

Федеральное государственное автономное образовательное учреждение высшего образования  
«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ имени В.И. ВЕРНАДСКОГО»  
Физико-технический институт (структурное подразделение)

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ**  
**«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»**

Направление подготовки  
09.03.04 "Программная инженерия"

Направленность программы  
Программная инженерия

## Оценочные средства для текущего контроля

Форма контроля	Вид учебной работы	Представление оценочного средства в фонде
Текущий контроль	Лекции	<p>По каждой лекции студенты должны написать краткое резюме, осветив самые важные вопросы. Устные вопросы из базы тестовых вопросов во время лекций для контроля понимания студентами материала лекции.</p> <p>Компьютерное тестирование. Полный список тестов заранее студентам не предоставляется и хранится в электронном виде.</p>
	Практические занятия	<p>Подробный перечень и описание практических работ содержатся в Практикуме по курсу ООП, изданному в электронном виде и размещенному в "Облаке" Mail.ru" и в Moodle.</p> <p>Контрольные вопросы для практических работ.</p> <p>Отчет по практической работе в электронном виде (в соответствии с образцом отчета).</p> <p>Предоставление и объяснение работы компьютерной программы, реализующей задания соответствующей работы.</p> <p>После выставления преподавателем оценки за практическую работу студент обязан предоставить электронную скан-копию отчета с оценкой преподавателя в формате (pdf).</p>
	Самостоятельная работа обучающихся: реферат, индивидуальный творческий проект	<p>Контрольные вопросы, оценка устного выступления, оценка программного кода.</p> <p>Перечень вопросов для самостоятельной работы содержатся в Практикуме по курсу ООП, изданному в электронном виде и размещенных в "облаке" Mail.ru" и Moodle.</p> <p>Текущий контроль самостоятельной работы проводится преподавателем на еженедельных консультациях.</p> <p>Оценка - дополнительные баллы по правилам, описанным в практикуме.</p>

## **Контрольные вопросы для практических работ:**

### **Практическая работа №1.**

1. Какие типы преобразований переменных вы знаете?
2. Можно ли неявно преобразовать тип Integer в Char?
3. Может быть потеряны значащие цифры при неявном преобразовании? Приведите пример.
4. Может быть потеряно значение переменной при явном преобразовании? Приведите пример.
5. Чем метод TryParse отличается от Parse? Что возвращает метод TryParse?
6. Что умеет делать класс Convert?
7. Как найти машинное эpsilon для типа Decimal? Чему оно равно?
8. Кто первым обработает событие нажатия клавиши? Форма, или элемент формы, находящийся в фокусе?
9. Запишите в виде формулы алгоритм Ньютона вычисления квадратного корня?
10. Как контролируется погрешность в методе Ньютона?
11. Как выбирается начальное приближение в методе Ньютона?
12. В чем отличие классов String и StringBuilder?
13. Как реализуется запись значений матрицы в Grid?

### **Практическая работа №2.**

1. Что такое перегрузка функций? Какое служебное слово используется для перегрузки?
2. Что такое переопределение (Override) методов?
3. Что такое статические и виртуальные методы?
4. Как можно передавать и возвращать из методов несколько значений?
5. Как можно передать в метод простой тип по ссылке?
6. Как можно передать в метод неопределенное число параметров?
7. Для чего служит метод Split?
8. Почему метод Штейна для вычисления наибольшего общего делителя работает быстрее, чем метод Евклида?

### **Практическая работа №3.**

1. Какие виды исключений вы знаете?
2. Как написать класс - потомок Exception?
3. С помощью какого служебного слова происходит выброс собственного исключения?
4. Что происходит если в методе не будет обработано исключение?
5. Как можно отследить переполнение арифметического типа?
6. Класс Exception и его методы.
7. Стратегия использования специализированных и общих исключений.
8. Что произойдет если сначала идет блок catch с общим исключением, а затем со специфическим?
9. Можно убрать все внутренние блоки catch?
10. Можно ли оставить только блок Try?
11. Если исключение в блоке Try – Catch не поймано, означает ли это крах программы.
12. Можно ли написать программу, которая никогда аварийно не завершается?
13. Какой метод предпочтительней throw или throw(ex), где Exception ex?

### **Практическая работа №4.**

1. Опишите недостатки и преимущества использования структур по сравнению с классами.

2. Какие изменения в использовании конструкторов структур произошли в C# 8? C# 10? C# 11?
3. Как осуществляется перегрузка операторов? Какие операции можно перегрузить?
4. В чем преимущество использования перечислений?
5. Что такое Nullable тип? Зачем он нужен?

