> | <p>З-ПК-19[1] - Знать: технологии создания систем обработки текстов ; У-ПК-19[1] - Уметь: применять навыки создания систем обработки текстов ; В-ПК-19[1] - Владеть: навыками создания систем обработки текстов</p> |

4. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Разделы учебной дисциплины, их объем, сроки изучения и формы контроля:

| № п.п | Наименование раздела учебной дисциплины | Недели | Лекции/ Практи. (семинары)/ Лабораторные работы, час. | Обязат. текущий контроль (форма*, неделя) | Максимальный балл за раздел** | Аттестация раздела (форма*, неделя) | Индикаторы освоения компетенции |
|-------|---|--------|--|---|-------------------------------|-------------------------------------|---|
| | <i>2 Семестр</i> | | | | | | |
| 1 | Элементы объектно-ориентированного программирования в современных языках программирования | 1-6 | 6/6/0 | к.р-6 () | 20 | КИ-6 | 3-ПК-12, У-ПК-12, В-ПК-12, 3-ПК-19, У-ПК-19, В-ПК-19 |
| 2 | Формализация ООП: Нетипизированные исчисления объектов | 7-8 | 2/2/0 | к.р-8 () | 30 | КИ-8 | 3-ПК-12, У-ПК-12, В-ПК-12, 3-ПК-19, У-ПК-19, В-ПК-19 |
| 3 | Формализация ООП: Типизированные исчисления объектов | 9-15 | 7/7/0 | к.р-15 () | 20 | КИ-15 | 3-ПК-12, У-ПК-12, В-ПК-12, 3-ПК-19, У- |

| | | | | | | | |
|--|---|--|---------|--|----|---|---|
| | | | | | | | ПК-19, В-ПК-19 |
| | <i>Итого за 2 Семестр</i> | | 15/15/0 | | 70 | | |
| | Контрольные мероприятия за 2 Семестр | | | | 30 | Э | 3-ПК-12, У-ПК-12, В-ПК-12, 3-ПК-19, У-ПК-19, В-ПК-19 |

* – сокращенное наименование формы контроля

** – сумма максимальных баллов должна быть равна 100 за семестр, включая зачет и (или) экзамен

Сокращение наименований форм текущего контроля и аттестации разделов:

| Обозначение | Полное наименование |
|-------------|---------------------|
| КИ | Контроль по итогам |
| к.р | Контрольная работа |
| Э | Экзамен |

КАЛЕНДАРНЫЙ ПЛАН

| Недели | Темы занятий / Содержание | Лек., час. | Пр./сем., час. | Лаб., час. |
|--------|--|------------------------|----------------|------------|
| | <i>2 Семестр</i> | 15 | 15 | 0 |
| 1-6 | Элементы объектно-ориентированного программирования в современных языках программирования | 6 | 6 | 0 |
| 1 - 2 | Основные понятия и механизмы программирования и языков С. С++. Специфика иерархического, объектного и объектно-ориентированного подходов в разработке программ. Схема построения приложений в объектно-ориентированной методологии проектирования. | Всего аудиторных часов | | |
| | | 2 | 2 | |
| | | Онлайн | | |
| 3 - 4 | Синтаксиса языка С++. Переход синтаксиса от С к С++. Синтаксис С++. Механизм защиты приложений. | Всего аудиторных часов | | |
| | | 2 | 2 | |

