**Программа учебной дисциплины «Программирование»**

Утверждена

Академическим советом ООП

Протокол № от «01» июля 2015 г.

|  |  |
| --- | --- |
| Автор  | Подбельский В.В., д.т.н.,профессор, vpodbelskiy@hse.ru |
| Число кредитов  | 8 |
| Контактная работа (час.)  | 144 |
| Самостоятельная работа (час.)  | 160 |
| Курс  | первый |
| Формат изучения дисциплины | без использования онлайн курса |

1. **ЦЕЛЬ, РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ И ПРЕРЕКВИЗИТЫ**

Цели освоения дисциплины "Программирование":

* обеспечить студентов базовыми знаниями в области разработки программных продуктов;
* заложить основы для последующих курсов, посвященных созданию современных информационных систем;
* познакомить студентов с прогрессивными парадигмами программирования и механизмами их реализации в программных продуктах;
* обучить студентов применению современных интегрированных инструментальных сред, предназначенных для разработки программ в интерактивном режиме;
* привить студентам навыки исследовательской работы, предполагающей самостоятельное изучение специфических инструментов и средств, необходимых для решения именно той конкретной проблемы, которая в качестве задачи поставлена перед ними.

В результате освоения дисциплины студент должен:

** Знать:**

- основные парадигмы и методологии создания программных продуктов (процедурный, объектный, объектно-ориентированный и функциональный подходы, механизм обобщений и параллелизм);

- современный язык программирования (синтаксис и семантику языка C#, включая LINQ–технологию и возможности библиотек классов);

- особенности применения средств платформы .NET;

- возможности интегрированных сред разработки консольных приложений и программ с графическим интерфейсом.

** Уметь :**

- разрабатывать прикладные программы и библиотеки классов с помощью инструментальных интегрированных сред;

- отлаживать и тестировать создаваемые программные продукты, используя диагностические возможности среды разработки;

- применять библиотеку классов платформы .NET и свободно (открыто) распространяемые библиотеки;

- выполнять проектирование пользовательских интерфейсов консольных программ и программ с графическим интерфейсом;

- самостоятельно находить новые знания и решения, необходимые для реализации функциональных требований, сформулированных в техническом задании на программный продукт.

** Владеть:**

- навыками решения типовых задач программирования с применением современного языка программирования и передовых инструментальных средств;

- навыками проектирования и программирования консольных и оконных приложений и библиотек классов с использованием процедурного, объектного, объектно-ориентированного и функционального подходов;

- навыками применения средств платформы .NET и свободно (открыто) распространяемых библиотек..

Изучение дисциплины «Программирование» базируется на следующих дисциплинах:

- математика в объеме средней школы;

- информатика в объеме средней школы.

Для освоения учебной дисциплины студенты должны владеть следующими знаниями и компетенциями:

* основы информатики и алгоритмизации в рамках учебной программы средней школы;
* обладать навыками применения математического аппарата при выборе метода решения задачи.

Основные положения дисциплины должны быть использованы в дальнейшем при изучении следующих дисциплин:

1. Объектно-ориентированный анализ и программирование;
2. Построение и анализ алгоритмов;
3. Базы данных;
4. Распределенные вычисления;
5. Компьютерная графика.
6. **СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

**Введение. Императивная и декларативная парадигмы программирования. Эволюция языков программирования**

Задачи, цели, содержание и порядок прохождения дисциплины. Данные и их типы. Тип в математике и в программировании. Парадигмы программирования. Абстракция классов и абстракция функциональная, декларативный и императивный подходы. Алгоритм и программа. Структурный подход к разработке алгоритмов. Процедурно-ориентированное программирование. Языки высокого уровня, поддерживающие процедурно-ориентированный подход. Объектно-ориентированное программирование. Типы, классы, объекты. Языки высокого уровня, поддерживающие объектно-ориентированный подход. Типы приложений.

**Тема 1. Принципы программирования на основе платформы .Net Framework**

Основные компоненты платформы .Net Framework: общеязыковая исполняющая среда (CLR) и библиотека классов (FCL). Управляемый код. Общая система типов (CTS) и общеязыковая спецификация (CLS). Типы значений, ссылочные типы и базовые типы в CLR и в языке C#. Виды приложений среды .NET. Интегрированная среда для разработки приложений. Проекты и решения, их создание и применения. Средства навигации (проводник решений). Структура исходного кода консольного приложения. Возможности кодирования, трансляции, синтаксического анализа, отладки и исполнения простой программы.

