Министерство науки и высшего образования Российской Федерации

Муромский институт (филиал)

федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(МИ ВлГУ)

Кафедра *ПИн*

«УТВЕРЖДАЮ»
Заместитель директора по УР
_____Д.Е. Андрианов
17.05.2022

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Объектно-ориентированное программирование

Направление подготовки *09.03.04 Программная инженерия*

Профиль подготовки *Методы и средства разработки программного обеспечения*

Семестр	Трудоемкость, час./зач. ед.	Лекции, час.	Практические занятия, час.	Лабораторные работы, час.	Консультация, час.	Контроль, час.	Всего (контактная работа), час.	СРС, час.	Форма промежуточного контроля (экз., зач., зач. с оц.)
2	**252 / 7**	**32**		**36**	**5,2**	**0,35**	**73,55**	**151,8**	**Экз.(26,65)**
3	**252 / 7**	**28**		**40**	**4,8**	**0,35**	**73,15**	**152,2**	**Экз.(26,65)**
Итого	**504 / 14**	**60**		**76**	**10**	**0,7**	**146,7**	**304**	**53,3**

Муром, 2022 г.

1. Цель освоения дисциплины

Цель дисциплины: приобретение студентами знаний об основных понятиях объектно-ориентированного программирования, изучение подходов программирования, проектирования и реализации изолированных классов, принципов объектно-ориентированного программирования, освоение принципов абстрагирования при выделении классов объектно-ориентированной программной системы, обзор современных технологий разработки объектно-ориентированных приложений.

Задачи дисциплины: освоение основ теории объектно-ориентированного программирования; освоение основных методов и технологий объектно-ориентированного программирования; получение студентами практических навыков в разработке на языке C#.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина «Объектно-ориентированное программирование» базируется на знаниях, полученных студентами по курсу «Основы алгоритмизации и программирования», «Архитектура вычисительных систем». Углубление и расширение вопросов, изложенных в данном курсе, будет осуществляться во время работы студентов над дисциплинами: "Структуры и алгоритмы обработки данных", "Организация баз данных", а также при написании бакалаврских работ.

3. Планируемые результаты обучения по дисциплине

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции	Результаты обучения по дисциплине	
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов;	ОПК-6.1 Использует современные языки, утилиты и среды программирования	Знает современные языки программирования (ОПК-6.1) Умеет использовать современные языки программирования для решения практических задач (ОПК-6.1) Имеет навыки использования современных технологий разработки ПО (ОПК-6.1)	Тест

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 14 зачетных единиц, 504 часа.

4.1. Форма обучения: очная

Уровень базового образования: среднее общее.
Срок обучения 4г.

4.1.1. Структура дисциплины

№ п\п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Основы программирования на языке C#	2	22		16					146	Тестирование
2	Введение в ООП	2	10		20					5,8	Тестирование
Всего за семестр		252	32		36			5,2	0,35	151,8	Экз.(26,65)
3	Введение в ООП	3	12		8					140,2	Тестирование
4	Стандартные классы .NET	3	10		8						Тестирование
5	Шаблоны проектирования	3	6		24					12	Тестирование
Всего за семестр		252	28		40			4,8	0,35	152,2	Экз.(26,65)
Итого		504	60		76			10	0,7	304	53,3

4.1.2. Содержание дисциплины

4.1.2.1. Перечень лекций

Семестр 2

Раздел 1. Основы программирования на языке C#

Лекция 1.

Введение в платформу .NET (2 часа).

Лекция 2.

Основные понятия языка C# (2 часа).

Лекция 3.

Линейные программы в языке C# (2 часа).

Лекция 4.

Управляющие операторы в языке C# (2 часа).

Лекция 5.

Массивы в языке C# (2 часа).

Лекция 6.

Строки в языке C# (2 часа).

Лекция 7.

Введение в технологию Windows Forms (2 часа).

Лекция 8.

Класс object. Преобразования типов (2 часа).

Лекция 9.

Преобразование типов в C# (2 часа).

Лекция 10.

Структуры. Записи. Кортежи (2 часа).

Лекция 11.

Обработка исключений. Класс Exception (2 часа).

Раздел 2. Введение в ООП

Лекция 12.

Классы и объекты в языке C# (2 часа).

Лекция 13.

Данные классов. Методы. Конструкторы. Свойства (2 часа).

Лекция 14.

Статические классы. Абстрактные классы (2 часа).

Лекция 15.

Полиморфизм конструкторов, методов (2 часа).

Лекция 16.

Наследование (2 часа).

Семестр 3

Раздел 3. Введение в ООП

Лекция 17.

Виртуальные методы. Позднее связывание (2 часа).

Лекция 18.

Перегрузка операций в C# (2 часа).

Лекция 19.

Обобщения. Обобщенные коллекции. Методы расширения (2 часа).

Лекция 20.

Интерфейсы (2 часа).

Лекция 21.

Делегаты. события (2 часа).

Лекция 22.

Делегаты. Анонимные методы. Лямбда выражения (2 часа).

Раздел 4. Стандартные классы .NET

Лекция 23.

Класс Array, ArrayList (2 часа).

Лекция 24.

Классы работы с файловой системой (2 часа).

Лекция 25.

Стандартные интерфейсы в .NET (2 часа).

Лекция 26.

LINQ (2 часа).

Лекция 27.

Работа с XML. Сериализация и десериализация (2 часа).

Раздел 5. Шаблоны проектирования

Лекция 28.

Структурные шаблоны: Адаптер, Декоратор, Фасад, Заместитель (2 часа).

Лекция 29.

Поведенческие шаблоны: Наблюдатель, Стратегия, Состояние, Итератор, Шаблонный метод (2 часа).

Лекция 30.

Порождающие шаблоны: Абстрактная фабрика, Фабричный метод, Одиночка (2 часа).

4.1.2.2. Перечень практических занятий

Не планируется.

4.1.2.3. Перечень лабораторных работ

Семестр 2

Раздел 1. Основы программирования на языке C#

Лабораторная 1.

Линейные программы на языке C# (4 часа).

Лабораторная 2.

Операторы ветвления в языке C# (4 часа).

Лабораторная 3.

Массивы и строки в языке C# (4 часа).

Лабораторная 4.

Разработка приложений Windows Forms (4 часа).

Раздел 2. Введение в ООП

Лабораторная 5.

Введение в классы (4 часа).

Лабораторная 6.

Создание простейшего класса "структура-пара" (4 часа).

Лабораторная 7.

Полиморфизм методов класса (4 часа).

