

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

УТВЕРЖДАЮ
Декан факультета
вычислительной техники




(Подпись)

Фионова Л.Р.
(Фамилия, Инициалы)

« _____ » _____ 2015 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.2.21.2 Высокоуровневые методы программирования

Направление подготовки: 01.03.02 Прикладная математика и информатика

Профиль подготовки: Системное программирование и компьютерные технологии

Квалификация (степень) выпускника – Бакалавр

Форма обучения: Очная

Пенза – 2015

1. Цели освоения дисциплины

Целью освоения дисциплины «Высокоуровневые методы программирования» является формирование и развитие у студентов общекультурных и профессиональных компетенций, формирование систематизированных знаний, умений и навыков в области прикладной математики и информатики, позволяющих подготовить конкурентоспособного выпускника для соответствующей сферы, готового к инновационной творческой реализации в области системного программирования и компьютерных технологий.

Задачи изучаемой дисциплины:

Исходя из общих целей подготовки бакалавра прикладной математики и информатики по профилю «Системное программирование и компьютерные технологии»:

- содействовать средствами дисциплины «Высокоуровневые методы программирования» развитию у студентов мотивации к профессиональной деятельности, творческого мышления, коммуникативной готовности, общей культуры;
- научить студентов ясно, точно, грамотно излагать свои мысли в устной и письменной речи.

Исходя из конкретного содержания дисциплины:

- ознакомление студентов с технологией разработки программ в современной среде программирования;
- формирование у студентов систематизированных знаний и практических навыков в области объектно-ориентированного программирования;
- отработать твердые практические навыки по использованию объектно-ориентированных языков и соответствующих систем программирования для разработки программного обеспечения информационных систем.

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина «Высокоуровневые методы программирования» относится к дисциплинам по выбору вариативной части учебного плана.

Для освоения дисциплины студенты используют знания, умения, навыки, сформированные в процессе изучения предмета «Языки и методы программирования».

Освоение данной дисциплины является основой для изучения дисциплины «Методы интеллектуального анализа данных», «Построение и администрирование компьютерных сетей», последующего прохождения практики, подготовки к итоговой государственной аттестации.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины «Объектно-ориентированные языки и системы программирования»

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО по данному направлению:

| Коды компетенции | Наименование компетенции | Структурные элементы компетенции (в результате освоения дисциплины обучающийся должен знать, уметь, владеть) |
|------------------|--|--|
| ОПК-1 | способность использовать базовые знания естественных наук, математики и информатики, основные факты, концепции, принципы теорий, связанные с прикладной математикой и информатикой | Знать: основные концепции и факты математики, которые используются при программном решении задач |
| | | Уметь: применять математические знания при программировании |
| | | Владеть: основными методами объектно-ориентированного программирования |
| ОПК-2 | способность приобретать новые научные и профессиональные знания, используя современные образовательные и информационные технологии; | Знать: о способах приобретения новой информации в области объектно-ориентированного проектирования программного обеспечения с помощью унифицированного языка моделирования UML. |
| | | Уметь: использовать возможности информационной среды в сфере сбора и представления новой информации для решения задач в области объектно-ориентированного проектирования и анализа программного обеспечения |
| | | Владеть: методами приобретения новой научной информации в области проектирования и анализа программного обеспечения с помощью унифицированного языка моделирования UML. |