**Тема 2. Основные элементы программ (данные, выражения, операторы)**

Определение класса. Пространство имен. Статические поля и методы. Типы значений и типы ссылок. Базовые типы: предельные значения, свойства и методы. Константы, переменные, именованные константы. Типы С# как классы платформы .NET Framework. Арифметические операции и выражения. Проблема преобразования типов. Явное преобразование типов. Присваивание простое и составное. Проблема преобразования типов при присваивании. Инициализация переменных. Целочисленные арифметические выражения. Переполнения при операциях с целыми. Особые ситуации в арифметических выражениях. Логические операции и операции отношения. Логические выражения. Условная (тернарная) операция. Операции сдвигов. Особенности выполнения операции сдвига применительно к знаковым и беззнаковым типам. Битовые операции. Применение битовых операций для проверки состояния битов и установки битов в заданное значение. Приоритеты операций. Управляющие структуры. Блок операторов. Условные операторы. Вложенность управляющих структур. Множественное ветвление (переключатели). Циклы и средства управления итерациями.

**Тема 3. Массивы и строки**

Одномерные массивы: объявление и инициализация. Доступ к элементу массива. Операции с массивом и элементами массива. Основные свойства и методы класса System.Array. Особенности операции присваивания применительно к ссылкам. Проблема потери ссылок. Сборка мусора. Многомерные массивы: объявление и инициализация. Доступ к элементу многомерного массива. Операции с массивом и элементами массива. Основные свойства и методы. Массивы ссылок на массивы. Объявление, создание и инициализация. Доступ к элементу. Операции с элементами массива ссылок на массивы. Массив с элементами типа **object**. Упаковка и распаковка значений. Динамическая идентификация типов.

Строка символов – объект типа **string**. Операции присваивания, сравнения и конкатенации строк. Преобразование строки в значение базового типа и преобразование значения базового типа в строку. Строки при вводе-выводе. Схема ввода данных с помощью средств библиотеки .NET Framework. Возможности библиотечного метода System.TryParse(). Контроль правильности вводимых данных. Объявление и инициализация строк. Доступ к элементу строки. Постоянство строк. Форматирование строки. Использование форматирования при выводе строк. Основные методы обработки строк: сравнение, поиск в строке, замена символов, вставка строк, удаление, разбиение строки на слова, формирование строки из слов, преобразование символов строки к заданному виду. Преобразование строки в массив символов и обратное преобразование. Массив строк. Параметры метода Main(). Запуск программы с передачей аргументов в метод Main().

**Тема 4. Методы как основа процедурного программирования**

Методы–процедуры и методы-функции. Соотношение фиксированных параметров и аргументов. Время жизни параметров. Локальные переменные и локальные константы. Параметры, передаваемые по значению и параметры, передаваемые по ссылке. Параметры с типами ссылок. Выходные параметры. Умалчиваемые значения параметров. Именованные аргументы. Модификатор **params** для методов с переменным числом аргументов. Перегрузка методов. Сигнатура метода при перегрузке. Рекурсивные методы. Стековые фреймы. Хвостовая рекурсия. Особенности применения метода Array.Sort().

**Тема 5. Класс как контейнер статических членов**

Определение собственных классов. Статические члены класса. Доступность статических членов (полей и методов) из других классов. Статические поля только для чтения. Поля классов (статические поля). Статические константы. Статические методы. Статический конструктор. Правила инициализации статических полей. Статические классы.

**Тема 6. Класс как тип. Объекты и их члены**

Объявление класса. Поля объектов. Объявления методов объектов. Инкапсуляция полей и методов. Средства определения доступности полей и методов из других классов. Поля только для чтения. Конструктор экземпляров класса. Конструктор умолчания. Конструктор с параметрами. Перегрузка конструкторов. Поверхностное и глубокое копирование объектов. Конструктор копирования. Свойства классов как средство доступа к закрытым полям. Определение свойств, доступных по чтению и по записи. Ссылка **this**. Деструкторы и финализаторы. Принцип инкапсуляции и методы объектов. Автореализуемые свойства. Индексаторы. Индексаторы, имитирующие наличие в объекте контейнера. Массивы объектов классов, определяемых программистом. Перегрузка операций. Декларации операций в классах и структурах. Синтаксис перегрузки унарных и бинарных операций. Перегрузка **true** и **false**.