Лабораторная 8.

Наследование (4 часа).

Лабораторная 9.

Виртуальные методы классов (4 часа).

Семестр 3

Раздел 3. Введение в ООП

Лабораторная 10.

Делегаты и события в классах (4 часа).

Лабораторная 11.

Интерфейсы (4 часа).

Раздел 4. Стандартные классы .NET

Лабораторная 12.

Реализация стандартных интерфеqсов .NET (4 часа).

Лабораторная 13.

Использование LINQ для обработки коллекций (4 часа).

Раздел 5. Шаблоны проектирования

Лабораторная 14.

Шаблон проектирования Декоратор (4 часа).

Лабораторная 15.

Шаблон проектирования Адаптер (4 часа).

Лабораторная 16.

Шаблон проектирования Наблюдатель (4 часа).

Лабораторная 17.

Шаблон проектирования Стратегия (4 часа).

Лабораторная 18.

Шаблон проектирования Состояние (4 часа).

Лабораторная 19.

Шаблон проектирования Фабрика (4 часа).

4.1.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Понятие объекта и класса.
2. Основные принципы ООП.
3. Абстрагирование.
4. Инкапсуляция.
5. Агрегирование.
6. Объектная модель программы. У.
7. ниверсальный язык моделирования UML.
8. Метаданные.
9. Промежуточный код (Intermediate Language).
10. Единая среда выполнения (Common Language Runtime).
11. Пространства имен (namespaces).
12. Единая библиотека типов (классов, интерфейсов, структур) платформы - Microsoft Framework Library, основные пространства имен.
13. Встроенный язык поисковых запросов LINQ.
14. Конструкторы класса.
15. Свойства класса.
16. втоматически реализуемые свойства. Инициализация объектов класса.
17. Индексаторы. Статические поля и методы класса.
18. Переопределение операций класса. Определение преобразования типов.
19. Отношение вложенности. Отношение наследования.
20. Описание производных классов. Конструкторы производного класса.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.1.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

Не планируется.

4.1.2.6. Примерный перечень тем курсовых работ (проектов)

Не планируется.

4.2 Форма обучения: заочная

Уровень базового образования: среднее общее.

Срок обучения 5л.

Семестр	Трудоемкость, час./ зач. ед.	Лекции, час.	Практические занятия, час.	Лабораторные работы, час.	Консультация, час.	Контроль,час.	Всего (контактная работа), час.	СРС, час.	Форма промежуточного контроля (экз., зач., зач. с оц.)
3	**252 / 7**	**10**	**4**	**16**	**5**	**0,6**	**35,6**	**212,65**	**Экз.(3,75)**
4	**252 / 7**	**10**	**4**	**12**	**5**	**0,6**	**31,6**	**211,75**	**Экз.(8,65)**
Итого	**504 / 14**	**20**	**8**	**28**	**10**	**1,2**	**67,2**	**424,4**	**12,4**

4.2.1. Структура дисциплины

№ п\п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Основы программирования на языке C#	3	4	4	8					198	Тестирование
2	Введение в ООП	3	6		8					14,65	Тестирование
Всего за семестр		252	10	4	16	+		5	0,6	212,65	Экз.(3,75)
3	Введение в ООП	4	4	2	8					159,75	Тестирование
4	Стандартные классы .NET	4	6		4					52	Тестирование
5	Шаблоны проектирования	4		2						0	Тестирование
Всего за семестр		252	10	4	12	+		5	0,6	211,75	Экз.(8,65)
Итого		504	20	8	28			10	1,2	424,4	12,4

4.2.2. Содержание дисциплины

4.2.2.1. Перечень лекций

Семестр 3

Раздел 1. Основы программирования на языке C#

Лекция 1.

Введение в платформу .NET (2 часа).

Лекция 2.

Основные понятия языка C# (2 часа).

Раздел 3. Введение в ООП

Лекция 3.

Классы и объекты в языке C# (2 часа).

Лекция 4.

Данные классов. Методы. Конструкторы. Свойства (2 часа).

Лекция 5.

Наследование (2 часа).

Семестр 4

Раздел 3. Введение в ООП

Лекция 6.

Класс object. Преобразования типов (2 часа).

Лекция 7.

Интерфейсы (2 часа).

Раздел 4. Стандартные классы .NET

Лекция 8.

Введение в технологию Windows Forms (2 часа).

Лекция 9.

Виртуальные методы. Позднее связывание (2 часа).

Лекция 10.

Стандартные интерфейсы в .NET (2 часа).

4.2.2.2. Перечень практических занятий

Семестр 3

Раздел 1. Основы программирования на языке C#

Практическое занятие 1.

Линейные программы на языке C# (2 часа).

Практическое занятие 2.

Разработка приложений Windows Forms (2 часа).

Семестр 4

Раздел 3. Введение в ООП

Практическое занятие 3.

Создание простейшего класса "структура-пара" (2 часа).

Раздел 5. Шаблоны проектирования

Практическое занятие 4.

Использование LINQ для обработки коллекций (2 часа).

4.2.2.3. Перечень лабораторных работ

Семестр 3

Раздел 1. Основы программирования на языке C#

Лабораторная 1.

Операторы ветвления в языке C# (4 часа).

Лабораторная 2.

Массивы и строки в языке C# (4 часа).

Раздел 2. Введение в ООП

Лабораторная 3.

Введение в классы (4 часа).

Лабораторная 4.

Наследование (4 часа).

Семестр 4

Раздел 3. Введение в ООП

Лабораторная 5.

Виртуальные методы классов (4 часа).

Лабораторная 6.

Делегаты и события в классах (4 часа).

Раздел 4. Стандартные классы .NET

Лабораторная 7.

Реализация стандартных интерфеqсов .NET (4 часа).