#### **Практическая работа №5.**

1. Что такое интерфейс?
2. В чем отличие интерфейсов и абстрактных классов?
3. Новые возможности интерфейсов в C# 8.
4. Возможно ли множественное наследование интерфейсов?
5. Явная реализация интерфейсов.
6. Зачем нужны интерфейсы?

#### **Практическая работа №6.**

1. Что такое delegate?
2. Что такое функции обратного вызова (callback);
3. Как строятся цепочки делегатов?
4. Что такое анонимная функция? Приведите пример.
5. Напишите пример лямбда -выражения.
6. Связывание обработчика с событием.
7. Отключение обработчика. Динамическое связывание событий с их обработчиками.
8. Как реализуется блокировка потока
9. Что такое мьютекс и семафор.
10. В чем преимущества использования новой парадигмы многопоточности в C# с использованием библиотеки TPL.
11. Что такое Parallel.For
12. Что означают термины Async, Await.

#### **Практическая работа №7.**

1. Как во время работы программы получить путь к каталогу, откуда запущена программа?
2. Как получить актуальный разделитель текста Windows?
3. Какие методы и классы для работы с файлами вы знаете?
4. Основные типы паттернов проектирования.
5. На каких абстрактных типах строятся поведенческие паттерны?

#### **Практическая работа №8.**

1. Как можно создать "резиновый" массив?
2. Что такое индексаторы?
3. Зачем в коллекциях используются сразу 2 интерфейса IEnumerable и IEnumerator
4. В чем преимущества использования LINQ?

### **Образцы тестов**

Время на тест в минутах:	60
Всего вопросов:	100
Количество предъявляемых вопросов:	80

Вопрос 1	
Что означает термин void для метода в C#?	
Ответ 1	абстрактный метод
Ответ 2	метод не возвращает никакого значения
Ответ 3	перегружаемый метод

Ответ 4	скрытый метод
Ответ 5	Правильного ответа нет
Номер правильного ответа	2
Время на ответ в секундах	40
Балл	5

Вопрос 2	
С помощью какого служебного слова осуществляется перекрытие (переопределение) метода в классе - потомке?	
Ответ 1	base
Ответ 2	new
Ответ 3	override
Ответ 4	overload
Ответ 5	Правильного ответа нет
Номер правильного ответа	3
Время на ответ в секундах	40
Балл	5

Вопрос 3	
Какой модификатор нужно использовать для скрытия метода базового класса?	
Ответ 1	base
Ответ 2	virtual
Ответ 3	override
Ответ 4	new
Ответ 5	Правильного ответа нет
Номер правильного ответа	4
Время на ответ в секундах	40
Балл	5

Вопрос 4	
Для чего предназначены поля класса?	
Ответ 1	Предназначены для хранения данных во время работы экземпляра класса
Ответ 2	Предназначены для хранения методов
Ответ 3	Предназначены для хранения свойств во время работы экземпляра класса
Ответ 4	Предназначены для передачи данных из класса
Ответ 5	Правильного ответа нет
Номер правильного ответа	1
Время на ответ в секундах	40
Балл	5

Ответ 5	Правильного ответа нет
Номер правильного ответа	2
Время на ответ в секундах	40
Балл	5

Вопрос 98	
Какой код будет выполнен без ошибок?	
Ответ 1	int i = new int (); Console.WriteLine(i);
Ответ 2	int i; Console.WriteLine (i);
Ответ 3	int i = new int(10); Console.WriteLine(i);
Ответ 4	int i (10); Console.WriteLine(i);
Ответ 5	Правильного ответа нет
Номер правильного ответа	1
Время на ответ в секундах	40
Балл	5

Вопрос 99		
Укажите правильное утверждение:		
Ответ 1	Класс имеет только один конструктор;	
Ответ 2	Описание класса должно содержать хотя бы один конструктор;	
Ответ 3	Класс может иметь только один закрытый конструктор;	
Ответ 4	Если класс содержит только закрытый конструктор, то класс запечатан	
Ответ 5	Правильного ответа нет	
Номер правильного ответа		4
Время на ответ в секундах		40
Балл	5	

Вопрос 100		
Укажите неправильное утверждение: Тело класса может содержать описание:		
Ответ 1	полей;	
Ответ 2	методов;	
Ответ 3	классов	
Ответ 4	пространства имен;	
Ответ 5	Правильного ответа нет	
Номер правильного ответа		4
Время на ответ в секундах		40
Балл	5	