**Тема 7. Исключения**

Классические способы обработки ошибок. Предварительная проверка параметров методов. Проверка кодов возвратов методов. О механизме исключений. Системные исключения и их обработка. Свойства исключений. Исключения при арифметических операциях. Перехват исключения. Блок обработки исключения. Использование нескольких блоков обработки исключения. Передача исключений во вложенных блоках. Передача исключений между методами. Управление программой с помощью исключений. Создание собственных исключений. Понятие объекта-исключения. Пользовательские классы исключений. Ретрансляция исключения.

**Тема 8. Регулярные выражения**

Регулярные выражения в командных языках и в языках программирования. Метасимволы и терминальные символы. Экранирование метасимволов. Подмножества метасимволов Директивы нулевой ширины. Квантификаторы жадные и ленивые. Группы регулярных выражений. Замены в тексте и деление текста на части. Особенности параметров метода Replace.

**Тема 9. Перечисления и структуры – типы значений**

Перечисления. Базовый класс перечислений. Структуры: определение типа, объявление переменных, операции над структурами. Отличие структур от классов. Упаковка и распаковка. Реализация структурами интерфейсов. Перегрузка операций в структурах.

**Тема 10. Визуальное проектирование приложений**

Формы, компоненты и элементов управления. Визуальное проектирование формы. Стандартные активные элементы графического пользовательского интерфейса: кнопки, флажки с зависимой фиксацией, флажки с независимой фиксацией, полосы прокрутки, панель индикации. Добавление элементов управления. Позиционирование элементов управления. Установка свойств формы и ее элементов. Связывание элементов формы со стандартными обработчиками событий. События от элементов формы, события клавиатуры и знако-координатных устройств (мышь). Программное изменение свойств элементов. Неотображаемые элементы (компоненты): таймер, всплывающие подсказки. Обработка событий от неотображаемых элементов.

Многооконный интерфейс. Меню. Создание и отображение формы. Передача данных в подчиненную форму. Прием данных из подчиненной формы. Диалоговые окна.

**Тема 11. Основы графики на платформе .Net Framework**

Организация графического вывода. Контекст отображения. Рисование на форме и в поле элемента управления. Идентификатор окна. Проблема перерисовки изображения и способы ее решения. Отслеживание состояния кнопок знако-координатного устройства. Отслеживание перемещения знако-координатного устройства. Рисование графических примитивов: линии, прямоугольники и многоугольники, эллипсы, сегменты, кривые, окрашенные фигуры. Растровые и векторные изображения. Буксировка, загрузка, рисование. Инструменты рисования: перья, кисти, шрифты. Вывод текста в графическом режиме.

**Тема 12. Отношения между классами**

Включение, вложение и наследование классов. Включение объектов классов. Отношение агрегации между классами. Отношение композиции. Вложение классов. Наследование классов. Доступность членов класса при наследовании. Методы при наследовании. Копирование полей при присваивании. Конструктор умолчания. Абстрактные методы и абстрактные классы. Опечатанные классы и методы. Применение абстрактных классов. Наследование полей и методов. Порядок вызова конструкторов при создании объекта класса наследника. Служебное слово **base**. Передача аргументов конструктору базового класса. Переопределение полей. Переопределение методов. Виртуальные методы. Полиморфизм и его виды.

**Тема 13. Делегаты, анонимные методы, лямбда-выражения, события и таймеры**

Синтаксис делегатов. Массивы делегатов. Многоадресные экземпляры делегатов. Делегаты и обратные вызовы. Анонимные методы. Лямбда-выражения. Понятие cобытия. Генерация и обработка событий. Типовая структура приложения, управляемого событиями.

**Тема 14. Интерфейсы**

Два вида наследования в ООП. Понятие интерфейса. Объявления интерфейсов. Реализация интерфейсов. Интерфейс как тип. Интерфейсы и наследование. Члены интерфейса: методы, свойства, события, индексаторы. Доступ к членам интерфейса. Реализация интерфейса в классе. Использование интерфейсов как альтернатива множественному наследованию.

**Тема 15. Обобщенное программирование**

Декларация обобщенных классов, интерфейсов и структур. Типизирующие параметры и список ограничений типизирующих параметров. Обобщенные методы и их сигнатуры. Виртуальные и абстрактные обобщенные методы. Вызовы обобщенных методов. Обобщенные коллекции. Обобщенные делегаты. Перегрузка операций в обобщенных классах. Использование обобщенных методов с делегатами.