4.2.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Делегаты. события.
2. Делегаты. Анонимные методы. Лямбда выражения.
3. LINQ.
3. Обработка исключений. Класс Exception.
4. Работа с XML. Сериализация и десериализация.
5. Структурные шаблоны: Адаптер, Декоратор, Фасад, Заместитель.
6. Поведенческие шаблоны: Наблюдатель, Стратегия, Состояние, Итератор, Шаблонный метод.
7. Порождающие шаблоны: Абстрактная фабрика, Фабричный метод, Одиночка.
8. Понятие объекта и класса.
9. Основные принципы ООП.
10. Абстрагирование.
11. Инкапсуляция.
12. Агрегирование.
13. Объектная модель программы.
14. Универсальный язык моделирования UML.
15. Метаданные.
16. Промежуточный код (Intermediate Language).
17. Единая среда выполнения (Common Language Runtime).
18. Пространства имен (namespaces).
19. Единая библиотека типов (классов, интерфейсов, структур) платформы - Microsoft Framework Library, основные пространства имен.
20. Автоматически реализуемые свойства. Инициализация объектов класса.
21. Индексаторы. Статические поля и методы класса.
22. Переопределение операций класса. Определение преобразования типов.
23. Отношение вложенности. Отношение наследования.
24. Описание производных классов. Конструкторы производного класса.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.2.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

Примерные темы контрольных работ:

1. Разработка структуры классов «Сотовая связь».
2. Разработка структуры классов «Салон красоты».
3. Разработка структуры классов «Адресная книга».
4. Разработка структуры классов «Телефонная книга».
5. Разработка структуры классов «Автосервис».
6. Разработка структуры классов «Магазин игрушек».
7. Разработка структуры классов «Библиотека».

8. Разработка структуры классов «Журнал студента».
9. Разработка структуры классов «Такси».
10. Разработка структуры класса «Матрица».
11. Разработка структуры классов «Продуктовый магазин».
12. Разработка структуры классов «Животные».

4.2.2.6. Примерный перечень тем курсовых работ (проектов)

Не планируется.

5. Образовательные технологии

При проведении лекционных занятий материал дается в устной форме с применением мультимедиа-проектора. На лабораторных занятиях каждый студент получает индивидуальное задание. Все задания носят практический характер и предполагают написание программного кода на языке высокого уровня. Активно применяются информационно-коммуникационные технологии.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Фонды оценочных материалов (средств) приведены в приложении.

7. Учебно-методическое и информационное обеспечение дисциплины.

7.1. Основная учебно-методическая литература по дисциплине

1. Зыков, С. В. Введение в теорию программирования. Объектно-ориентированный подход : учебное пособие / С. В. Зыков. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 187 с. — ISBN 978-5-4497-0926-4. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: https://www.iprbookshop.ru/102007.html (дата обращения: 18.10.2021). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/102007.html

2. Зайцев, М. Г. Объектно-ориентированный анализ и программирование : учебное пособие / М. Г. Зайцев. — Новосибирск : Новосибирский государственный технический университет, 2017. — 84 с. — ISBN 978-5-7782-3308-9. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: https://www.iprbookshop.ru/91284.html (дата обращения: 18.10.2021). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/91284.html

7.2. Дополнительная учебно-методическая литература по дисциплине

1. Ланских, Ю. В. Основы объектно-ориентированного и компонентно-ориентированного программирования в C# : учебно-методическое пособие для студентов, обучающихся по направлению «Прикладная математика и информатика» / Ю. В. Ланских, Л. В. Пешнина. — Соликамск : Соликамский государственный педагогический институт, 2017. — 84 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: https://www.iprbookshop.ru/86557.html (дата обращения: 18.10.2021). — Режим доступа: для авторизир. пользователей - https://www.iprbookshop.ru/86557.html

2. Маляров, А. Н. Объектно-ориентированное программирование : учебник для СПО / А. Н. Маляров. — Саратов : Профобразование, 2021. — 331 с. — ISBN 978-5-4488-1238-5. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: https://www.iprbookshop.ru/106837.html (дата обращения: 18.10.2021). — Режим доступа: для авторизир. пользователей. - DOI: https://doi.org/10.23682/106837 - https://www.iprbookshop.ru/106837.html

7.3. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

В образовательном процессе используются информационные технологии, реализованные на основе информационно-образовательного портала института (www.mivlgu.ru/iop), и инфокоммуникационной сети института:

- предоставление учебно-методических материалов в электронном виде;
- взаимодействие участников образовательного процесса через локальную сеть института и Интернет;
- предоставление сведений о результатах учебной деятельности в электронном личном кабинете обучающегося.

Информационные справочные системы:

- электронная библиотечная система "BOOK.ru" (http://book.ru/);
- электронная библиотечная системы "IPRBooks" (http://www.iprbookshop.ru/);
- электронная библиотечная система "iBooks.ru" (http://www.ibooks.ru/);
- библиотека MSDN: http://msdn.microsoft.com

Программное обеспечение:

Microsoft Visual Studio (Программа Microsoft Azure Dev Tools for Teaching (Order Number: IM126433))

7.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

iprbookshop.ru
book.ru
ibooks.ru
msdn.microsoft.com
mivlgu.ru/iop

8. Материально-техническое обеспечение дисциплины

Лаборатория системного и прикладного программирования

6 шт. компьютеров Intel Core i5, 3500 MHz/ ОЗУ 4гб/ LG 21’; 6 шт. персональных компьютеров Digitech (комплект 2); проектор NEC V300X 3D; экран проекционный настенный Lumien Master Picture; маршрутизатор Gigabit Switch TEG-S16S; макет системы мобильного мониторинга; лабораторный стенд для изучения микроконтроллера; роботизированная платформа IE-POP-BOT; аппаратно-программный комплекс «Изучение принципов построения и исследования инфокоммутационных локальных сетей». Маркерная доска. Доступ к сети Интернет.

9. Методические указания по освоению дисциплины

Для успешного освоения теоретического материала обучающийся: знакомится со списком рекомендуемой основной и дополнительной литературы; уточняет у преподавателя, каким дополнительным пособиям следует отдать предпочтение; ведет конспект лекций и прорабатывает лекционный материал, пользуясь как конспектом, так и учебными пособиями.

До выполнения лабораторных работ обучающийся изучает соответствующий раздел теории. Перед занятием студент знакомится с описанием заданий для выполнения работы, внимательно изучает содержание и порядок проведения лабораторной работы. Лабораторная работа проводятся в компьютерном классе. Обучающиеся выполняют индивидуальную задачу компьютерного моделирования в соответствии с заданием на лабораторную работу. Полученные результаты исследований сводятся в отчет и защищаются по традиционной методике в классе на следующем лабораторном занятии. Необходимый теоретический материал, индивидуальное задание, шаги выполнения лабораторной работы и требование к

отчету приведены в методических указаниях, размещенных на информационно-образовательном портале института.

