Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Усынин Максим Валерьевич
Должность: Ректор
Дата подписания: 17.08.2023 12:04:40
Уникальный программный ключ:
f498e59e83f65dd7c3ce7bb8a25cbbabb33ebc58

Частное образовательное учреждение высшего образования
«Международный Институт Дизайна и Сервиса»
(ЧОУВО МИДиС)
Кафедра математики и информатики

УТВЕРЖДАЮ
Ректор
_____ М.В. Усынин
«29» мая 2023 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАИМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ АНАЛИЗ И ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#

Направление подготовки 09.03.03 Прикладная информатика
Профиль подготовки: Разработка Web и мобильных приложений
Квалификация выпускника: бакалавр
Год набор – 2023

Автор-составитель: Чеботарев С.С.

Челябинск 2023

СОДЕРЖАНИЕ

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы……3

2. Показатели и критерии оценивания компетенций на различных этапах их формирования, описание шкал оценивания……5

3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы…….8

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы……...12

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс изучения дисциплины «Объектно-ориентированный анализ и программирование на языке C#» направлен на формирование следующих компетенций:

Код и наименование компетенций выпускника	Код и наименование индикатора достижения компетенций
ПК-1 Способен кодировать на языках программирования (объектно-ориентированных, современных структурных языках, языках современных бизнес-приложений)	ПК-1.1 Разрабатывает код информационных систем и баз данных информационных систем. ПК-1.2 Осуществляет верификацию кода, баз данных и структуры баз данных информационных систем ПК-1.3 Устраняет обнаруженные несоответствия с применением методик тестирования разрабатываемых информационных систем
ПК-2 Способен проводить обследование организаций, выявлять информационные потребности пользователей, формировать требования к информационной системе.	ПК-2.1. Применять методы обследования организации и анализа входной информации для формирования требований к информационной системе ПК-2.2. Осуществлять деятельность по проведению переговоров и презентаций для информирования заказчиков о возможностях информационной системы. ПК-2.3. Выявлять информационные потребностей пользователей, определяет возможности достижения соответствия информационных систем первоначальным требованиям заказчика, разрабатывает стратегии управления заинтересованными сторонами в проекте.

№ п/п	Код компетенции	Наименование компетенции	Этапы формирования компетенций
1.	ПК-1	Способен кодировать на языках программирования (объектно-ориентированных, современных структурных языках, языках современных бизнес-приложений)	*1 Этап – Знать:* ПК-1.1. Способы разработки кода информационных систем и баз данных информационных систем, способы осуществления верификации кода, баз данных и структуры баз данных информационных систем
			2 Этап – Уметь: ПК-1.2. Разрабатывать код информационных систем и баз данных информационных систем, осуществлять верификацию кода, баз данных и структуры баз данных информационных систем
			3 Этап – Владеть: ПК-1.3. Навыками устранения обнаружен-

			ных несоответствий с применением методик тестирования разрабатываемых информационных систем
2.	ПК-2	Способен проектировать и разрабатывать информационные системы в соответствии с требованиями заказчика	*1 Этап – Знать:* ПК-2.1. Методы обследования организации и анализа входной информации для формирования требований к информационной системе, приемы проведения переговоров и презентаций для информирования заказчиков о возможностях информационной системы, стратегии управления заинтересованными сторонами в проекте
			2 Этап – Уметь: ПК-2.2. Применять методы обследования организации и анализа входной информации для формирования требований к информационной системе, осуществлять деятельность по проведению переговоров и презентаций для информирования заказчиков о возможностях информационной системы, выявлять информационные потребностей пользователей, определять возможности достижения соответствия информационных систем первоначальным требованиям заказчика, разрабатывать стратегии управления заинтересованными сторонами в проекте
			3 Этап – Владеть: ПК-2.3. Методами обследования организации и анализа входной информации для формирования требований к информационной системе, методикой проведения переговоров и презентаций для информирования заказчиков о возможностях информационной системы, методами разработки стратегии управления заинтересованными сторонами в проекте

2. ПОКАЗАТЕЛИ И КРИТЕРИИ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

№ п/п	Код компетенции	Наименование компетенции	Критерии оценивания компетенций на различных этапах формирования	Шкала оценивания
1.	ПК-1	Способен кодировать на языках программирования (объектно-	*1 Этап – Знать:* ПК-1.1. Способы разработки кода информационных си-	«ОТЛИЧНО» 1. Глубокое и прочное усвоение программного материала. 2. Правильная формулировка