**Тема 16. Коллекции и итераторы**

Коллекции и их отличие от массивов. Стандартные интерфейсы. Итераторы и оператор **foreach**. Динамический массив: объявление, доступ к элементам, методы добавления и удаления элементов. Стек: объявление, доступ к элементам, методы добавления и удаления элементов. Очередь: объявление, доступ к элементам, методы добавления и удаления элементов. Хэш-таблица: объявление, доступ к элементам, методы добавления и удаления элементов. Использование коллекций при решении прикладных задач.

**Тема 17. Потоковый ввод-вывод**

Понятие потока данных. Стандартные потоки: стандартный поток ввода, стандартный поток вывода, стандартный поток вывода сообщений об ошибках. Символьные потоки данных, связанные с файлами. Открытие и закрытие потока. Методы чтения и записи текстовых данных. Методы чтения и записи двоичных данных. Кодировка символов в потоках: кодовые страницы, кодировка UNICODE, кодировка в текстовых потоках, кодировка текстовых строк в двоичных потоках. Буферизация потоков: буферизация двоичных потоков, буферизация текстовых потоков, принудительный сброс буферов. Потоки, ориентированные на байты. Открытие и закрытие потока. Методы чтения и записи данных. Потоки в основной памяти: создание потока, методы чтения и записи данных. Сериализация. Сериализация на основе атрибутов. Открытие потока. Определение объекта форматирования. Двоичное форматирование и XML-форматирование (двоичная сериализация и XML-сериализация).

**Тема 18. Язык интегрированных запросов LINQ и элементы функционального программирования**

Потоковые последовательности данных. Операции (директивы) языка LINQ. LINQ-провайдеры. Методы, используемые в LINQ-запросах. Две формы записи LINQ-запроса. Переменные запросов и разделы LINQ-запросов. Отложенное выполнение LINQ-запросов. Анонимные типы в LINQ-запросах. Операции стандартных запросов. Делегаты как параметры методов, представляющих операции запросов. Предопределенные типы делегатов в .NET. Лямбда-выражения в качестве аргументов. Функции высших порядков.

**Тема 19. Асинхронные методы и параллельные программы**

Процессы операционной системы и потоки исполнения. Многопоточность. Пул потоков исполнения. Класс Thread и его члены. Создание потоков. Синхронизация потоков. Передача данных в поток и из потока. Обмены между потоками. Патерны асинхронного программирования: патерн опроса (polling), патерн ожидания, патерн ответного вызова. Механизм async/await. Возможности делегатов в параллельном программировании. Таймеры.

**Тема 20. Небезопасный код. Препроцессорные директивы**

Небезопасный код. Указатели и адресная арифметика. Препроцессорная обработка. Препроцессорные директивы.

1. **ОЦЕНИВАНИЕ**

Текущий контроль в каждом модуле предусматривает: самостоятельные работы на практических занятиях (на 10-15 минут), контрольную (письменную) работу и коллоквиум в виде теста на компьютере. Кроме того, во 2-м и 3-м модулях текущий контроль предусматривает контрольное домашнее задание. Промежуточный контроль: экзамен в конце 2-го модуля. Итоговый контроль: экзамен в конце 4-го модуля.

Контрольная письменная работа предусматривает программную реализацию на компьютере одной или нескольких задач. Выполняется на практических занятиях. На выполнение контрольной письменной работы в рамках практического занятия отводится 80 мин. учебного времени. За контрольную работу независимо от количества задач выставляется одна оценка по десятибалльной шкале.

Тесты коллоквиумов содержат вопросы по теоретическому материалу текущего и предыдущих модулей. За тест выставляется нормированная по сложности теста оценка по 10-ти балльной шкале.

Контрольное домашнее задание включает разработку, кодирование, тестирование и отладку программ для решения одной или нескольких задач с применением средств изучаемой темы. По контрольному домашнему заданию оформляется отчет в бумажном и в электронном виде. В установленный срок студент сдает полностью оформленный подписанный автором отчет и электронную копию программ, решающих задачи контрольного домашнего задания. За контрольное домашнее задание выставляется оценка по десятибалльной шкале.

Промежуточный контроль в форме экзамена предусматривает решение на компьютере задач по пройденным темам и тестирование на компьютере. В тест промежуточного экзамена входят вопросы по теоретическому и практическому материалу соответствующих модулей. Оценки по решению задач и тесту выставляются по 10-ти балльной шкале.