### Темы рефератов

№ п/п	Название темы
1.	Асинхронные методы в C#
2.	Метаданные и отражение
3.	Неуправляемый код
4.	Generic - типы
5.	Шаблоны проектирования на C#
6.	Сборка мусора (Garbage Collector)
7.	Исполняемые DLL и библиотеки ресурсов. Динамически загружаемые библиотеки. Создание собственных DLL.
8.	Программирование на C# для платформы Unity
9.	Защита программ, написанных на C#

## Критерии оценивания видов учебной работы

### 1. Критерии оценивания при проведении практических работ

Защита результатов работы происходит по результатам собеседования по выполненной работе и ответов на вопросы преподавателя и при наличии оформленного отчета о работе.

Работа отправляется на доработку, если студент:

- не может ясно сформулировать цели и задачи работы;
- или не владеет понятийным аппаратом, используемым в работе;
- или не может ответить на большинство вопросов из банка вопросов;
- содержатся грубые ошибки в программном коде;
- грубо нарушены правила оформления отчета.

При проверке задания, оценивается:

- ✓ понимание целей и задач работы;
- ✓ владение понятийным аппаратом, используемым в работе;
- ✓ ответы на вопросы преподавателя;
- ✓ понимание программного кода;
- ✓ грамотное оформление отчета.

Оценка «неудовлетворительно» выставляется студенту, если работа списана; при этом ему выставляется нулевой рейтинговый балл. В таком случае повторная сдача работы не допускается.

### 2. Критерии оценивания при проведении тестирования знаний

#### Критерии оценки:

- оценка «отлично» выставляется студенту, если студент верно ответил более, чем на 90% вопросов;
- оценка «хорошо» – ответил верно более, чем на 74% вопросов;
- оценка «удовлетворительно» – верно ответил более, чем на 60% вопросов;
- оценка «неудовлетворительно» – верно ответил менее, чем на 60% вопросов;

### 3. Критерии оценивания индивидуальных и творческих проектов

№ п/п	Показатель оценки	Кол-во баллов
1.	Правильный выбор технологий программирования в соответствии с поставленной задачей	30 баллов
2.	Грамотное использование принципов ООП	30 баллов
3.	Хорошая структурированность и "читабельность" кода	20 баллов
4.	Удобный и интуитивно понятный интерфейс программы	10 баллов
5.	Качественное комментирование и документирование приложения	10 баллов
<b>Итого:</b>		<b>100 баллов</b>

## Оценочные средства для промежуточной аттестации

### Экзаменационные вопросы

#### Теоретические вопросы:

1. Концепции языков программирования (автоматное, функциональное, логическое, процедурное, структурное, прототипное, контрактное и др).
2. Понятие классов, объектов (экземпляров классов), методов, полей и свойств в C#.
3. Синтаксис классов, создание и уничтожение объектов. Статические и виртуальные конструкторы.
4. Принципы ООП: инкапсуляция, наследование и полиморфизм. Сравнение принципов ООП в C++ и C#.
5. Модификаторы доступа к классам, структурам, перечислениям, интерфейсам, делегатам. Сравнение уровня доступа к классам и экземплярам, к экземплярам и методам, возвращающим значения.
6. Перекрытие (переопределение) методов. Перегрузка методов. Замещение метода предка.
7. Контроль и преобразование типов объектов. Явное и неявное преобразование типов. Защищенное преобразование, метод TryParse. Класс Convert. Операторы as, is.
8. Принципы создания простейших приложений .net. Windows Forms и WPF приложения. Структура обозревателя решений. Файлы сборки.
9. Структура панели элементов. Контейнеры для быстрой разработки приложений. Меню, компоненты, диалоги.
10. Основные методы и свойства для работы с буфером обмена.
11. Механизм обработки исключений в .net. Создание защищенных блоков в программе. Создание собственных исключений. Механизм контроля выхода за границу значимых типов.
12. Потоки и процессы. Создание дополнительных потоков. Запуск и завершение работы потока.
13. Каркас среды Framework .Net, CLR (Common Language Runtime) и FCL (Framework Class Library).
14. Обзор среды MSVS. Приложения и сборки. Пространство имен в .net.
15. Динамические массивы. Массивы массивов. Отличие массивов C# от массивов C++.
16. Статические типы и классы. Классы и структуры. Структуры и перечисления. Синтаксис структур. Сравнение структур и классов.
17. Инкапсуляция полей в C#. Методы свойств. Аксессоры (get, set). Автоматически реализуемые свойства.
18. Свойства и Индексаторы. Создание "умных массивов".
19. Виртуальные функции и позднее связывание. Переопределение методов, виртуальные методы.
20. Реализация полиморфизма в C#. Перегрузка методов.
21. Особенности реализации наследования в C# для классов и интерфейсов. Запечатанные классы. Множественное и одиночное наследование.
22. Абстрактные классы. Интерфейсы. Интерфейсные свойства.
23. Делегаты и события. Асинхронные вызовы. Анонимные методы. Лямбда выражения.
24. Обработчик события. Класс Sender. Связывание обработчика с событием. Динамическое связывание событий с их обработчиками.
25. Сборка «мусора» в .net. Опасный код.