		ориентированных, со-временных структурных язы-ках, языках со-временных бизнес-приложений)	стем и баз данных информационных систем, способы осуществления ве-рификации кода, баз данных и структуры баз данных инфор-мационных систем	определения качества. 3. Понимание значения реализации технологий продвижения гостиничного продукта. «ХОРОШО» 1. Хорошее знание программного материала. 2. Недостаточно полное изложение теоретического вопроса. 3. Наличие незначительных неточностей в употреблении терминов, классификаций. 4. Логичное изложение вопроса, соответствие изложения научному стилю. «УДОВЛЕТВОРИТЕЛЬНО»: 1. Поверхностное усвоение программного материала. 2. Недостаточно полное изложение теоретического вопроса. 3. Наличие неточностей в употреблении терминов, классификаций. 4. Неумение четко сформулировать задачи. 5. Отсутствие навыков научного стиля изложения. «НЕУДОВЛЕТВОРИТЕЛЬНО»: 1. Незнание значительной части программного материала. 2. Неспособность объяснить основные понятия и закономерности. 3. Неумение выделить главное, сделать выводы и обобщения.
			2 Этап – Уметь: ПК-1.2. Разрабатывать код информационных систем и баз данных инфор-мационных систем, осуществлять вери-фикацию кода, баз данных и структуры баз данных инфор-мационных систем	
			3 Этап – Владеть: ПК-1.3. Навыками устранения обнару-женных несоответ-ствий с применени-ем методик тестиро-вания разрабатывае-мых информацион-ных систем	
2.	ПК-2	Способен проекти-ровать и разраба-тывать информа-ционные системы в соответствии с требованиями за-казчика	*1 Этап – Знать:* ПК-2.1. Методы об-следования органи-зации и анализа входной информа-ции для формирова-ния требований к информационной системе, приемы проведения перего-воров и презентаций для информирования заказчиков о воз-можностях инфор-мационной системы, стратегии управле-ния заинтересован-ными сторонами в проекте	
			2 Этап – Уметь: ПК-2.2. Применять методы обследова-ния организации и	

			анализа входной информации для формирования требований к информационной системе, осуществлять деятельность по проведению переговоров и презентаций для информирования заказчиков о возможностях информационной системы, выявлять информационные потребностей пользователей, определять возможности достижения соответствия информационных систем первоначальным требованиям заказчика, разрабатывать стратегии управления заинтересованными сторонами в проекте	
			3 Этап – Владеть: ПК-2.3. Методами обследования организации и анализа входной информации для формирования требований к информационной системе, методикой проведения переговоров и презентаций для информирования заказчиков о возможностях информационной системы, методами разработки стратегии управления заинтересованными сторонами в проекте	

3. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

1 ЭТАП – ЗНАТЬ

Комплект вопросов/заданий

1. Проведите сравнения структурного программирования и объектно ориентированного программирования. Чем отличаются способы организации программ в объектно ориентированном программировании и структурном программировании?

2. Расскажите об объектно ориентированном программировании. Что вызвало его появление? Перечислите три основные черты которые характерны для всех объектно ориентированных языков.

3. Расскажите об инкапсуляции. Что является единицей инкапсуляции в **C#**? Чем отличаются открытые и закрытые данные? К каким преимуществам приводит использование инкапсуляции? Почему инкапсуляции недостаточно для объектно ориентированности языка?

4. Расскажите о наследовании. Проведите сравнение множественного и одиночного наследования. Проведите сравнение интерфейсного наследования и наследования реализации.

5. Расскажите о полиморфизме. Как можно описать полиморфизм одной фразой? Каким образом с помощью полиморфизма можно снизить сложность программы? Каким образом с помощью полиморфизма можно упростить расширяемость программ?

6. Расскажите о массивах. Какие типы данных могут храниться в массивах? Возможно ли изменить длину массива? Что произойдёт в случае выхода за пределы массива?

7. Расскажите о многомерных массивах. Чем отличаются массив массивов и многомерный массив? Что такое рваный массив? Должны ли все данные в рваном массиве быть одного типа?

8. Расскажите о коллекциях. Перечислите основные типы коллекций. Перечислите используемые интерфейсы. Что такое типизированные коллекции? В чём преимущество использования типизированных коллекций?

9. Расскажите о хэш таблицах. Почему хэш таблица называется хэш таблицей? Какие типы данных могут храниться в хэш таблице? Какие типы могут использоваться в качестве ключей? Приведите пример приложения, для которого хэш таблица подходит лучше других коллекций.

10. Расскажите о динамических массивах. Почему они называются динамическими? Какие типы данных могут храниться в динамических массивах? Какие типы могут использоваться в качестве индекса? Приведите пример приложения, для которого динамический массив подходит лучше других коллекций.

11. Расскажите об интерфейсах. Что определяет интерфейс? Что интерфейс не определяет? Что может содержать интерфейс? Наследует ли интерфейс тип **Object**? Чем интерфейс отличается от абстрактного класса? Почему в интерфейсах не указываются модификаторы доступа?

12. Проведите сравнение использования базовых классов и интерфейсов. При каком отношении между типами следует использовать интерфейсы? Перечислите преимущества использования базовых классов.