| | | |
|-------|---|--|
| ОПК-3 | способностью к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям | Знать: сущность и значение понятия алгоритма, его применения в области анализа и проектирования программного обеспечения с помощью объектно-ориентированного подхода. |
| | | Уметь: использовать объектно-ориентированный подход для создания программного обеспечения |
| | | Владеть: основными методами создания программного обеспечения с использованием объектно-ориентированного подхода. |
| ОПК-4 | способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности. | Знать: основные стандартные задачи создания программного обеспечения на основе объектно-ориентированного подхода. |
| | | Уметь: использовать современные библиотеки алгоритмов, современные среды программирования. |
| | | Владеть: основными алгоритмическими и системными решениями в области системного и прикладного ПО |
| ПК-7 | способность к разработке и применению алгоритмических и программных решений в области системного и прикладного ПО | Знать: основные направления современных научных исследований в области объектно-ориентированного проектирования программного обеспечения. |
| | | Уметь: использовать данные современных научных исследований, необходимые для формирования выводов в области методологии объектно-ориентированного проектирования |
| | | Владеть: основными алгоритмическими и системными решениями в области системного и прикладного ПО |

| | | |
|-------|--|---|
| ПСК-1 | Способностью к формализации и алгоритмизации поставленных задач | <p>Знать: основные этапы решения задач с помощью ЭВМ.</p> <p>Владеть: современными способами решения задач с помощью ЭВМ.</p> <p>Уметь: применять полученные знания для решения конкретных задач</p> |
| ПСК-2 | Способностью к написанию программного кода с использованием языков программирования, определения и манипулирования данными | <p>Знать: основные конструкции языка C++.</p> <p>Владеть: современными средствами создания программного кода.</p> <p>Уметь: создавать приложения в среде MS Visual Studio</p> |

4. Структура и содержание дисциплины «Объектно-ориентированные языки и системы программирования»

4.1. Структура дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единиц, 144 часов.

| № п/п | Наименование разделов и тем дисциплины | Семестр | Недели семестра | Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах) | | | | | | | Формы текущего контроля успеваемости (по неделям семестра) | | |
|-------|---|---------|-----------------|--|--------|----------------------|------------------------|----------------------------------|---|---------------------------------------|--|---------------------------------------|--|
| | | | | Аудиторная работа | | | Самостоятельная работа | | | | Опрос на лабораторном занятии | Проверка выполнения курсового проекта | Проверка отчёта выполнения лабораторной работы |
| | | | | Всего | Лекция | Лабораторные занятия | Всего | Подготовка к аудиторным занятиям | Подготовка к выполнению курсового проекта | Подготовка отчёта лабораторной работы | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 1 | Раздел 1. Понятие о высокоуровневых методах программирования | | 1 | 4 | 2 | 2 | 8 | 3 | 3 | 2 | | | |
| 1.1 | Стили программирования, основные принципы, история развития. | 3 | 1 | 4 | 2 | 2 | 8 | 3 | 3 | 2 | 1 | 1 | 1 |
| 2 | Раздел 2. Базовые средства языка C++ | 3 | 2-5 | 16 | 8 | 8 | 18 | 6 | 6 | 6 | | | |
| 2.1 | Базовые конструкции структурного программирования | 3 | 2 | 4 | 2 | 2 | 6 | 2 | 2 | 2 | 1 | 1 | 1 |
| 2.2 | Работа с типами данных | 3 | 3-4 | | 4 | 4 | 6 | 2 | 2 | 2 | 1 | 1 | 1 |
| 2.3 | Модульное программирование | 3 | 5 | | 2 | 2 | 6 | 2 | 2 | 2 | 1 | 1 | 1 |