Самостоятельная работа оказывает важное влияние на формирование личности будущего специалиста, она планируется обучающимся самостоятельно. Каждый обучающийся самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием дисциплины. Он выполняет внеаудиторную работу и изучение разделов, выносимых на самостоятельную работу, по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

Форма заключительного контроля при промежуточной аттестации – экзамен. Для проведения промежуточной аттестации по дисциплине разработаны фонд оценочных средств и балльно-рейтинговая система оценки учебной деятельности студентов. Оценка по дисциплине выставляется в информационной системе и носит интегрированный характер, учитывающий результаты оценивания участия студентов в аудиторных занятиях, качества и своевременности выполнения заданий в ходе изучения дисциплины и промежуточной аттестации.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению *09.03.04 Программная инженерия* и профилю подготовки *Методы и средства разработки программного обеспечения*
Рабочую программу составил *к.т.н., доцент каф. ПИН Привезенцев Д.Г.*_____

Программа рассмотрена и одобрена на заседании кафедры *ПИн*

протокол № 11 от 05.05.2022 года.
Заведующий кафедрой *ПИн* _____*Жизняков А.Л.*
(Подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии факультета

протокол № 4 от 12.05.2022 года.
Председатель комиссии ФИТР _____ *Рыжкова М.Н.*
(Подпись) (Ф.И.О.)

Лист актуализации рабочей программы дисциплины

Программа одобрена на_____учебный год.

Протокол заседания кафедры № _____от _____20__ года.

Заведующий кафедрой_____ _____ _____
(Подпись) (Ф.И.О.)

Программа одобрена на_____учебный год.

Протокол заседания кафедры № _____от _____20__ года.

Заведующий кафедрой_____ _____ _____
(Подпись) (Ф.И.О.)

Программа одобрена на_____учебный год.

Протокол заседания кафедры № _____от _____20__ года.

Заведующий кафедрой_____ _____ _____
(Подпись) (Ф.И.О.)

Приложение

Фонд оценочных материалов (средств) по дисциплине

Объектно-ориентированное программирование

1. Оценочные материалы для проведения текущего контроля успеваемости по дисциплине

Семестр 3.

Рейтинг-контроль №1

1.Определение класса. Данные и методы класса.
2. Конструктор и деструктор класса. Определение. Виды конструкторов.
3. Указатели на компонентные функции.
4. Объекты класса.
5. Массив объектов класса.
6. Статические члены класса.
7. Наследование. Простое и множественное наследование
8. Конструкторы производных классов. Особенности вызова
9. Виртуальные функции. Назначение
10. Абстрактные классы. Чистые виртуальные функции.

Рейтинг-контроль №2

10. Абстрактные классы. Чистые виртуальные функции.
11. Базовые и производные классы. Атрибуты наследования.
12. Перегрузка операций. Назначение
12. Какие операции не перегружаются
13. Перегрузка унарных операций в C# функцией-членом класса
14. Перегрузка унарных операций в C# дружественной функцией
15. Перегрузка бинарных операций в C# функцией-членом класса
16. Перегрузка бинарных операций в C# дружественной функцией
17. Какие виды преобразований можно реализовать в C#
18. Преобразование из типа класса к скалярному типу
19. Преобразование из скалярного типа к типу класса
20. Преобразован из из типа одного класса к типу другого класса.
21. Конструкторы преобразований. Назначение
22. Операции преобразований. Назначений.

Рейтинг-контроль №3

1. Преимущества перед процедурным программированием.
2. Принципы разработки иерархии классов и ее использования.
3. Разрешение вопросов, связанных с неоднозначностью доступа.
4. Иерархия классов, используемая в среде Microsoft Visual Studio.
5. Программирование на C# 10.
6. Объектно-ориентированное программирование как основа современного программирования.
7. Совместное использование функций и перегрузка операторов.
8. Использование динамической памяти.
9. Механизм исключительных ситуаций.
10.Поддержка модульности. Разбиение программы на единицы компиляции.
11. Шаблонные классы и стандартные контейнеры.
12. Организация пользовательского интерфейса.

Семестр 3.

Вопросы к устному зачету.

Рейтинг-контроль №1

1. Новейшие направления в области создания технологий программирования. Законы эволюции программного обеспечения.

2. Программирование в средах современных информационных систем: создание модульных программ, элементы теории модульного программирования, объектно-ориентированные проектирование и программирование

3. Среда Net. Framework. Общеязыковая исполняющая среда (CLR).

4. Объектно-ориентированный подход к проектированию и разработке программ. Сущность объектно-ориентированного подхода

5. Инкапсуляция. Понятие класса. Управление доступом к элементам класса. Конструкторы и деструкторы. Переменные объектного типа.

6. Друзья класса.

7. Наследование. Базовый класс. Управление доступом при наследовании.

8. Простое наследование. Виртуальные функции.

9. Чистые виртуальные функции. Абстрактные классы. Непрямые базовые классы. Виртуальные деструкторы. Виртуальные базовые классы

10. Полиморфизм.

Рейтинг-контроль №2

11. Методы общего полиморфизма: перегрузка операций. Реализация перегруженной операции. Перегрузка операции присваивания, инкремента, декремента, бинарных арифметических операций.

12. Методы общего полиморфизма: перегрузка операций. Перегрузка операций ввода-вывода в потоки, индексации массивов, операций выделения и освобождения памяти.

13. Методы общего полиморфизма: Преобразования типов, определяемые классом.

14. Методы общего полиморфизма: перегрузка функций. Чистый полиморфизм.

15. Параметрический полиморфизм. Шаблоны функций.

16. Параметрический полиморфизм. Шаблоны классов

Рейтинг-контроль №3

17. Классы в C#. Управление доступом. Конструкторы. Деструкторы. Наследование.

18. Свойства классов в C#. Скалярные и индексированные свойства.

19. Перегрузка операций в C#. Преобразование типов в C#.

20 Делегаты. события. Интерфейсы

21. Концепции программирования для Windows. Структура Windows программ. Использование Windows Forms для создания приложений с графическим интерфейсом пользователя.