13. Расскажите о явной реализации интерфейсов. Почему при явной реализации интерфейсов не указывается модификатор доступа? Как с помощью явной реализации интерфейсов можно добиться большей безопасности типов? Как с помощью явной реализа-

ции интерфейсов можно реализовать методы нескольких интерфейсов с одинаковыми именами и сигнатурами?

14. Расскажите о механизме обработки ошибок используемом в WIN32 API. Что возвращает функция если она не может нормально выполнить свою работу? Как узнать что именно произошло?

15. Расскажите о механизме обработки ошибок используемом в COM. Как узнать что произошло нарушение допущений метода? Как можно установить его причину?

16. Расскажите, как вы понимаете что такое исключение. Объясните, почему многие специалисты считают, что рассматривать исключение как ошибку неправильно. Проведите сравнение исключений и механизмов обработки ошибок используемых в WIN32 API и COM. Перечислите преимущества использования исключений.

17. В какой последовательности должны распологаться блоки catch ассоциированные с одним блоком try? Сколько блоко catch и finally может быть ассоциированно с одним блоком try? Что должно находиться в блоке finally? Может ли в блоке finally бросаться исключение?

18. Расскажите, как происходит обработка сгенерированного исключения. Когда блок catch ловит исключение? Что произойдёт, если блок catch исключение не поймал? Когда и в какой последовательности будут выполняться блоки finally, если блок catch соответствующий первому finally исключение не поймал, а блок catch соответствующий второму поймал? Что произойдёт если в приложении произошло исключение, но ни один блок catch его не поймал?

19. Расскажите об иерархии классов для исключений. Зачем в этой иерархии были созданы два потомка базового класса для исключений? Было ли это оправдано? Приведите пример исключения которое плохо вписывается в эту иерархию. Чем нужно руководствоваться при создании своих классов для исключений?

20. Расскажите о базовом классе для исключений. Перечислите его свойства. Для чего они предназначены? Перечислите свойства у которых нет мутатора и объясните почему.

21. Расскажите о делегатах. Почему делегаты называются делегатами? Какую проблему они решают? Что на самом деле происходит когда компилятор встречает в коде делегат? Можно ли обойтись без делегатов с помощью интерфейсов?

22. Расскажите о цепочках делегатов. В каком порядке вызываются делегаты в цепочке? Что произойдёт с возвращаемыми значениями делегатов в цепочке? Что произойдёт если один из делегатов в цепочке сгенерирует исключение? Почему для делегатов существует два класса? Можно ли сейчас сделать один класс для делегатов?

23. Расскажите о поддержке делегатами асинхронных вызовов. Чего позволяет избежать поддержка делегатами асинхронных вызовов? Что такое пул потоков? Можно ли сказать, что использование асинхронных вызовов позволяет приложению работать быстрее? Как можно получить возвращаемое значение метода вызванного асинхронно? Что произойдёт, если вызванный асинхронно метод сгенерирует исключение?

24. Расскажите о ковариантности и контрвариантности делегатов. Чем они отличаются? Чего можно с их помощью добиться? Как ковариантность и контравариантность связаны с неявным приведением типов?

25. Расскажите о позднем связывании делегатов. Что нужно знать во время компиляции, чтобы во время выполнения программы использовать позднее связывание? Можно ли использовать для позднего связывания делегат который не был объявлен?

26. Расскажите о событиях. На чём основанны события в C#? Что на самом деле происходит когда компилятор встречает в коде объявление события? Что такое источник события? Что такое слушатель события? Что такое обработчик события? Как можно зарегистрировать обработчика события? Возможно ли реализовать функциональность событий не используя объявления событий?

27. Расскажите о реализации методов для регистрации и дерегистрации событий самостоятельно. Объясните как это можно сделать. Что нужно определить в классе помимо этих двух методов? Перечислите две ситуации когда возникает в ручной реализации методов для регистрации и дерегистрации событий.

28. Расскажите о шаблоне рекомендуемом .NET Framework. Соответствуют ли этому шаблону классы .NET Framework? В чём преимущество использования этого шаблона? На чём основан этот шаблон? Какой дополнительный класс может потребоваться создать для использования этого шаблона?

29. Расскажите, в чём отличие класса с одним событием и класса с большим числом событий. Приведите пример такого класса. Какой шаблон рекомендуется для класса со многими событиями? Зачем в этом случае используется коллекция для делегатов?

2 ЭТАП – УМЕТЬ

Комплект типовых задач

1. Построить систему классов для описания плоских геометрических фигур: круга, квадрата, прямоугольника. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и поворота на заданный угол.

2. Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность раздельного изменения составных частей адреса, создания и уничтожения объектов этого класса.

3. Составить описание класса для представления комплексных чисел с возможностью задания вещественной и мнимой частей как числами типов double, так и целыми числами. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел.