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|-----|---|---|--------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|--------------------------|---------|----|
| 3 | Раздел 3. Базовые понятия и основные свойства объектно-ориентированного программирования | 3 | 6-13 | 32 | 16 | 16 | 32 | 8 | 16 | 8 | | | |
| 3.1 | Сущность объектно-ориентированного подхода: объектный тип данных, переменные объектного тип. | 3 | 6-7 | 8 | 4 | 4 | 7 | 2 | 4 | 2 | 1 | 1 | 1 |
| 3.2 | Наследование | 3 | 8-9 | 8 | 4 | 4 | 7 | 2 | 4 | 2 | 1 | 1 | 1 |
| 3.3 | Шаблоны классов | 3 | 10-11 | 8 | 4 | 4 | 7 | 2 | 4 | 2 | 1 | 1 | 1 |
| 3.4 | Обработка исключительных ситуаций | 3 | 12-13 | 8 | 4 | 4 | 7 | 2 | 4 | 2 | 1 | 1 | 1 |
| 4 | Раздел 4. Объектно-ориентированный анализ. Статические UML-диаграммы | 3 | 14 | 4 | 2 | 2 | 6 | 2 | 2 | 2 | | | |
| 4.1 | Разработка UML-диаграмм объектной области | 3 | 14 | 4 | 2 | 2 | 6 | 2 | 2 | 2 | 1 | 1 | 1 |
| 5 | Раздел 5. Особенности разработки программ в объектно-ориентированных системах программирования | 3 | 15-18 | 16 | 8 | 8 | 12 | 4 | 4 | 2 | | | |
| 5.1 | Состав и характеристики объектно-ориентированных систем программирования | 3 | 15-16 | 8 | 4 | 4 | 6 | 2 | 2 | 2 | 1 | 1 | 1 |
| 5.2 | Управление проектом и подготовка среды визуального программирования | 3 | 17-18 | 8 | 4 | 4 | 6 | 2 | 2 | 2 | 1 | 1 | 1 |
| | Общая трудоемкость, в часах | | | | | | | | | | Промежуточная аттестация | | |
| | | | 144 | 72 | 36 | 36 | 72 | 21 | 29 | 22 | Форма | Семестр | |
| | | | | | | | | | | | Зачет | 3 | |
| | | | | | | | | | | | Экзамен | | |

4.2. Содержание дисциплины

Раздел 1. Понятие о высокоуровневых методах программирования

Тема 1.1. Стили программирования, основные принципы, история развития.

Объектно-ориентированное и структурное программирование: данные и функции, моделирование и кодирование. Эволюция языков и стилей программирования. Объектно-ориентированные языки программирования. Преимущества объектно-ориентированного программирования. Общение в группе разработчиков. Многократное использование.

Раздел 2. Базовые средства языка C++

Тема 2.1. Базовые конструкции структурного программирования

Алфавит языка, простые типы данных, переменные и выражения, операции. Оператор присвоения, ветвления, цикла, передачи управления.

Тема 2.2. Работа с типами данных

Указатели, массивы, типы данных, определяемые пользователем.

Тема 2.3. Модульное программирование

Объявление и определение функции, глобальные переменные, возвращение значений, параметры функции. Рекурсивные функции, шаблоны функций. Функция main().

Раздел 3. Базовые понятия и основные свойства объектно-ориентированного программирования

Тема 3.1. Сущность объектно-ориентированного подхода: объектный тип данных, переменные объектного типа.

Понятие объекта и класса. Классы и методы в языке C++. Конструктор. Статические элементы класса. Деструктор.

Тема 3.2. Наследование

Ключ доступа. Простое наследование. Виртуальные методы. Множественное наследование.

Тема 3.3. Шаблоны классов

Создание шаблонов классов. Использование шаблонов классов. Специализация шаблонов классов.

Тема 3.4. Обработка исключительных ситуаций

Общий механизм обработки исключений. Синтаксис исключений. Перехват исключений. Список исключений функции. Исключения в конструкторах и деструкторах. Иерархии исключений.

Раздел 4. Объектно-ориентированный анализ. Статические UML-диаграммы

Тема 4.1. Разработка UML-диаграмм объектной области

Переход от сбора требований к анализу. Статическая и динамическая модель системы. Методики идентификации объектов. Диаграмма классов. Актер и приложение как объекты. Управление выполнением. Классы проблемной области и классы-утилиты.

Раздел 5. Особенности разработки программ в объектно-ориентированных системах программирования

Тема 5.1. Состав и характеристики объектно-ориентированных систем программирования

Особенности программирования в оконных операционных средах. Среда визуального построителя приложений. Настраиваемая среда разработчика. Основные стандартные модули, обеспечивающие работу в оконной операционной среде: система окон разработки, система меню. Библиотека объектных Визуальных Компонент. Формы, модули и методы. Делегирование и ссылки на классы. Размещение нового компонента. Обработка исключительных ситуаций.