Итоговый экзамен предусматривает компьютерное решение задач по всем темам дисциплины и тестирование на компьютере. В тест итогового экзамена входят вопросы по теоретическому и практическому материалу всех 4-х модулей. Оценки по решению задач и тесту выставляются по 10-ти балльной шкале.

**Формы контроля знаний студентов**

|  |  |  |  |
| --- | --- | --- | --- |
| Тип контроля | Форма контроля | 1 год | Параметры |
| 1 | 2 | 3 | 4 |
| Текущий(неделя) | Контрольная работа | 5-я неделя | 5-я неделя  | 6-я неделя  | 6-я неделя  | Работа на компьютере, 80 минут |
| Текущие Контрольные мероприятия | недели:2,3,4 | недели:2,3,4 |  недели:2,3,4 | недели:2,3,4 |  Самостоятельнаяработа на компьютере, 10-15 минут |
| Контрольное домашнее задание |   |  6-я неделя  | 7-я неделя  |    |  Продолжительность КДЗ от 5 до 10 дней |
| Промежу­точный | Экзамен |  | \* |  |   | Реализация предложенного задания на компьютере, продолжительность 80 минут, экзаменационный компьютерный тест продолжительность 40 минут |
| Итоговый | Экзамен |  |  |  | \* | Реализация предложенного задания на компьютере, продолжительность 80 минут, экзаменационный компьютерный тест, продолжительность 40 минут |

**Оценочные средства для текущего контроля и аттестации студента**

**Тематика заданий текущего контроля**

**Тематика контрольных работ:**

* программирование ветвящихся и циклических алгоритмов;
* программирование алгоритмов обработки данных в виде массивов;
* использование методов, определяемых программистом-пользователем;
* обработка данных, организованных в виде массивов и строк;
* обработка исключений;
* объекты классов, определяемых программистом-пользователем;
* отношения между классами;
* событийное программирование;
* обработка массива объектов классов определяемых пользователем;
* классы, производные от библиотечного класса Form, и элементы управления;
* работа с потоками ввода-вывода, сериализация;
* основы компьютерной графики;
* работа с коллекциями, язык LINQ и элементы функционального программирования;
* асинхронные методы и параллельные программы.

**Тематика контрольных домашних заданий:**

* работа с массивами объектов пользовательских классов и потоками ввода-вывода;
* отношения между классами и обработка событий;
* применение средств библиотеки .NET для реализации параллельных алгоритмов;
* разработка Windows-приложения с развитым пользовательским интерфейсом и графическим выводом.

**Порядок формирования оценок по дисциплине**

По всем видам работ выставляется 10-балльная оценка.

**Десятибалльная оценка в конце 2-го модуля (1-го семестра):**

ИО = 0.2\*НО + 0.8\*ЭО; (ИО – итоговая оценка семестра).

ЭО – экзаменационная оценка;

Если ЭК<4, то ЭО = 0,8\*ЭК + 0,2\*ЭТ.

Если ЭT<4, то ЭО = 0,8\*ЭТ + 0,2\*ЭК.

В противном случае (когда ЭК>3 и ЭТ>3) ЭО = 0,5\*ЭК + 0,5\*ЭТ.

Где:

ЭК – экзаменационная контрольная работа;

ЭТ – экзаменационный тест;

НО – накопленная оценка (по семинарам) модулей семестра:

НО = 0,3\*ТП + 0,35\*КР +0,35\*КДЗ;

ТП – оценка текущих (контрольных) проверок в семестре:

ТП = 0.2\*ТП1 + 0.8\*ТП2

ТП1 – оценка проверок 1-го модуля;

ТП2 – оценка проверок 2-го модуля;

 ТПi = (сумма баллов по проверкам модуля)/(число проверок в модуле).

КР – оценка контрольных работ модулей:

КР = 0,2\*КР1 + 0,8\*КР2

КР1 – оценка контрольной 1-го модуля;

КР2 – оценка контрольной 2-го модуля;

КДЗ – контрольное домашнее задание

**Десятибалльная оценка в конце 4-го модуля (2-го семестра) рассчитывается аналогично, но используются накопленные оценки не 1-го и 2-го, а 3-го и 4-го модулей.**

**При пересдаче экзамена** (независимо от предыдущих оценок)

ИО = ЭО\*0.8;

Если ЭК<4, то ЭО = 0,8\*ЭК + 0,2\*ЭТ.