4. Составить описание класса для работы с цепными списками строк (строки произвольной длины) с операциями включения в список, удаления из списка элемента с заданным значением данного, удаления всего списка или конца списка, начиная с заданного элемента.

5. Составить описание класса для объектов-векторов, задаваемых координатами концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами.

6. Составить описание класса прямоугольников со сторонами, параллельными осям координат. Предусмотреть возможность перемещения прямоугольников на плоскости, изменения размеров, построения наименьшего прямоугольника, содержащего два заданных прямоугольника, и прямоугольника, являющегося общей частью (пересечением) двух прямоугольников.

7. Составить описание класса для определения одномерных массивов целых чисел (векторов). Предусмотреть возможность обращения к отдельному элементу массива с контролем выхода за пределы индексов, возможность задания произвольных границ индексов при создании объекта и выполнения операций поэлементного сложения и вычитания массивов с одинаковыми границами индексов, умножения и деления всех элементов массива на скаляр, печати (вывода на экран) элементов массива по индексам и всего массива.

8. Составить описание класса для определения одномерных массивов строк фиксированной длины. Предусмотреть возможность обращения к отдельным

9. Разработать класс «Комплексное число». Определить в нем конструктор, перегрузить арифметические операции, операции сравнения, операцию преобразования в строку и статический метод получения комплексного числа из строки

10. Разработать класс «Комплексное число», в котором данные хранятся в двух видах: алгебраической и тригонометрической формах. Определить в нем конструкторы и

деструктор, перегрузить арифметические операции, операции сравнения, операцию преобразования в строку и статический метод получения комплексного числа из строки, написать методы преобразования числа из одной формы в другую. Протестировать все возможности класса

11. Разработать класс «Дата». Определить в нем конструкторы и деструктор, перегрузить операцию добавления к дате заданного количества дней, операцию вычитания двух дат, операции сравнения и операцию преобразования в символьную строку, а также статический метод получения даты из строки

12. Разработать класс «Прямоугольник». Определить в нем конструкторы и деструктор, перегрузить операцию пересечения прямоугольников (операция “*”), операцию вычисления площади прямоугольника операции сравнения (по площади), операцию преобразования в символьную строку и метод получения объекта-прямоугольника из строки

13. Разработать класс «Треугольник». Определить в нем конструкторы и деструктор, перегрузить операцию преобразования в вещественное число (площадь треугольника), операцию проверки включения точки в треугольник, операции сравнения треугольников (по площади), операцию преобразования в символьную строку и метод получения объекта-треугольника из строки

14. Создать класс «Студент», который определяется полями ФИО, номер группы, название факультета, название специальности, средний балл успеваемости. Пусть имеется массив объектов этого класса. Разработать метод выбора студентов из массива по условию (учится на конкретном факультете, имеет средний балл более заданного уровня и пр.). Для определения, удовлетворяет ли объект условию, передать в метод параметр- делегат.

15. Пусть имеется класс «Матрица». Определить различные методы, которые осуществляют преобразование матрицы (транспонирование, поворот, сортировка строк, изменение порядка столбцов на обратный и пр.). В диалоговом режиме задать последовательность действий, которую нужно произвести с объектом-матрицей, сохраняя ее в переменной-делегате. Предусмотреть команду меню выполнения этих действий, которая обращается к сформированному делегату.

16. Разработать класс для решения задачи поиска точки, минимизирующей функцию на отрезке. Реализовать функции решения этой задачи различными методами – методом деления отрезка пополам, методом золотого сечения, методом касательных, методом Фибоначчи. Выбранный способ решения сохранить в делегат

17. Разработать класс «Множество». Определить конструкторы и деструктор. Переопределить операции объединения, пересечения и разности двух множеств, методы для организации ввода-вывода. Написать методы проверки включения одного множества в другое, проверки равенства двух множеств, проверки пустоты множества

18. Разработать класс «Целое число в заданной системе счисления». Число должно храниться в виде массива целых чисел (разрядов числа). Определить конструктор, методы для организации ввода-вывода, операции сложения, вычитания, умножения, деления и взятия остатка от деления двух чисел и операции сравнения. Написать методы перевода числа из одной системы счисления в другую

19. Разработать класс «Бинарное дерево сортировки». Написать конструкторы и деструктор, методы добавления нового узла, удаления узла по ключевому значению, вычисления глубины дерева, объединения двух деревьев, вычисления количества узлов на заданном уровне, определения подобия двух деревьев.

20. Разработать класс «Граф» в виде списка смежности. Определить конструкторы и деструктор. Переопределить операции ввода-вывода. Написать методы проверки связности графа, проверки полноты графа, проверки двудольности графа, получения дополнения графа, нахождения источника графа, нахождения стока графа. Наследовать от этого класса класс «Взвешенный граф». Написать методы получения кратчайшего пути между двумя вершинами алгоритмом Дейкстры, получения каркаса минимального веса алгоритмами Прима и Краскал

21. Определить интерфейс «Фигура на плоскости» и раскрыть его для классов «Треугольник», «Прямоугольник», «Многоугольник», «Круг» и пр. Определить класс «Рисунок» как массив объектов-фигур. Реализовать для рисунка операции перемещения, распечатки информации о рисунке, повороте и пр.