| | | | | |
|---------|---|------------------------|---|---|
| | Конструирование объектов. Описание функций. Наследование и виртуальные определения. | Онлайн | | |
| 5 - 6 | Элементы объектно-ориентированного программирования. Понятие и определение класса и его методов. Конструктор класса. Наследование. Перегрузка операций. Полиморфизм описаний. | Всего аудиторных часов | | |
| | | 2 | 2 | |
| | | Онлайн | | |
| 7-8 | Формализация ООП: Нетипизированные исчисления объектов | 2 | 2 | 0 |
| 7 - 8 | Использование шаблонов при описании объектов. Понятие «шаблона» и его использование при программировании. Абстракции данных и процедур. | Всего аудиторных часов | | |
| | | 2 | 2 | |
| | | Онлайн | | |
| 9-15 | Формализация ООП: Типизированные исчисления объектов | 7 | 7 | 0 |
| 9 - 11 | Обработка исключений. Синтаксис обработки ситуаций обработки исключений. Примеры описания и использования. | Всего аудиторных часов | | |
| | | 3 | 3 | |
| | | Онлайн | | |
| 12 - 15 | Разбор конкретных примеров использования методологии объектно-ориентированного программирования Разбор конкретных примеров использования методологии объектно-ориентированного программирования | Всего аудиторных часов | | |
| | | 4 | 4 | |
| | | Онлайн | | |

Сокращенные наименования онлайн опций:

| Обозначение | Полное наименование |
|-------------|----------------------------------|
| ЭК | Электронный курс |
| ПМ | Полнотекстовый материал |
| ПЛ | Полнотекстовые лекции |
| ВМ | Видео-материалы |
| АМ | Аудио-материалы |
| Прз | Презентации |
| Т | Тесты |
| ЭСМ | Электронные справочные материалы |
| ИС | Интерактивный сайт |

ТЕМЫ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

| Недели | Темы занятий / Содержание |
|--------|---|
| | <i>2 Семестр</i> |
| 1 - 6 | Элементы объектно-ориентированного программирования Элементы объектно-ориентированного программирования |
| 7 - 8 | Использование шаблонов при описании объектов Использование шаблонов при описании объектов |
| 9 - 17 | Разбор конкретных примеров использования методологии объектно-ориентированного программирования |

| |
|---|
| Разбор конкретных примеров использования методологии объектно-ориентированного программирования |
|---|

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Практические занятия проводятся по традиционной схеме, на отдельных практических занятиях студенты выступают с докладами и проводят их обсуждение. На практических занятиях регулярно проводится контрольный опрос для проверки освоения студентами теоретического материала, предусмотрена контрольная работа в компьютерном классе.

6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий текущего, рубежного и промежуточного контроля по дисциплине.

Связь между формируемыми компетенциями и формами контроля их освоения представлена в следующей таблице:

| Компетенция | Индикаторы освоения | Аттестационное мероприятие (КП 1) |
|-------------|---------------------|-----------------------------------|
| ПК-12 | З-ПК-12 | Э, КИ-6, КИ-8, КИ-15 |
| | У-ПК-12 | Э, КИ-6, КИ-8, КИ-15 |
| | В-ПК-12 | Э, КИ-6, КИ-8, КИ-15 |
| ПК-19 | З-ПК-19 | Э, КИ-6, КИ-8, КИ-15 |
| | У-ПК-19 | Э, КИ-6, КИ-8, КИ-15 |
| | В-ПК-19 | Э, КИ-6, КИ-8, КИ-15 |

Шкалы оценки образовательных достижений

Шкала каждого контрольного мероприятия лежит в пределах от 0 до установленного максимального балла включительно. Итоговая аттестация по дисциплине оценивается по 100-балльной шкале и представляет собой сумму баллов, заработанных студентом при выполнении заданий в рамках текущего и промежуточного контроля.

Итоговая оценка выставляется в соответствии со следующей шкалой:

| Сумма баллов | Оценка по 4-ех балльной шкале | Оценка ECTS | Требования к уровню освоению учебной дисциплины |
|--------------|-------------------------------|-------------|---|
| 90-100 | 5 – «отлично» | A | Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, использует в ответе материал монографической литературы. |
| 85-89 | 4 – «хорошо» | B | Оценка «хорошо» выставляется студенту, если он твёрдо знает |
| 75-84 | | C | |

| | | | |
|---------|------------------------------|---|---|
| 70-74 | | D | материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос. |
| 65-69 | 3 – «удовлетворительно» | E | Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала. |
| 60-64 | | | |
| Ниже 60 | 2 – «неудовлетворительно» | F | Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине. |