Тема 5.2. Управление проектом и подготовка среды визуального программирования к работе

Основы визуального программирования. Состав проекта и назначение его файлов. Формирование проекта. Конфигурация среды программирования (IDE). Реакция на события. Отладка и тестирование программ.

5. Образовательные технологии

В ходе освоения дисциплины «Высокоуровневые методы программирования», при проведении аудиторных занятий, используются технологии традиционных и нетрадиционных учебных занятий.

Технология традиционного обучения предусматривает такие методы и формы изучения материала как лекция, лабораторные занятия.

- Проведение интерактивной лекции, демонстрирующей работу по конструированию объектов (Тема 3.1. Сущность объектно-ориентированного подхода: объектный тип данных, переменные объектного типа).
- Проведение проблемной лекции (Тема 5.1. Состав и характеристики объектно-ориентированных систем программирования).

При изучении дисциплины «Высокоуровневые методы программирования» используются активные и интерактивные технологии обучения, такие как:

- Проведение деловой игры (Тема 1.1.Стили программирования, основные принципы, история развития).
- кейс-технология (Тема 3.4.Обработка исключительных ситуаций).

Занятия, проводимые в интерактивной форме, в том числе с использованием интерактивных технологий составляют 25% от общего количества аудиторных занятий.

Самостоятельная работа студентов подразумевает работу под руководством преподавателя (консультации, коллоквиумы) и индивидуальную работу студента, выполняемую, в том числе, в компьютерном классе с выходом в сеть «Интернет» на физико-математическом факультете университета.

При реализации образовательных технологий используются следующие виды самостоятельной работы:

- работа с конспектом лекции;
- подготовка к лабораторной работе;
- подготовка доклада по заданной теме с компьютерной презентацией;
- поиск информации в сети «Интернет» и дополнительной и справочной литературе.

В целях реализации индивидуального подхода к обучению студентов, осуществляющих учебный процесс по собственной траектории в рамках индивидуального рабочего плана, изучение данной дисциплины базируется на следующих возможностях: обеспечение внеаудиторной работы со студентами в том числе в электронной образовательной среде с использованием соответствующего программного оборудования, дистанционных форм обучения, возможностей интернет-ресурсов, индивидуальных консультаций и т.д.

| | | | |
|--------------------|---|--------------|----------------------------|
| <p>2.2.</p> | <ul style="list-style-type: none"> • <i>Подготовка к аудиторному занятию: работа с конспектом лекции;</i> <p>Проанализировать возможность работы с массивами, указателями и строками в современной среде программирования.</p> <ul style="list-style-type: none"> • <i>Подготовка к лабораторной работе</i> <p>Решить следующие задачи:</p> <p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <ol style="list-style-type: none"> 1) сумму отрицательных элементов массива; 2) произведение элементов массива, расположенных между максимальным и минимальным элементами. 3) Упорядочить элементы массива по возрастанию. <ul style="list-style-type: none"> • <i>Подготовка к выполнению курсового проекта</i> <p>Дополнительно решить следующие задачи:</p> <ol style="list-style-type: none"> 1. Поменять местами наибольший и наименьший элементы. 2. Найти сумму элементов массива, находящимися между наибольшим и наименьшим. 3. В массиве X из 50 элементов найти наименьший элемент и заменить его значением суммы всех элементов, предшествующих ему. 4. Задан массив X из 30 элементов. Поместить в массив Y все отрицательные элементы массива X, сменив знак чисел, в массив Z - все положительные элементы массива X (в порядке их следования). Подсчитать количество нулевых элементов. <p>Выполнение индивидуального задания Подготовка отчета</p> | <p>3,4,5</p> | <p>2</p> <p>5</p> <p>2</p> |
| <p>2.3.</p> | <ul style="list-style-type: none"> • <i>Подготовка к аудиторному занятию: работа с конспектом лекции;</i> <p>Рассмотреть возможности работы с функциями в современной среде программирования.</p> <ul style="list-style-type: none"> • <i>Подготовка к выполнению курсового проекта</i> <p>Решить следующие задачи:</p> <ol style="list-style-type: none"> 1. Написать функцию, которая записывает в бинарный файл данные о сканере из приведенной структуры. Структура файла: в первых четырех байтах размещается значение типа long, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах. 2. Написать функцию, которая сортирует записи в описанном выше бинарном файле по наименованию модели сканера. Привести пример программы, создающей файл с данными о сканерах (данные вводятся с клавиатуры) из не менее восьми записей и осуществляющий его сортировку. <p>Дополнительно выполнить задание:</p> <p>С помощью текстового редактора создать файл, содержащий текст, длина которого не превышает 1000 символов (длина строки текста не должна превышать 70 символов).</p> <p>Имя файла должен иметь расширение DAT.</p> <p>Написать программу, которая:</p> <ul style="list-style-type: none"> • выводит текст на экран дисплея; • по нажатию произвольной клавиши поочередно выделяет каждое предложение текста; | <p>1,2,3</p> | <p>2</p> |