22. Создание элемента управления Windows Forms

Общее распределение баллов текущего контроля по видам учебных работ для студентов

Рейтинг-контроль 1	тест	до 5 баллов
Рейтинг-контроль 2	тест	до 5 баллов
Рейтинг-контроль 3	тест	до 5 баллов
Посещение занятий студентом	Отметка в журнале посещений	до 5 баллов
Дополнительные баллы (бонусы)		0
Выполнение семестрового плана самостоятельной работы	Защита лабораторных работы	до 4 баллов за каждую работу

2. Промежуточная аттестация по дисциплине

Перечень вопросов к экзамену / зачету / зачету с оценкой.
Перечень практических задач / заданий к экзамену / зачету / зачету с оценкой (при наличии)

ОПК-6.1

Структурные единицы, из которых состоит программа

A. Модули
B. Переменные
C. Циклы
D. Условия

Абстрактный тип данных, поставляемый с возможно частичной реализацией это:

A. Класс
B. Объект
C. Процедура
D. Модуль

В разделе Аксиомы спецификации абстрактного типа данных перечисляются

A. Свойства значений функции
B. Значения функций
C. Перечисляются типы функций
D. Приводится код функций

В разделе Предусловий спецификации абстрактного типа данных задаются:

A. Области частичных функций
B. Условия выполнения методов объектов
C. Предварительные описания ветвлений
D. Все из перечисленного

В разделе Функции спецификации абстрактного типа данных перечисляются:

A. Операции, применимые к экземплярам данного абстрактного типа данных
B. Операции, применимые к данному типу данных
C. Реализация основных функций модуля
D. Ничего из перечисленного

Всякий раз, когда система программного обеспечения должна поддерживать множество альтернатив, их полный список должен быть известен только одному модулю системы

A. Принцип единственного выбора
B. Принцип Открытости
C. Принцип модульной защищенности
D. Принцип модульной достоверности

В чем заключается принцип группирования подпрограмм?

A. Самодостаточный, повторно используемый модуль должен включать множество подпрограмм, обеспечивающих операции создания, включения, удаления, поиска.
B. Повторно используемые модули должны группироваться в библиотеки
C. Подпрограммы группируются по функциональному признаку
D. Подпрограммы группируются по размеру кода

В чем заключается принцип изменчивости типов?

A. Необходимо средство для описания модулей, в которых типы выступают в роли параметров.
B. Типы переменных должны изменяться во время выполнения программы

C. Вызываемые модели связаны только с определенными типами переменных.

D. Типы связей между модулями могут быть выбраны из списка

В чем заключается принцип независимости представлений?

A. Общая структура повторно используемого модуля должна позволять модулям-клиентам определять свои действия при отсутствии сведений о реализации модуля.

B. Общая методика создания и применения повторно используемых модулей должна поддерживать идею семейства модулей

C. Типы связей между модулями могут быть выбраны из списка

D. Шаблон подпрограммы предполагает наличие в таблицах объектов перечислимого типа

Выпуск ПО в нужный момент, то есть тогда или незадолго до того, как у пользователей появилась соответствующая потребность в подобной системе это:

A. Желательность.

B. Своевременность.

C. Функциональность.

D. Экономичность.

Для улучшения расширяемости важно обеспечить:

A. Децентрализацию

B. Максимально большое количество связей между моделями

C. Простую архитектуру

D. "Заморозку" требований к ПО после завершения начальной ступени анализа

Класс может быть доступен:

A. только одному клиенту.

B. всем клиентам.

C. ни одному клиенту.

D. избранным клиентам.

Класс называется отложенным, если:

A. он реализован не полностью

B. выполнение его объектов отложено на фиксированное время

C. у него нет общих свойств с другими классами

D. объекты этого класса не имеют методов

Класс это:

A. абстрактный тип данных, снабженный некоторой (возможно частичной) реализацией

B. это совокупность функций, выполняющих определенные действия

C. Потомок абстрактного типа данных , снабженный некоторой (возможно частичной) реализацией

D. ничего из перечисленного

Легкость, с которой люди с различными знаниями и квалификацией могут научиться использовать ПО и применять его для решения задач.

A. Модифицируемость

B. Простота использования

C. Эффективность ПО.

D. Функциональность.

Метакласс это:

A. класс, экземпляры которого сами являются классами
B. класс, который не имеет потомков
C. класс с фиксированным числом потомков
D. класс, экземпляры которого сами являются процедурными абстракциями

Метод проектирования удовлетворяет этому критерию, если он помогает разложить задачу на несколько менее сложных подзадач, объединяемых простой структурой, и настолько независимых, что в дальнейшем можно отдельно продолжить работу над каждой из них

A. Декомпозиция
B. Модульность
C. Поломорфизм
D. Самодостаточность

Метод разработки ПО, который строит архитектуру всякой программной системы на модулях, выведенных из типов объектов, с которыми система работает (а не на одной или нескольких функциях, которые она должна предоставлять)

A. ОО-конструирование
B. Процедурное конструирование
C. Модульное конструирование
D. Функциональное конструирование

Метод удовлетворяет этому критерию, если незначительное изменение спецификаций разработанной системы приведет к изменению одного или небольшого числа модулей

A. Модульная непрерывность
B. Модульная композиция
C. Модульная Инкапсуляция
D. Связанная модернизация

Метод удовлетворяет этому критерию, если он обеспечивает разработку элементов программного продукта, свободно объединяемых между собой для получения новых систем, быть может, в среде, отличающейся от той, для которой эти элементы первоначально разрабатывались.

A. Модульная композиция
B. Декомпозиция
C. Универсальность
D. Связывание

Метод удовлетворяет этому критерию, если он помогает получить такую программу, читая которую можно понять содержание каждого модуля, не зная текста остальных, или, в худшем случае, ознакомившись лишь с некоторыми из них.

A. Модульная понятность
B. Модульная композиция
C. Документированность
D. Объектная ориентированность

Метод удовлетворяет этому критерию, если он приводит к архитектуре системы, в которой аварийная ситуация, возникшая во время выполнения модуля, ограничится только этим модулем, или, в худшем случае, распространится лишь на несколько соседних модулей

A. Модульная защищенность
B. Модульная достоверность
C. Обработчик исключений
D. Модульная спецификация

Механизм, делающий определенные компоненты недоступными для клиентов:

A. Шифрование информации.
B. Скрытие информации.
C. Авторизация доступа.
D. Удаление мусора в памяти.

Объект, принадлежащий множеству объектов, описываемых спецификацией абстрактного типа данных называется:

A. экземпляром этого абстрактного типа данных
B. потомком этого абстрактного типа данных
C. прототипом этого абстрактного типа данных
D. функцией этого абстрактного типа данных

ОО-язык и ОО-среда, вместе с поддерживающим их методом, должны быть применимы ко всему жизненному циклу, минимизируя сложность переходов между последовательными шагами. Это определение относится к:

A. Бесшовности
B. Классовости
C. Пошаговой реализации.
D. Запланированной модернизации.

Отметьте требования к модульным структурам, которые позволяют создавать компоненты для повторного использования

A. Независимость представлений
B. Факторизация общего поведения
C. Неизменность типов
D. Закрытость модулей
E. Неформальное описание ресурсов

Отметьте требования к модульным структурам, которые позволяют создавать компоненты для повторного использования

A. Изменчивость типов
B. Группирование подпрограмм
C. Изменчивость реализаций
D. Преобразование переменных
E. Непрерывность разработки

Перегрузка это:

A. Связывание с одним именем более одного содержания
B. Динамическая загрузка в программу модулей
C. Недостаток ресурсов компьютера
D. Увеличение количества сваязей между подпрограммами

Переносимость это:

A. Легкость переноса ПО в различные программные и аппаратные среды.
B. Совместимость с различными типами оборудования и сред выполнения.
C. легкость сочетания однох элементов ПО с другими.
D. Способность оборудования работать в различных условиях эксплуатации.