Оценочные средства приведены в Приложении.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОСНОВНАЯ ЛИТЕРАТУРА:

1. ЭИ Б 87 Вопросы и упражнения на понимание физики : учебное пособие, Санкт-Петербург: Лань, 2018
2. ЭИ П 30 Моделирование процессов и систем : учебное пособие, Санкт-Петербург: Лань, 2015

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА:

1. 004 С83 Программирование. Принципы и практика использования С++ : , Москва [и др.]: Вильямс, 2011

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ:

Специальное программное обеспечение не требуется

LMS И ИНТЕРНЕТ-РЕСУРСЫ:

<https://online.mephi.ru/>

<http://library.mephi.ru/>

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Специальное материально-техническое обеспечение не требуется

9. УЧЕБНО-МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ СТУДЕНТОВ

Основное назначение данного раздела - выработка у студентов навыков правильного оформления документации по проекту, в данном случае, тем задачам, которые входят в объем задания практикума. Это стимулирует:

- самостоятельное изучение языков программирования,
- ясное представление и соответствующее оформление постановки решаемой задачи,
- четкий выбор и обоснование изобразительных средств используемого инструментального языка про-граммирования.

Каждому студенту предлагается набор из 3 - 4 задач общего задания, по которым он должен предста-вить помимо программной реализации на выбранных (или указанных преподавателем) языках (Паскаль, Модула-2, Си, С++) полную проектную документацию.

Первым этапом подготовки документации является разработка достаточно подробной спецификации для каждой выбранной задачи. Спецификация оформляется и согласуется с ведущим преподавателем до начала программирования (кодирования) и не должна зависеть от особенностей языка реализации. Допускается на данном этапе производить оформление основной логической схемы реализации в виде текста на псевдокоде.

Спецификация должна включать в себя требования к проверке правильности функционирования программной реализации. Требования должны покрывать все основные функции программной реализации (ввод, обработку ошибок, вычисление результата и т.п.).

Вторым этапом решения задачи является составление программы (программ) на языке (языках) реализации. При этом основное внимание должно быть уделено точности соблюдения требований спецификации и обоснованию используемых средств инструментального языка.

Если в процессе реализации произведены уточнение и(или) декомпозиция требований по проверке правильности функционирования программной реализации, то рекомендуется оформить это в виде дополни-тельных требований по проверке реализации.

Дальнейшим этапом является составление тест-плана - программы испытаний (проверки, тестирова-ния) программной реализации. Критерием качества тест-плана является покрытие (выполнение) всех тре-бований к проверке правильности функционирования программной реализации. Желательной характери-стикой тест-плана является проверка исполнения всех веток схемы программной реализации. Тест-план должен быть оформлен и согласован с ведущим преподавателем в процессе исполнения программной реализации задачи.

Заключительным этапом работы является исполнение тест-плана, и предъявление результатов преподавателю. При приеме (защите результатов) домашнего задания преподаватель должен проверить степень обоснованности проектных решений, принятых при программной реализации задачи. С этой целью рекомендуется рассмотреть и попросить студента обосновать предпочтение выбора из нескольких вариантов представления данных, возможных альтернатив организации вычислительного процесса и конструкций языка реализации.

Далее приведена рекомендуемая структура оформления спецификации задачи.

Задание

1. Постановка задачи

Написать программу, выполняющую кодирование входной строки перестановкой символов по правилу 2-3-1.

Первый этап работы.

Необходимо представить поведение программы в целом. Необходимо ли получить программу кодирования одной единственной строки или требуется диалоговый режим управления кодированием группы строк, каждая из которых вводится после некоторого приглашающего сообщения?