| | | | |
|--------------------|---|----------------|----------------------------|
| <p>3.4.</p> | <ul style="list-style-type: none"> • <i>Подготовка к аудиторному занятию: работа с конспектом лекции;</i> <p>Проанализировать возможность использования механизма обработки исключений при разработке программ.</p> <ul style="list-style-type: none"> • <i>Подготовка к лабораторной работе</i> <p>Решить задачи: Написать функцию, генерирующую исключение в случае ошибки: $f(a,b,c)$- вычисляет площадь треугольника по трем сторонам.</p> <ul style="list-style-type: none"> • <i>Подготовка к выполнению курсового проекта</i> <p>Дополнительно решить задачи: Написать функцию, генерирующую исключение в случае ошибки: функция вычисляет день недели по дате. Дата представляется структурой с тремя полями: год, месяц, день. Первое января считается понедельником.</p> <p>Выполнение индивидуального задания</p> <p>Подготовка отчета</p> | <p>1,3,4</p> | <p>2</p> <p>8</p> <p>2</p> |
| <p>4.1.</p> | <ul style="list-style-type: none"> • <i>Подготовка к аудиторному занятию: работа с конспектом лекции;</i> <p>Проанализировать возможность использования UML-диаграмм при конструировании классов.</p> <ul style="list-style-type: none"> • <i>Подготовка к лабораторной работе</i> <p>Решить задачи: Разработать диаграмму классов «Заместитель».</p> <ul style="list-style-type: none"> • <i>Подготовка к зачету</i> <p>Дополнительно рассмотреть вопросы:</p> <ol style="list-style-type: none"> 1. Наследование. 2. Полиморфизм. 3. Объекты и их классы. 4. Конструкторы и деструкторы. 5. Шаблоны классов. 6. Обработка исключительных ситуаций. 7. Среда разработки объектно-ориентированных программ. 8. Стандартные модули оконной операционной среды. 9. Система окон среды разработки объектно-ориентированных программ. 10. Система меню среды разработки объектно-ориентированных программ. 11. Основы визуального программирования. <ul style="list-style-type: none"> • <i>Подготовка к выполнению курсового проекта</i> <p>Рассмотреть создание проекта по одной из тем курсовой работы: Разработать приложения показа в форме текущего времени в виде обычных стрелочных часов со стрелками часов, минут и секунд. Предусмотреть возможность выбора часового пояса (относительно московского времени).</p> <p>Выполнение индивидуального задания</p> <p>Подготовка отчета</p> | <p>1,2,3,4</p> | <p>2</p> <p>5</p> <p>2</p> |