26. Разработка пользовательского интерфейса в C#.
27. Основы разработки приложений WPF. Графика в C#
28. Статические, виртуальные, динамические и абстрактные методы.
29. Система типов в .net и C#. Перечислимые типы (Enum).
30. Создание главного и дополнительного меню. Панели инструментов и быстрых кнопок.
31. Методы с переменным количеством параметров.
32. Передача в метод и из метода переменных несылочного типа по ссылке. Упаковка и распаковка.
33. Классы для работы со строками в C#. Отличие от строк C++. Конструкторы класса String. Изменяемые и неизменяемые строки. Основные методы класса StringBuilder.
34. Коллекции. "Резиновые" массивы.

#### **Практические вопросы:**

1. Создать разветвленное действующее главное меню проекта и связать его с быстрыми кнопками доступа к меню.
2. Продемонстрировать на примере использование простейших компонентов управления, а также вывод сообщений.
3. Создать плавающий контейнер для быстрых кнопок. Связать реакции на нажатие кнопок с соответствующими пунктами меню.
4. Ограничить ввод данных в Edit только цифрами и разделителем Windows.
5. Написать фрагмент программы, которая позволяла бы открывать текстовый файл, редактировать его и сохранять на диск.
6. Реализовать на примере защищенные блоки.
7. Необходимо создать программу, реализующую основные графические примитивы.
8. Необходимо создать программу, реализующую динамическое создание визуальных компонент.
9. Объявить общий класс – предок для описания физического объекта, определить необходимые поля, методы и свойства.
10. Создать класс – потомок, перекрывающий отдельные методы предка. Реализовать перегрузку метода
11. Динамически создать несколько потоков в программе.
12. Реализовать наследование полей, методов и свойств класса – предка.

## Критерии оценивания при промежуточной аттестации

Разрабатываются согласно приказу ректора от 31.08.2023 № 964 «Об утверждении применения балльно-рейтинговой системы оценивания успеваемости обучающихся по основным профессиональным образовательным программам высшего образования в ФГАОУ ВО «Крымский федеральный университет им. В.И. Вернадского».

Итоговая семестровая оценка текущей работы студента вычисляется как суммарная оценка за выполненные практических работ (7 обязательных работ из 8 по списку по 4 балла максимум, восьмая работа факультативно), оценка компьютерного тестирования (до 10 баллов) и оценка посещений и конспектов лекций, от -8 до +8 баллов. Бонусы за работы, выполненные на профессиональном уровне до 4 баллов. Таким образом, максимальная оценка за текущую работу в семестре 50 баллов.

Максимальная оценка за экзамен - 50 баллов.

Студенты, выполнившие и отчитавшиеся менее чем по 4 работам из 8, к экзамену не допускаются и направляются на пересдачу экзамена.

На пересдаче экзамена студент обязан предоставить отчеты не менее чем по 4 лабораторным работам. На пересдаче экзамена используются следующие контрольно-измерительные мероприятия: устный ответ, написание простейшего программного кода, дополнительные вопросы, позволяющие выяснить понимание студентом принципов ООП, при необходимости может использоваться компьютерное тестирование.