22. Создать иерархию классов «Вагоны пассажирского поезда» с разделением на купейные, плацкартные, СВ. Каждый класс вагона должен содержать информацию о количестве мест разных типов (нижнее, верхнее, нижнее боковое, верхнее боковое), о наличии дополнительных услуг и ценах на них. С помощью виртуальных функций получить полный доход от эксплуатации вагона. Создать класс «Пассажирский поезд», который хранит список вагонов. Подсчитать доход от одного рейса поезда.

23. Создать иерархию классов-многоугольников: «Треугольник», «Четырехугольник», «Пятиугольник», «Шестиугольник». Создать класс «Фигура на плоскости», который задает фигуру как массив объектов- многоугольников. Определить в классе методы перемещения фигуры, определения, принадлежит ли точка фигуре и др.

3 ЭТАП – ВЛАДЕТЬ

Типовые кейс-задачи

Кейс-задача 1. **Singleton**

Задание:

Использовать паттерн синглтон (Singleton) для создания системы логгирования сценариев взаимодействия пользователя с графическим интерфейсом приложения.

Для решения кейса необходимо:

1) познакомиться с паттерном «Одиночка» (Singleton);
2) актуализировать знания по использованию классов .NET для работы с файловой системой;
3) подготовить UML-диаграмму проекта;
4) разработать приложение с интерфейсом (на выбор) на платформе Windows Forms или WPF или ASP.NET
5) обеспечить взаимодействие приложения и Logger

Кейс-задача 2. **Abstract Factory**

Задание:

Использовать паттерн фабрика (Abstract Factory) для создания системы позволяющей производить разные модели персональных компьютеров на основе набора различных комплектующих.

Для решения кейса необходимо:

1) познакомиться с паттерном «Фабрика» (Abstract Factory);
2) подготовить UML-диаграмму проекта;
3) разработать приложение с интерфейсом (на выбор) на платформе Console или Windows Forms
4) обеспечить взаимодействие приложения и Фабрики классов

Кейс-задача 3. **Стратегия**

Задание:

Использовать паттерн «Стратегия» (Strategy) для создания системы, обладающей модульной архитектурой.

Для решения кейса необходимо:

1) познакомиться с паттерном «Стратегия» (Strategy);
2) подготовить UML-диаграмму проекта;
3) разработать приложение с интерфейсом (на выбор) на платформе Console или Windows Forms

4) обеспечить взаимодействие приложения и архитектуры «Стратегия»

Кейс-задача 4. **Итератор**

Задание:

Использовать паттерн «Итератор» (Iterator) для создания структуры данных «Одно-связанный список» или «Бинарное дерево».

Для решения кейса необходимо:

1) познакомиться с паттерном «Итератор» (Iterator);
2) подготовить UML-диаграмму проекта;
3) разработать структуру данных «Одно-связанный список» или «Бинарное дерево» и приложение для тестирования этой структуры

Темы групповых творческих проектов

1) Разработка компьютерной игры в жанре платформер с использованием framework Microsoft XNA Game Studio
2) Разработка компьютерной игры в жанре арканоид с использованием framework Microsoft XNA Game Studio
3) Разработка игрового движка для компьютерной игры с использованием framework Microsoft XNA Game Studio
4) Разработка компьютерной игры в жанре квест с использованием платформы Windows Presentation Foundation
5) Разработка прототипа бизнес-приложения “Банкомат” с использованием платформы WPF и базы данных MS SQL Server
6) Разработка прототипа бизнес-приложения “Кинотеатр” с использованием платформы WPF и базы данных MS SQL Server
7) Разработка прототипа бизнес-приложения “Авиакасса” с использованием платформы WPF и базы данных MS SQL Server
8) Разработка прототипа бизнес-приложения “Поисковая система” с использованием платформы WPF и базы данных MS SQL Server

Примерные темы курсовых работ

Проектирование и разработка программного обеспечения с использованием принципов и технологий объектно-ориентированного анализа и программирования. При реализации проекта для хранения информации нужно использовать файловое хранилище собственного или открытого (XML) формата.

1. **Проектирование и разработка информационной системы «ГАИ»**.

Разработать приложение, помогающее сотрудникам ГАИ: заполнение информации об автомобилях (регистрационный номер, цвет автомобиля, год выпуска, адрес проживания владельца, наложенные штрафы) и выполнение запросов. Например, организовать запросы на выдачу сведений об автолюбителях, имеющих:

а) автомобиль заданной марки определенного цвета;
б) авто с заданным номером;
в) авто заданной марки с известной цифровой частью номера;
г) авто заданного цвета.
д) авто со штрафами
е) авто с наибольшей суммой штрафов
ж) авто с определенным количеством штрафов

2. **Проектирование и разработка информационной системы «СПОРТ»**.