| | | | |
|------|---|-------|-----------------------------|
| 5.1. | <ul style="list-style-type: none"> • <i>Подготовка к аудиторному занятию: работа с конспектом лекции;</i> Проанализировать возможность работы в современной среде программирования Microsoft Visual Studio. • <i>Подготовка к лабораторной работе</i> Разработать консольное приложение, вычисляющее сумму чисел, передаваемых ему в виде аргументов командной строки. Выполнение индивидуального задания Подготовка отчета | 3,5 | 2 4 2 |
| 5.2. | <ul style="list-style-type: none"> • <i>Подготовка к аудиторному занятию: работа с конспектом лекции;</i> Проанализировать возможности создания визуальных приложений в Microsoft Visual Studio. • <i>Подготовка к лабораторной работе</i> Разработать приложение, вычисляющее сумму чисел, задаваемых пользователем в окне редактора. Выполнение индивидуального задания Подготовка отчета | 1,2,5 | 3 2 |

6.2. Методические указания по организации самостоятельной работы студентов

Планируются следующие виды самостоятельной работы:

- подготовка к лабораторным занятиям,
- выполнение индивидуального задания,
- оформление отчётов по лабораторным работам,
- подготовка к зачету,
- подготовка к выполнению курсового проекта,
- работа с конспектом лекций и изучение литературы при подготовке к лабораторным ра-

ботам.

6.3. Материалы для проведения текущего и промежуточного контроля знаний студентов

Контроль освоения компетенций

| № п\п | Вид контроля | Контролируемые темы (разделы) | Компетенции, компоненты которых контролируются |
|-------|---|-------------------------------|--|
| 1 | Текущий: проверка выполнения индивидуального задания на лабораторной работе | Разделы 1 – 5 | ОПК-1,2,3,4; ПК-7; ПСК 1,2 |
| 2 | Текущий: собеседование при защите лабораторных работ | Разделы 1 – 5 | ОПК-1,2,3,4; ПК-7; ПСК 1,2 |

Средства текущего контроля

Текущий контроль выполняется на лабораторных занятиях при защите выполненных студентом лабораторных работ. Защита предполагает представление отчета с обоснованием примененных решений выполнения лабораторного задания, ответы на контрольные вопросы преподавателя.

Примерный вариант теста

1. Объектно-ориентированный подход не отличается от процедурной разработки приложений.
Это всего лишь новый язык программирования
 - Да
 - Нет
2. Назовите преимущества объектно-ориентированного программирования
 - Устойчивость к изменениям
 - Меньшие размеры исполняемого файла
 - Более дешевые инструментальные средства
 - Все вышеперечисленные
3. Инкапсуляция – это
 - Скрытие реализации
 - Главное преимущество объектной технологии
 - Способ заставить Вас забыть, как реализован объект
 - Все вышеперечисленные
4. Объектно-ориентированное программирование не поддерживает многократное использование посредством
 - Копирования кода из одного объекта и вставки в другой
 - Использования уже написанных объектов
 - Разрешения нескольким объектам использовать одинаковые имена методов
 - Разрешения двум объектам использовать метод, описанный в третьем

Примерный перечень тем для курсового проектирования

1. Создать клавиатурный тренажер, позволяющий загружать задания из текстового файла, сохранять результат. Для отображения клавиш использовать компонент SpeedButton.
2. Создать программу хранения файлов e-mail. Предусмотреть поля почтового сообщения «От», «Кому», «Копия», «Скрытая копия», «Тема», «Сообщение», «Вложения» (пути к файлам). Данные хранятся в результирующем файле.
3. Создать проект «Составитель кроссвордов». Создается поле из клеток, в каждую из которых, если необходимо, может быть вписана буква. Цвет клеток можно менять на серый. Т.е. Пользователь сам может сформировать кроссворд, вводя нужные буквы в различные квад-

ратики. Справа от сетки есть колонка, куда можно было записывать описание к словам кроссворда.