### Шкала оценивания на экзамене:

Оценка	Знать	Уметь	Владеть
«Отлично»	основы функционирования среды .Net Framework, CLR, FCL, CLS, концепции языков программирования, принципы ООП и среду разработки Visual Studio, основы визуального программирования WinForm и WPF, основные понятия и термины: пространство имен, инкапсуляция, наследование и полиморфизм, типы-значения и ссылочные типы, классы, структуры, перечисления, поля, свойства, индексаторы, лямбда выражения, инкапсуляция, модификаторы доступа, конструкторы, перегрузка и переопределение методов, перегрузка операторов, интерфейсы, делегаты и события, преобразование типов, метаданные, принцип работы сборщика мусора, классы многопоточности программные конструкции в языках высокого уровня, позволяющие быстро конструировать эффективный код, средства отладки и тестирования кода, средства отладки и обработки исключительных ситуаций, стратегию инкапсуляции полей, сигнатуру событий, принципы разработки надежного кода	создавать программные проекты средней и высокой сложности с использованием основных принципов ООП, создавать Windows Forms и WPF программы с логически верным интерфейсом пользователя, создавать и управлять событиями с помощью делегатов, управлять инкапсуляцией полей, работать с абстрактными классами и виртуальными методами, динамически создавать компоненты, создавать классы, наследующие классы и интерфейсы, создавать графические приложения, создавать многопоточные приложения, управлять исключениями и защищенными блоками, обрабатывать события мыши и клавиатуры, использовать различные программные средства в зависимости от задачи, писать компактный хорошо читабельный код, самостоятельно получать знания, без помощи преподавателя находить и обрабатывать необходимую информацию, уметь использовать приобретенные знания и умения для принятия решений в нестандартных ситуациях, убедительно аргументировать свои мысли	основными приемами визуального программирования с использованием основных программных конструкций Си - подобных языков, владеть приемами получения и использования информации из электронных источников, включая MSDN, спецификации языков, навыками проектирования и реализации прикладного ПО, владение приемами оценки достоверности полученных результатов, практическими навыками создания эффективного программного кода с использованием современных технологий ООП программирования, владеть средствами отладки и тестирования кода, принципами разработки надежного ПО
«Хорошо»	основы функционирования среды .Net Framework, CLR, FCL, CLS,	создавать программные проекты средней сложности с	основными приемами визуального

	<p>концепции языков программирования, принципы ООП и среду разработки Visual Studio, основы визуального программирования WinForm и WPF, основные понятия и термины: пространство имен, инкапсуляция, наследование и полиморфизм, типы-значения и ссылочные типы, классы, структуры, перечисления, поля, свойства, индексаторы, лямбда выражения, инкапсуляция, модификаторы доступа, конструкторы, перегрузка и переопределение методов, перегрузка операторов, интерфейсы, делегаты и события, преобразование типов, метаданные, принцип работы сборщика мусора, классы многопоточности</p>	<p>использованием основных принципов ООП, создавать Windows Forms и WPF программы с логически верным интерфейсом пользователя, управлять инкапсуляцией полей, работать с абстрактными классами и виртуальными методами, создавать классы, наследующие классы и интерфейсы, создавать графические приложения, создавать многопоточные приложения, управлять исключениями и защищенными блоками, обрабатывать события мыши и клавиатуры, использовать различные программные средства в зависимости от задачи, писать компактный хорошо читабельный код, самостоятельно получать знания, без помощи преподавателя находить и обрабатывать необходимую информацию</p>	<p>программирования с использованием основных программных конструкций Си - подобных языков, владеть приемами получения и использования информации из электронных источников, включая MSDN, спецификации языков, навыками проектирования и реализации прикладного ПО, владение приемами оценки достоверности полученных результатов, практическими навыками создания эффективного программного кода с использованием современных технологий ООП программирования</p>
«Удовлетворительно»	<p>основы функционирования среды .Net Framework, CLR,FCL,CLS, концепции языков программирования, принципы ООП и среду разработки Visual Studio, основы визуального программирования WinForm и WPF, основные понятия и термины: пространство имен, инкапсуляция, наследование и полиморфизм, типы-значения и ссылочные типы, классы, структуры, перечисления, поля, свойства, инкапсуляция, модификаторы доступа, конструкторы, перегрузка и переопределение методов</p>	<p>Студент умеет создавать несложные программные проекты с использованием основных положений ООП. Использует возможности языка C# на 30-40%. Не умеет использовать современные программные конструкции. Умеет устанавливать простейшие приложения.</p>	<p>простыми приемами визуального программирования с использованием основных программных конструкций Си - подобных языков, владеть приемами получения и использования информации из электронных источников, включая MSDN, спецификации языков</p>
«Неудовлетворительно»	<p>знать часть понятий и терминов: пространство имен, инкапсуляция, наследование и полиморфизм, типы-значения и ссылочные типы, классы, структуры, перечисления, поля, свойства, инкапсуляция, модификаторы доступа, конструкторы, перегрузка и переопределение методов</p>	<p>Студент умеет программировать простейшие задачи, используя возможности языка C# на 5-10%.</p>	<p>простейшими приемами визуального программирования с использованием основных программных конструкций Си - подобных языков</p>