Дан список участников соревнования, для каждого указаны время старта и финиша (часы, мин., сек). Разработать приложение, обеспечивающее заполнение данных и выполнение запросов. Например, запросы на выдачу сведений:

а) Выводить список участников по возрастанию или убыванию

б) лучший результат соревнования;

в) наихудший результат;

г) участников, сошедших с дистанции;

д) время 5 лучших призеров, …

3. Проектирование и разработка информационной системы «СКЛАД».

Имеются сведения о товарах, находящихся на складе: наименование, объем партии, дата поступления на склад, стоимость единицы товара. Разработать приложение, которое бы позволяло вводить и выводить информацию по запросу. В перечень запросов ввести требование отсортировать данные по различным критериям, например, по дате поступления на склад.

4. Проектирование и разработка информационной системы «БИБЛИОТЕКА». Имеются сведения о книгах, находящихся в библиотеке: ФИО автора, название, наименование издательства, год издания, жанр, количество страниц. Также имеются сведения о читателях (ФИО, телефон, адрес). Также имеются сведения о том, какие книги находятся на руках у читателя, и дата получения книги. Разработать приложение, которое бы позволяло вводить и выводить информацию по запросу. Например, вывести список читателей-должников, список книг определённого жанра и т.д.

5. Проектирование и разработка информационной системы «АВТОВОКЗАЛ». Имеется расписание движения автобусов: номер рейса, тип автобуса, название маршрута, время отправления, время в пути. Для каждого маршрута хранится пункт назначения (пункт отправления у всех общий) и список всех остановок (для каждой остановки хранится название и расстояние от пункта отправления). Для каждого рейса ведётся учёт свободных/проданных и забронированных мест.

6. Проектирование и разработка информационной системы «МАСТЕРСКАЯ». Имеется информация о сданной в ремонт радиоаппаратуре: марку изделия, дату приёма в ремонт, состояние готовности заказа. Разработать функцию анализа данных и выдачи информации о числе и характере заказов на текущие сутки и объёме выполненных услуг за текущий квартал.

7. Проектирование и разработка информационной системы «СЕКРЕТАРЬ». Разработать приложение «секретарь». Программа должна выполнять следующие функции (помимо ввода данных в систему).

а) по заданной дате сообщать перечень ФИО тех, кому нужно позвонить

б) ФИО тех, с кем нужно встретиться;

г) ФИО тех, кого нужно поздравить с днем рождения;

д) список дел на заданный день

8. Проектирование и разработка информационной системы «КАФЕДРА».

Имеются информация вида: шифр и название кафедры, наименование и шифр предмета изучения, наименование и шифр специальности, курс и семестр. Разработать запросы и подготовить функции для ввода и вывода информации по запросу. Например, ввести функцию, которая по требованию выдает перечень дисциплин, относящихся к отдельной кафедре.

9. Проектирование и разработка информационной системы «ФАЙЛОВЫЙ МЕНЕДЖЕР».

Разработка функционального аналога известного файлового менеджера (Far Manager/ Total Commander)

10. Проектирование и разработка информационной системы «ГРАФИЧЕСКИЙ РЕДАКТОР».

Разработка графического векторного редактора (пример - CorelDraw)

11. Проектирование и разработка информационной системы «МОНОПОЛИЯ». Разработка компьютерной версии настольной игры (монополия, http://www.mirf.ru/Articles/print2950.html)

12. Проектирование и разработка информационной системы «АУКЦИОН». Разработка информационной системы проведения on-line аукционов

13. Проектирование и разработка информационной системы «ПЛАТФОРМЕР». Разработка игрового спрайтового движка и реализация на его основе простейшего платформера.

14. Проектирование и разработка информационной системы «СТРАТЕГИЯ». Разработка игрового спрайтового движка и реализация на его основе пошаговой или стратегии в реальном времени.

Вопросы к экзамену

3 семестр

1) Причины появления и актуальность объектно-ориентированного программирования.
2) Содержание концепции объектно-ориентированного программирования.
3) Основные идеи объектно-ориентированного программирования: использование объекта в качестве основной компоненты программы, представление программы как совокупности взаимодействующих объектов.
4) Характеристики объектно-ориентированных языков программирования
5) Основные области применения объектно-ориентированного программирования: разработка программ, создание баз данных, построение прикладных систем.
6) Модульный подход в программировании
7) Основные преимущества применения объектно-ориентированного программирования.
8) Основные понятия ООП: объект, класс, данные, методы, доступ, наследование свойств.
9) Объект как совокупность данных и набора методов.
10) Семантика объекта: данные как отражение состояния объекта и методы как средства обеспечения взаимодействия объектов.
11) Представление данных.
12) Классификация методов: конструкторы, деструкторы, селекторы и модификаторы.
13) Инкапсуляция данных и скрытие информации.
14) Создание, использование и уничтожение объектов.
15) Обращение к объекту в целом и его отдельным полям, методам и свойствам.
16) Создание объектов уже существующего типа.
17) Использование объекта.