4. Создать проект «Тамагоччи». Пользователь указывает начальные значения параметров: Сытость, Здоровье, Настроение, Чистота. Они убавляются каждые 10 секунд. Есть возможность пополнения параметров. В зависимости от увеличения или уменьшения параметров изменяется картинка, символизирующая питомца-тамагоччи.
5. В парке установлен автомат, продающий кофе. Можно выбрать тип кофе: с сахаром, без сахара, капучино, глясе (цены на кофе установить самостоятельно). Можно выбрать «налить сл стаканом» или «налить без стакана» (стоимость стакана 70 коп.). В автомат можно опускать монеты по 5, 10 рублей и банкноты по 10, 50 и 100 рублей. Описать работу такого автомата, т.е. сообщать пользователю цену выбранного напитка, принимать деньги и, при необходимости, отсчитывать сдачу.
6. Разработать приложение, реализующее игру «Бег лошади на ипподроме». Предусмотреть возможность устанавливать ставки на лошадей и рассчитывать выигрыш. Скорость движения лошадей должна задаваться случайным образом функцией Random.

Вопросы для зачета

1. Новейшие направления в области создания технологий программирования.
2. Законы эволюции программного обеспечения.
3. Основные методы программирования.
4. Элементы теории модульного программирования.
5. Создание модульных программ.
6. Сущность объектно-ориентированного подхода.
7. Объектно-ориентированное проектирование.
8. Алфавит языка, простые типы данных, переменные и выражения, операции.
9. Оператор присвоения, ветвления, цикла, передачи управления.
10. Указатели, массивы, типы данных, определяемые пользователем.
11. Объявление и определение функции, глобальные переменные, возвращение значений, параметры функции.
12. Рекурсивные функции, шаблоны функций. Функция main().
13. Объектный тип данных.
14. Переменные объектного типа.
15. Инкапсуляция.
16. Наследование.
17. Полиморфизм.
18. Объекты и их классы.

19. Конструкторы и деструкторы.
20. Шаблоны классов.
21. Обработка исключительных ситуаций.
22. Среда разработки объектно-ориентированных программ.
23. Стандартные модули оконной операционной среды.
24. Система окон среды разработки объектно-ориентированных программ.
25. Система меню среды разработки объектно-ориентированных программ.
26. Основы визуального программирования.

7. Учебно-методическое и информационное обеспечение дисциплины «Объектно-ориентированные языки и системы программирования»

а) основная литература:

1. Иванова Г.С. Технология программирования: учебник. – М.: Изд-во МГТУ им. Баумана, 2006. (17)

http://kleopatra.pnzgu.ru/cgi-bin/irbis64r_91/cgiirbis_64.exe?P21DBN=KATL&I21DBN=KATL_PRINT&S21FMT=fullw_print&C21COM=F&Z21MFN=7433

2. Шибанов С.В. Основы программирования на языке C++: учебное пособие. – Пенза: Изд-во Пенз. гос. ун-та, 2008. (119)

http://kleopatra.pnzgu.ru/cgi-bin/irbis64r_91/cgiirbis_64.exe?P21DBN=KATL&I21DBN=KATL_PRINT&S21FMT=fullw_print&C21COM=F&Z21MFN=7260

3. Довбуш Г. Ф. Visual C++ на примерах / Г. Ф. Довбуш, А. Д. Хомоненко ; под ред. А. Д. Хомоненко. — СПб.: БХВ-Петербург, 2008. (60)

http://kleopatra.pnzgu.ru/cgi-bin/irbis64r_91/cgiirbis_64.exe?P21DBN=KATL&I21DBN=KATL_PRINT&S21FMT=fullw_print&C21COM=F&Z21MFN=8109

б) дополнительная литература:

4. Павловская Т.А. C/C++. Программирование на языке высокого уровня: Учебник для бакалавров и магистров. – СПб.: Питер, 2012. (5)

http://kleopatra.pnzgu.ru/cgi-bin/irbis64r_91/cgiirbis_64.exe?P21DBN=KATL&I21DBN=KATL_PRINT&S21FMT=fullw_print&C21COM=F&Z21MFN=14891

5. [Фридман А.Л.](#) Объектно-ориентированное программирование на языке СИ++ . - М. : Горячая Линия - Телеком, 2001. - 232 . (5)

[http://kleopatra.pnzgu.ru/cgi-](http://kleopatra.pnzgu.ru/cgi-bin/irbis64r_91/cgiirbis_64.exe?P21DBN=KATL&I21DBN=KATL_PRINT&S21FMT=fullw_print&C21COM=F&Z21MFN=4350)

[bin/irbis64r_91/cgiirbis_64.exe?P21DBN=KATL&I21DBN=KATL_PRINT&S21FMT=fullw_print&C21COM=F&Z21MFN=4350](http://kleopatra.pnzgu.ru/cgi-bin/irbis64r_91/cgiirbis_64.exe?P21DBN=KATL&I21DBN=KATL_PRINT&S21FMT=fullw_print&C21COM=F&Z21MFN=4350)

в) программное обеспечение и Интернет-ресурсы

| № | Название | Электронный адрес | Содержание |
|---|---|---|--|
| 1 | Компьютерная документация от А до Я | http://www.compdoc.ru/prog/builder/ | Документация - программирование на С++ Builder |
| 2 | Visual Studio | http://www.microsoft.com/visualstudio/11/ru-ru | Последние сведения о загрузках, новости и советы по следующему выпуску Visual Studio |
| 3 | Объектно-ориентированное программирование и программная инженерия | http://www.intuit.ru/department/se/oope/ | Учебный курс по объектно-ориентированному программированию |
| 4 | Объектно-ориентированное программирование (ООП) в С++ | http://itedu.ru/courses/cpp/oop-in-cpp | Обучение программированию для студентов математического факультета МПГУ (и всех желающих обучаться по материалам данного сайта самостоятельно) |

8. Материально-техническое обеспечение дисциплины

В целях оптимизации учебного процесса студенты используют рабочие места в компьютерном классе, оборудованном локальной сетью и выходом в Internet, имеющиеся в библиотеке учебники. Все работы выполняются на персональных компьютерах, на соответствующих практических занятиях - с использованием среды программирования Microsoft Visual Studio.

Рабочая программа дисциплины «Высокоуровневые методы программирования» составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 01.03.02 «Прикладная математика и информатика»

Программу составил:
Горюнов Ю.Ю., к.ф.-м.н., доцент


(подпись)

Настоящая программа не может быть воспроизведена ни в какой форме без предварительного письменного разрешения кафедры-разработчика программы.

Программа одобрена на заседании кафедры "Компьютерные технологии"

Протокол № 8^а

от "16" апреля 2015 года

Зав. кафедрой КТ



В.И. Горбаченко

Программа одобрена методической комиссией ФВТ

Протокол № 5^а

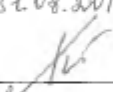
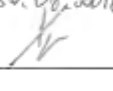
от "17" апреля 2015 года

Председатель методической комиссии
ФВТ



Н. Н. Казанов

Сведения о переутверждении программы на очередной учебный год и регистрации изменений

| Учебный год | Решение кафедры (№ протокола, дата, подпись зав. кафедрой) | Внесенные изменения | Номера листов (страниц) | | |
|-------------|---|---------------------|-------------------------|-------|----------------|
| | | | замененных | новых | аннулированных |
| 2015/2016 | Переутвердить б/з изменений № 01 от 31.08.2015  | | | | |
| 2016/2017 | Переутвердить б/з изменений № 1 от 30.08.2016  | | | | |
| | | | | | |