Повторное использование может обеспечить прогресс на следующих направлениях:

A. Своевременность
B. Сокращение объема работ по сопровождению ПО
C. Надежность
D. Верифицируемость

E. Устойчивость

Повторное использование может обеспечить прогресс на следующих направлениях:
A. Эффективность
B. Совместимость
C. Инвестирование
D. Ограниченность
E. Корректность

Подходы обеспечения совместимости включают:
A. Стандартные форматы файлов
B. Стандартные структуры данных
C. Стандартные пользовательские интерфейсы
D. Стандартные сообщения программы
E. Стандартные носители данных

Пользователями ПО являются
A. Люди, взаимодействующие с конечным продуктом
B. Люди, занимающиеся администрирование программного продукта
C. Люди, закупающие программный продукт
D. Все три категории людей

Расширяемость это
A. Легкость адаптации ПО к изменениям спецификации
B. Способность ПО соответствующим образом реагировать на аварийные ситуации
C. Принцип построения пользовательского интерфейса
D. Легкость переноса ПО в различные программные и аппаратные среды

Скрытие информации
A. Разработчик каждого модуля должен выбрать некоторое подмножество свойств модуля в качестве официальной информации о модуле, доступной авторам клиентских модулей.
B. Сторонним пользователя модуля недоступна никакая информация об интерфейсе модуля.
C. Вся информация внутри модуля зашифровывается.
D. Скрывается сам факт существования модуля от соседних модулей.

Совместимость это:
A. Легкость сочетания одних элементов ПО с другими
B. Способность ПО как можно меньше зависеть от ресурсов оборудования
C. Метод создания архитектуры системы, позволяющий проектировщику производить системы с простой и децентрализованной структурой
D. Корректное поведение системы при обработке данных различных типов

Способность ПО выполнять точные задачи так, как они определены их спецификацией
A. Корректность
B. Условность
C. Устойчивость
D. Эффективность

Способность ПО как можно меньше зависеть от ресурсов оборудования: процессорного времени, пространства, занимаемого во внутренней и внешней памяти, пропускной способности, используемой в устройствах связи.

A. Совместимость.
B. Эффективность.
C. Достаточность.
D. Децентрализованность.
E. Переносимость.

Способность ПО соответствующим образом реагировать на аварийные ситуации
A. Устойчивость
B. Расширяемость
C. Верифицируемость
D. Эффективность

Способность присоединять к сущности объекты различных возможных типов:
A. Полиморфизм.
B. Инкапсуляция.
C. Наследование.
D. Переопределение.

Способность элементов ПО служить для построения многих различных приложений это
A. Повторное использование
B. Расширяемость
C. Децентрализация
D. Целосность

Степень возможностей, обеспечиваемых системой это:
A. Функциональность.
B. Проектируемость.
C. Эффективность.
D. Стандартность.

Функция из исходного множества X в результирующее множество Y является
A. Частичной
B. Полной
C. Частичной, если она определена не для всех элементов X
D. Частичной, если она не реализует все методы подмножества Y

Этот вид проектирования помогает проектировщикам создать систему, состоящую из автономных элементов с простыми и согласованными структурными связями между ними.
A. Модульное.
B. Процедурное.
C. Объектно-ориентированное.
D. Защищенное.

Этот модуль доступен для расширения
A. Открытый модуль
B. Закрытый модуль
C. Изменяемый модуль
D. Структурированный модуль

Этот модуль доступен только для использования другими модулями
A. Закрытый модуль
B. Открытый модуль
C. Унифицированный модуль

D. Связанный модуль

Описание, основанное на типах объектов, с течением времени обеспечивает лучшую устойчивость и лучшие возможности для повторного использования, чем описание, основанное на анализе функций системы.

Верно
Неверно

Функциональное проектирование сверху вниз не подходит для программных систем с долгим жизненным циклом, включающим их изменения и повторное использование.

Верно
Неверно

Задайте соответствие названиям правил и их определениям

1. Прямое отображение Модульная структура, создаваемая в процессе конструирования ПО, должна оставаться совместимой с модульной структурой, создаваемой в процессе моделирования проблемной области.
2. Минимум интерфейсов Каждый модуль должен поддерживать связь с возможно меньшим числом других модулей.
3. Слабая связность интерфейсов Если два модуля общаются между собой, то они должны обмениваться как можно меньшим объемом информации.
4. Явные интерфейсы Всякое общение двух модулей A и B между собой должно быть очевидным и отражаться в тексте A и/или B.

Задайте соответствия терминам и определениям

1. Программный элемент, который может быть вызван другими элементами для выполнения некоторого алгоритма, используя некоторые входные данные, создавая некоторые выходные данные, и, возможно, модифицируя другие данные. Подпрограммы
2. Описание каждого такого элементы содержит ряд объявлений связанных с ним компонентов. Он также может точно определять права доступа, ограничивающие использование его компонентов другими элементами. Пакеты
3. Функционально законченный фрагмент программы, оформленный в виде отдельного файла с исходным кодом или поименованной непрерывной её части, предназначенный для использования в других программах. Модуль
4. Некоторая сущность в виртуальном пространстве, обладающая определённым состоянием и поведением, имеющая заданные значения свойств и операций над ними. Объект

Аксиомы и предусловия выражают _____ данного типа

A. Семантику
B. Синтаксис
C. Относительную зависимость
D. Непротиворечивость

В классе, реализующем абстрактный тип данных, функции становятся операциями, применимыми к экземплярам класса. Это

A. Компоненты
B. Объекты
C. Процедуры
D. Безусловными переходами

В чем заключается Объектный принцип?