18) Идентификация объектов.
19) Определение динамики поведения объектов.
20) Принципы объектно-ориентированного программирования и их применение к классам
21) Классы объектов: назначение и семантика.
22) Классы и абстрактные типы данных.
23) Инкапсуляция: назначение, предметы, прагматика.
24) Инкапсуляция и области видимости.
25) Защита на уровне объекта и на уровне класса.
26) Представление классов. Объекты как экземпляры классов.
27) Отношение наследования для классов.
28) Реализация спецификации и обобщения свойств объектов.
29) Простое и множественное наследование.
30) Сценарии наследования.
31) Иерархия классов.
32) Системы объектов и классов
33) Статические, абстрактные и виртуальные методы.
34) Сферы применения статических и виртуальных методов.
35) Реализация статических и виртуальных методов.
36) Переопределение методов и свойств.
37) Абстрактные методы.
38) Распределение функций при использовании виртуальных методов.
39) Повторное использование кода за счет использования виртуальных методов.
40) Использование виртуальных функций при внутренней реализации класса.

Вопросы к экзамену

4 семестр

1) Объектно-ориентированное проектирование – основные понятия и модели.
2) Составные части анализа, составление технического задания, приемы и организация анализа.
3) Объектно-ориентированные языки
4) Объектно-ориентированные языки и их основные возможности.
5) Классификация и номенклатура языков, средства, технология и сферы применения.
6) Язык моделирования UML
7) Язык моделирования UML. Понятие проектирования по образцам.
8) Библиотеки функций и библиотеки классов.
9) Программа как система взаимодействующих объектов
10) Библиотеки функций и библиотеки классов. Программа как система взаимодействующих объектов.
11) Интегрированная среда разработки (IDE), интерфейс. Взаимодействие объектов через интерфейс
12) Интегрированная среда разработки (IDE), интерфейс: правила организации, методы и средства программирования.
13) Проекты. Формы. Модули. Взаимодействие объектов.
14) Интерфейс объекта по умолчанию. Применение интерфейсов.
15) Описание и реализация интерфейсов.
16) Наследование интерфейсов.
17) Указание и реализация интерфейса в объявлении класса.
18) Переменные типа интерфейс.
19) Разработка и применение интерфейсов.
20) Методы и средства программирования компонентов.

21) Создание собственных компонентов.
22) Создание и установка компонента. Разработка визуального компонента.
23) Пример построения компонента.
24) Добавление свойств, методов, событий.
25) Модификация существующего компонента.
26) Объектно-ориентированная реализация алгоритмов.
27) Реализация алгоритма внутри класса.
28) Определение параметров вычислений.
29) Разбиение алгоритма на процедуры.
30) Использование глобальных процедур и процедур, относящихся к классам.
31) Сохранение результатов расчетов.
32) Правила распределения функций между объектами.
33) Выделение объектов, отвечающих за отдельные операции в отдельные классы.
34) Распределение функций между различными уровнями обобщения.
35) Создание обобщенных классов для реализации общих функций.
36) События. Суть и назначение событий. Стандартная схема реализации событий.
37) Описание событий. Присвоение обработчика события.
38) Сферы применения событий.
39) Объектно-ориентированные системы: методы, языки и способы программирования.

4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

1 ЭТАП – ЗНАТЬ

Критерии оценивания ответов на комплект вопросов/заданий

Оценка	**Критерии**
«отлично»	Глубокое и прочное усвоение знаний программного материала. Правильная формулировка, знание основных терминов. Знание научной литературы по вопросам. Точные, полные и логичные ответы на дополнительные вопросы.
«хорошо»	Хорошее знание программного материала. Наличие неточностей в употреблении терминов. Логичное изложение вопроса, соответствие изложения научному стилю. Правильные ответы на дополнительные вопросы.
«удовлетворительно»	Поверхностное усвоение программного материала. Недостаточное знание научной литературы по вопросу. Затруднение в приведении примеров, подтверждающих теоретические положения. Отсутствие навыков научного стиля изложения. Неправильные ответы на дополнительные вопросы.
«неудовлетворительно»	Незнание значительной части программного материала. Неспособность проиллюстрировать теоретические положения языковым материалом.

	Незнание научной литературы по вопросу. Неправильные ответы на дополнительные вопросы.