Можно предположить, что из общих соображений простоты общения и проверки работоспособности второй вариант (циклическая программа) представляется предпочтительным. Соответственно, первый раздел требований (спецификаций) может быть сформулирован следующим образом:

1. Необходимо разработать программу, выполняющую следующие функции:

1.1. Ввод строки символов с клавиатуры.

1.2. Преобразование входной строки путем перестановки символов в порядке 2-3-1.

1.3. Вывод преобразованной строки на экран.

1.4. Программа должна обрабатывать строки последовательно, каждый раз предупреждая пользователя о необходимости очередного ввода. Обработка должна быть закончена при вводе пользователем пустой строки.

Второй этап работы.

Необходимо принять решения по основным позициям внешнего проекта программы.

Что такое входная строка?

(Последовательность символов, ограниченная входным алфавитом.)

Существенно ли деление строки на слова?

(Пробел, как специальный символ алфавита разделитель/ограничитель слов.)

Может ли в строку (в слово) быть включен символ не имеющий адекватного представления на клавиатуре или символ-разделитель, как часть слова?

Существует ли ограничение на длину строки?

Как определить, что входная строка введена полностью?

(Ограничитель строки и признак конца ввода.)

Как выходная (преобразованная) строка сопоставляется с входной?

(Формат вывода, размещение строки на экране.)

Как производится преобразование при длине входной строки не кратной трем?

Каждый автор может принять собственное решение по данным пунктам. В любом случае свои решения в рамках домашнего практикума необходимо согласовать с преподавателем.

В рассматриваемом далее варианте внешнего проекта предлагается несколько упрощенный вариант требований, предполагающий разбику строки на слова и выполнение заданных правил перестановки только в пределах отдельного слова. При этом не накладывается сложных ограничений на разделитель слов. Им может быть многоточие или произвольная последовательность точек, запятых и пробелов.

2. Внешний проект

2.1. Входные данные программы.

2.1.1. Входная строка должна вводиться с клавиатуры после вывода на экран приглашающего сообщения.

2.1.2. Входная строка представляет собой последовательность слов, разделенных (ограниченных) символами разделителями ",", ".", " " (запятая, точка, пробел). Признаком конца ввода строки с клавиатуры является нажатие клавиши <Enter>. Длина строки не должна превышать строки экрана (80 символов).

2.1.3. Слово входной строки может состоять из букв (русского и латинского алфавита) и/или цифр.

2.1.4. Первому слову в строке может предшествовать последовательность пробелов.

2.1.5. Признаком конца слова является любой символ ограничитель (см. 2.1.2). Последнее слово строки может не иметь за собой ограничителя, если его последний символ есть последний символ строки.

2.1.6. Ввод пустой строки является указанием пользователя на необходимость закончить работу программы.

2.2. Выходные данные программы.

2.2.1 Сообщения программы.

Сообщения программы должны выводиться с новой строки. Курсор после вывода устанавливается на новую строку.

2.2.1.1. "Введите строку для кодирования" - приглашающее сообщение программы. Указывает на необходимость ввода с клавиатуры входной строки.

2.2.1.2. "Работа закончена" - завершающее сообщение программы. Должно выводиться после ввода пустой строки. Работа программы заканчивается.

2.2.1.3. "Ошибка во входной строке" - информационное сообщение программы. Должно выводиться после ввода входной строки, не соответствующей пункту 2.1. Работа программы должна быть продолжена приглашением ввода новой строки.

2.2.2. Выходная строка - результат кодирования.

2.2.2.1. Выходная строка представляет собой последовательность преобразованных слов, организованную (в смысле разделителей и ограничителей) идентично входной строке.

2.2.2.2. Преобразованные слова выходной строки должны получаться из соответствующих слов входной строки применением к ним процедуры кодирования.

2.2.2.3. При выводе на экран выходная строка должна располагаться строго под входной (разделитель под разделителем, слово под словом).

Автор(ы):

Шапкин Павел Александрович, к.т.н.