A. Каждый объект является экземпляром некоторого класса
B. Каждый класс является экземпляром некоторого объекта

C. Функции являются потомками объектов
D. Объекты являются составной частью моделей

В чем заключается принцип изменчивости реализаций
A. Общая методика создания и применения повторно используемых модулей должна поддерживать идею семейства модулей
B. Самодостаточный, повторно используемый модуль должен включать множество подпрограмм, обеспечивающих операции создания, включения, удаления, поиска
C. Должен быть предоставлен выбор реализации в зависимости от параметров
D. Общая методика создания и применения повторно используемых модулей не поддерживает семейства модулей

Вычисление включает в себя следующие ингредиенты
A. Процессоры (потоки управления)
B. Действия (функции)
C. Данные (объекты)
D. Классы
E. Алгоритмы
F. Условия

Для конструирования ОО-программ
A. реализации абстрактного типа данных должны быть полными
B. реализации абстрактного типа данных могут быть неполными
C. реализации абстрактного типа данных должны быть неполными

Какие виды спецификаций абстрактных типов данных являются неявными?
A. Метод абстрактного типа данных определяет неявно некоторое множество объектов, задавая применимые к ним функции.
B. Функции абстрактного типа данных определяются неявно
C. Верны оба утверждения
D. Верных ответов нет

Какой это принцип: "Клиент должен иметь возможность доступа к свойствам объекта, используя одинаковую нотацию, вне зависимости от того, как это свойство реализовано - в памяти или как результат вычислений"?
A. Принцип унифицированного доступа
B. Принцип модульности
C. Принцип однородности памяти
D. Принцип ОО-управления

Класс является:
A. одновременно модулем и типом
B. модулем
C. типом
D. ни модулем, ни типом

Компоненты, представленные во времени, для доступа к которым требуется описать некоторые вычисления (алгоритмы), применимые далее ко всем экземплярам данного класса называются
A. атрибутами
B. функциями
C. методами
D. процедурами

Компоненты, представленные в пространстве и ассоциируемые с некоторой частью информации каждого экземпляра класса называются

A. атрибутами
B. функциями
C. методами
D. процедурами

Метод модульного проектирования должен удовлетворять требованиям:

A. Декомпозиции
B. Композиции
C. Понятности
D. Непрерывности
E. Защищенности
F. Связности
G. Скрытия

На что распространяется объектный принцип?

A. На составные объекты, определяемые разработчиками
B. На целые и действительные числа
C. На булевы значения и символы
D. На все перечисленное

Не влияет на форму записи исходных текстов программ и не определяет их функциональность В чем заключается принцип изменчивости реализаций?

A. Общая методика создания и применения повторно используемых модулей должна поддерживать идею семейства модулей
B. Самодостаточный, повторно используемый модуль должен включать множество подпрограмм, обеспечивающих операции создания, включения, удаления, поиска
C. Должен быть предоставлен выбор реализации в зависимости от параметров
D. Не влияет на форму записи исходных текстов программ и не определяет их функциональность

Отметьте принципы конструирования ПО

A. Принцип Лингвистических Модульных Единиц
B. Принцип Самодокументирования
C. Принцип Унифицированного Доступа
D. Принцип Открыт-Закрыт
E. Принцип Единственного выбора
F. Принцип Скрытия информации
G. Принцип Соответствия
H. Принцип Явных интерфейсов

Процесс объединения классов в систему должен:

A. идти снизу-вверх
B. идти сверху-вниз
C. идти параллельно по всем классам
D. любое утверждение верно

Разбиение на модули

A. влияет на форму записи исходных текстов программ, но не определяет их функциональность

B. на форму записи исходных текстов программ и определяет их функциональность

C. не влияет на форму записи исходных текстов программ, но определяет их функциональность

Спецификация абстрактного типа данных является:

A. является формальным математическим описанием, а не текстом программы

B. является текстом программы, а не формальным математическим описанием

C. является и формальным математическим описанием, и текстом программы

D. не является формальным математическим описанием, и не является текстом программы

Спецификация абстрактных типов данных состоит из следующих разделов

A. Типы
B. Функции
C. Аксиомы
D. Предусловия
E. Условия
F. Заголовки

Статическое описание определенных динамических объектов - элементов данных, которые обрабатываются во время выполнения программной системы

A. Типы
B. Объекты
C. Процедуры
D. Переменные

Функции бывают следующих категорий

A. Функции-конструкторы
B. Функции-запросы
C. Функции-команды
D. Функции-сигнатуры
E. Функции-модели

Чем является класс?

A. Модулем
B. Типом
C. Модулем и типом
D. ничем из перечисленного

Чем являются модули и типы?

A. Модули и типы являются синтаксической концепцией
B. Модули и типы являются семантической концепцией
C. Модули - семантическая концепция, типы - синтаксическая концепция
D. Модули - синтаксическая концепция, типы - семантическая концепция

Чтобы получить надлежащие описания объектов, наш метод должен удовлетворять условиям:

A. Описания должны быть точными и недвусмысленными.
B. Описания должны быть полными.
C. Описания не должны быть излишне специфицированы.

D. Должно быть малое количество описаний.
E. Описание не применимы к абстрактным типам данных.

Что содержится в Открытой и Секретной части абстрактного типа данных?
A. Спецификация - открытая часть; Выбор представления - секретная часть; Реализация функций посредством свойств - Секретная часть.
B. Спецификация - открытая часть; Выбор представления - открытая часть; Реализация функций посредством свойств - Секретная часть.
C. Спецификация - Секретная часть; Выбор представления - секретная часть; Реализация функций посредством свойств - Открытая часть.
D. Спецификация - открытая часть; Выбор представления - секретная часть; Реализация функций посредством свойств - Открытая часть.

Эти правила должны соблюдаться для обеспечения модульнсти
A. Прямое отображение
B. Минимум интерфейсов
C. Слабая связность интерфейсов
D. Явные интерфейсы
E. Инкапсуляция
F. Унифицированный доступ
G. Обработка исключений
H. Композиция модулей

Методические материалы, характеризующих процедуры оценивания

На основе перечня вопросов формируются индивидуальные задания для студентов: 4 вопроса из блока 1, 3 вопроса из блока 2, 3 вопроса из блока 3. Результатом итогового контрольного теста является балл, рассчитанный на основе количества правильных ответов. С учетом индивидуального семестрового рейтинга студента формируется итоговый балл по курсу.

Максимальная сумма баллов, набираемая студентом по дисциплине равна 100.