2 ЭТАП – УМЕТЬ

Критерии оценивания результатов решения комплекта задач

Оценка	**Критерии**
«отлично»	Алгоритм составлен верно; программный код написан грамотно; использован оптимальный метод решения; программа выдаёт верный результат при разных входных данных
«хорошо»	Алгоритм составлен верно; программный код написан грубо, без соблюдения правил и рекомендаций; использован неоптимальный метод решения; программа иногда выдаёт неверный результат при некоторых граничных входных данных
«удовлетворительно»	Алгоритм составлен, но содержит ошибки но программа написана и иногда выдаёт верный результат
«неудовлетворительно»	Алгоритм составлен неверно;

3 ЭТАП – ВЛАДЕТЬ

Критерии оценивания работы с кейс-задачами

Оценка	**критерии**
«отлично»	решение задачи осуществляется командой; разработана архитектура классов; разработано приложение; на защите кейс-задачи были получены верные ответы на все дополнительные вопросы
«хорошо»	решение задачи осуществляется командой; разработана архитектура классов; разработано приложение; на защите кейс-задачи при ответах на вопросы были допущены ошибки
«удовлетворительно»	разработана архитектура классов; разработано приложение, но в приложении имеются ошибки и недоработки; на защите кейс-задачи при ответах на вопросы были допущены ошибки
«неудовлетворительно»	разработана архитектура классов; не было разработано приложение

Критерии оценивания групповых творческих проектов

Оценка	**критерии**
«отлично»	разработана архитектура классов; разработано приложение; разработанное приложение полностью соответствует техническому заданию и реализует требуемый функционал
«хорошо»	разработана архитектура классов; разработано приложение; разработанное приложение соответствует техническому заданию не в полном объёме имеются недоработки и ошибки

«удовлетворительно»	разработана архитектура классов; разработано приложение; разработанное приложение не соответствует техническому заданию работа не выполнена в полном объёме
«неудовлетворительно»	работа не выполнена

Критерии оценивания курсовой работы

Оценка	**Критерии**
«отлично»	разработана архитектура классов; разработано приложение; разработанное приложение полностью соответствует техническому заданию и реализует требуемый функционал дано всестороннее и глубокое освещение избранной темы в тесной взаимосвязи с практикой, а её автор показал умение работать с различными видами источников, систематизировать, классифицировать, обобщать материал, формулируя выводы, соответствующие поставленным целям;
«хорошо»	разработана архитектура классов; разработано приложение; разработанное приложение соответствует техническому заданию не в полном объёме имеются недоработки и ошибки работа, отвечающая основным, предъявляемым к ней требованиям. Студент обнаруживает глубокие знания по предмету и владеет навыками научного исследования, но при этом имеются незначительные замечания по содержанию работы, по процедуре защиты (студент не может дать аргументировано ответы на вопросы);
«удовлетворительно»	разработана архитектура классов; разработано приложение; разработанное приложение не соответствует техническому заданию работа не выполнена в полном объёме в основном, соблюдены общие требования, но неполно раскрыты разделы плана, работа носит реферативный характер, отсутствуют аргументированные выводы. Автор курсовой работы посредственно владеет материалом, поверхностно отвечает на вопросы, в процессе защиты курсовой работы;
«неудовлетворительно»	работа, если установлен акт несамостоятельного выполнения, работы, имеются принципиальные замечания по многим параметрам, содержание не соответствует теме, допущены грубые теоретические ошибки.

Критерии оценивания знаний на экзамене

Оценка	Критерии
«отлично»	Глубокое и прочное усвоение программного материала. Знание пакетов прикладных программ.

	Знание основных принципов построения пакетов прикладных программ. Знание основных задач прикладных программ. Свободное владение пакетами прикладных программ. Точность и обоснованность выводов. Безошибочное выполнение практического задания. Точные, полные и логичные ответы на дополнительные вопросы.
«хорошо»	Хорошее знание программного материала. Недостаточно полное изложение теоретического вопроса экзаменационного билета. Наличие незначительных неточностей в употреблении терминов, классификаций. Знание основных пакетов прикладных программ. Неполнота представленного иллюстративного материала. Точность и обоснованность выводов. Логичное изложение вопроса, соответствие изложения научному стилю. Негрубая ошибка при выполнении практического задания. Правильные ответы на дополнительные вопросы.
«удовлетворительно»	Поверхностное усвоение программного материала. Недостаточно полное изложение теоретического вопроса экзаменационного билета. Затруднение в приведении примеров, подтверждающих теоретические положения. Наличие неточностей в употреблении терминов, классификаций. Неумение четко сформулировать выводы. Отсутствие навыков научного стиля изложения. Грубая ошибка в практическом задании. Неточные ответы на дополнительные вопросы.
«неудовлетворительно»	Незнание значительной части программного материала. Неспособность привести примеры пакетов прикладных программ Неумение выделить главное, сделать выводы и обобщения. Грубые ошибки при выполнении практического задания. Неправильные ответы на дополнительные вопросы.