Оценка в баллах	Оценка по шкале	Обоснование	***Уровень сформированности компетенций***
Более 80	«Отлично»	Содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	***Высокий уровень***
66-80	«Хорошо»	Содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения	***Продвинутый уровень***

		ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	
50-65	«Удовлетворительно»	Содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки	***Пороговый уровень***
Менее 50	«Неудовлетворительно»	Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки	***Компетенции не сформированы***

3. Задания в тестовой форме по дисциплине

Примеры заданий:

Дана строка " System.Console.WriteLine("String"); " Задайте соответствия частям строки и их определениям:

1. SystemПространство имен
2. Console Класс, определенный в пространстве имен
3. WriteLine Метод, определенный в этом классе
4. String Строка, выводимая на экран

Задайте соответствие

1. Синтаксическая перегрузкаПерегрузка подпрограмм, перегрузка операторов
2. Семантическая перегрузка Динамическое связывание
3. Универсальность Механизм определения параметризованных шаблонов модулей, параметры которых представляют собой типы

Задайте соответствие видов документации

1. Внешняя Даёт пользователям возможность понять сильные стороны системы и удобство их использования.
2. Внутренняя Даёт разработчикам возможность понять структуру и реализацию системы.
3. Описывающаа интерфейс модулей Дает возможность разработчикам понять функции без изучения реализации.

Задайте соответствие качеств ПО и их описаний

1. Верифицируемость Легкость подготовки процедур приемки, особенно тестовых данных, процедур обнаружения неполадок и трассировки ошибок на этапах заключительной проверки и введения проекта в действие.
2. Целостность Способность ПО защищать свои различные компоненты (программы, данные) от несанкционированного доступа и модификации.
3. Восстанавливаемость Способность облегчать устранение дефектов.
4. Экономичность способность системы завершиться, не превысив выделенного бюджета или даже не истратив его.

Задайте соответствие компонентов класса их ролям

1. Компонент без возвращаемого результата Команда
2. Компонент, возвращающий результат Запрос
3. Запрос с аргументами Функция
4. Запрос без аргументов, возвращающий данные из памяти Атрибут

Задайте соответствие компонентов класса способам реализации

1. Компонент, возвращающий память Атрибут
2. компонент, производящий вычисления Подпрограмма
3. Подпрограмма, возвращающая результат Функция
4. Подпрограмма, не возвращающая результат Процедура

Задайте соответствие названий и определений принципов конструирования ПО

1. Принцип Лингвистических Модульных Единиц Модули должны соответствовать синтаксическим единицам используемого языка
2. Принцип Самодокументирования Разработчик модуля должен стремиться к тому, чтобы вся информация о модуле содержалась в самом модуле
3. Принцип Унифицированного Доступа Все службы, предоставляемые модулем, должны быть доступны в нотации, которая не подведет вне зависимости от реализации, использующей память или вычисления
4. Принцип Открыт-Закрыт Принцип семафора
5. Принцип единственного выбора Всякий раз, когда система программного обеспечения должна поддерживать множество альтернатив, их полный список должен быть известен только одному модулю системы.

Задайте соответствие операциям и их обозначению на языке C#

1. Логическое И &&
2. Логическое ИЛИ ||
3. Логическое РАВНО = =
4. Логическое НЕ РАВНО ! =
5. =
6. OR
7. AND

Расположите слои в разработке ПО в правильном порядке
(первый слой - это верхний уровень)

1. 1 Приложение
2. 2 Библиотека приложения
3. 3 Другие библиотеки
4. 4. Базовая библиотека
5. 5 Библиотека ядра
6. 6 Компилятор
7. 7 Операционная система
8. 8 Оборудование

Дана строка " Console.WriteLine("Код символа {0} = {1}",sim,kod);". Какие действия будут выполнены?

A. Выведется на экран - "Код символа {0} = {1}",sim,kod
B. Выведется на экран - "Код символа = ",sim,kod
C. Выведется на экран - "Код символа = "
D. Выведется на экран сообщение "Код символа" и требование для ввода значений переменных sim, kod

Какие действия выполняет строка "Console.WriteLine("x={0} \ty={1}",x, z); " ?

A. выводит "x= y="

B. выводит "x={0} y={1}"

C. выводит "x={0} \ty={1}"

D. выводит "x={0} y={1}",x,z

Какие действия выполняет строка "int kod=Console.Read(); " ?

A. Выводит значение переменной kod

B. Записывает в текстовую переменную kod значения, полученные с консоли

C. Записывает в целочисленную переменную kod значения, полученные с консоли

D. Сравниваются значения переменной kod и Read

Какие утверждения верны?

A. Конструирование ОО-ПО - это построение программной системы как структурированной совокупности реализаций (возможно частичных) абстрактных типов данных.

B. Конструирование ОО-ПО - это метод разработки ПО, который строит архитектуру всякой программной системы на модулях, выведенных из типов объектов, с которыми система работает (а не на одной или нескольких функциях, которые она должна предоставлять).

C. Верных утверждений нет.

D. Конструирование ОО-ПО - это построение программной системы с применением модульного программирования без использования абстрактных типов данных.

Как правильно объявить массив с явной инициализацией?

A. int [ ] q = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

B. double [1;1] q = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

C. double [;10] q = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

D. integer [] q = new { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

Набор критериев ОО-ориентации программ состоит из следующих частей:

A. Метод и язык

B. Реализация и среда

C. Библиотеки

D. Спецификации

E. Оборудование

F. Технологичность

Спецификация абстрактного типа данных является непротиворечивой, когда для всякого правильно построенного выражения ее аксиомы позволяют вывести

A. Не более одного значения

B. Одно значение

C. Более одного значения

D. Во всех случаях

Укажите правильный вариант задания Условия на языке C#

A. if (x>=0) && (x

B. if (x>=0) & (x

C. if (x>=0) && (x

D. if (x=0) || (x=2) Console.WriteLine("X")

Укажите правильный вариант задания Цикла на языке C#

A. for (x = 1; x

B. for x = 10 to x = 100 Console.WriteLine(x, y);
C. for (x = 1; x 2+0.1; x ++) Console.WriteLine(x, y);
D. for (x = 1 : 2+0.1 : x ++) Console.WriteLine(x, y);

Укажите правильный вариант объявления многомерного массива
A. int[,] myArr = new int[4, 5];
B. int[; ;] myArr = new int[4, 5];
C. double [; ;] myArr = new int[4 4 5 6 7 7];
D. int[] myArr = new (4, 5);

Спецификация абстрактного типа данных является _____ тогда и только тогда, когда для всякого правильно построенного выражения ее аксиомы позволяют вывести не более одного значения.

непротиворечивой

Полный перечень тестовых заданий с указанием правильных ответов, размещен в банке вопросов на информационно-образовательном портале института по ссылке https://www.mivlgu.ru/iop/question/edit.php?courseid=381

Оценка рассчитывается как процент правильно выполненных тестовых заданий из их общего числа.