Если ЭT<4, то ЭО = 0,8\*ЭТ + 0,2\*ЭК.

В противном случае (когда ЭК>3 и ЭТ>3) ЭО = 0,5\*ЭК + 0,5\*ЭТ.

ИО – итоговая оценка;

ЭО – экзаменационная оценка пересдачи;

ЭК – экзаменационная контрольная пересдачи;

ЭТ – экзаменационный тест пересдачи.

Округление оценок, проставляемых в ведомости (НО, ЭО, ИО), выполняется до целых значений. Промежуточные вычисления производятся с точностью два знака после десятичного разделителя.

При выставлении итоговых оценок по 5-балльной шкале используется правило:

 Если Оит ≥ 8, то Оит5 = 5 (“отлично”);

 Если 6 ≤ Оит < 8, то Оит5 = 4 (“хорошо”);

 Если 4 ≤ Оит < 6, то Оит5 = 3 (“удовлетворительно”);

 Если Оит < 4, то Оит5 = 2 (“неудовлетворительно”).

**Методика формирования оценки за программную реализацию задачи**

Оценка за письменную работу (за работу по программированию на ЭВМ) формируется в соответствии со следующими ниже критериями

**Критерии выставления оценки за программную реализацию задачи**

При выполнении задачи на ЭВМ критерии делятся на две группы: основные и дополнительные. Основные критерии определяют нижний предел оценки по десятибалльной шкале в рамках соответствующей оценки по пятибалльной шкале. Дополнительные критерии определяют возможность повышения десятибалльной оценки

**1. Основные критерии**

**“ОТЛИЧНО”:**

8 баллов

1. Программа решает поставленную задачу и полностью соответствует спецификации.

2. Студент может обосновать принятые конструктивные решения.

3. Исходный текст документирован присутствуют сведения: назначение программы (условие задачи), номер учебной группы, фамилия и инициалы студента, дата выполнения, назначение используемых переменных, назначение и параметры определяемых программистом методов.

4. Программа остается работоспособной при вводе неверных исходных данных.

5. Предусмотрено повторное решение задачи без повторного запуска программы.

9 баллов

1. Программа соответствует критериям получения оценки 8 баллов.

2. Программа в целом соответствует дополнительным критериям.

10 баллов

1. Программа соответствует критериям получения оценки 8 баллов.

2. Программа полностью соответствует дополнительным критериям.

3. Студент может выполнить анализ альтернативных вариантов решения задачи.

**“ХОРОШО”:**

6 баллов

1. Программа решает поставленную задачу и соответствует спецификации. Отклонения от спецификации допущены при реализации второстепенных подзадач.

2. Студент может объяснить принятые конструктивные решения.

3. Исходный текст документирован.

7 баллов

1. Программа соответствует критериям получения оценки 6 баллов.

2. Программа в целом соответствует дополнительным критериям.

**“УДОВЛЕТВОРИТЕЛЬНО”:**

4 балла

1. Программа решает поставленную задачу, но имеет отклонения от спецификации.

2. Студент может объяснить функционирование программы по ее исходному тексту.

3. Исходный текст документирован.

5 баллов

1. Программа соответствует критериям получения оценки 4 балла.

2. Программа соответствует отдельным дополнительным критериям.

**“НЕУДОВЛЕТВОРИТЕЛЬНО”:**

1 балл:

1. Разработка программы не завершена.

2. Программа имеет синтаксические ошибки.

2 балла:

1. Программа не решает поставленную задачу или не соответствует спецификации.
2. В программе обнаруживаются не обработанные исключения.
3. Студент не может объяснить функционирование программы по ее исходному тексту.

3 балла:

1. Программа не решает поставленную задачу.

2. Программа завершается аварийно при некоторых вариантах исходных данных.

3. Студент может объяснить функционирование программы по ее исходному тексту.

**2. Дополнительные критерии**

1. Исходный текст программы структурирован.

2. Программа имеет средства изменения размерности и формы представления данных без изменения исходного текста или при минимальных изменениях (динамическое выделение памяти, именованные константы, и т.п.).

3. Программа реализована по модульному принципу и декомпозирована по реализуемым функциям.

4. В коде программы обоснованно использованы конструкции, изученные студентом самостоятельно, и их применение не противоречит основным требованиям, предъявляемым к решаемой задаче.

5. Предусмотрено само-документирование программы.

**Защита представленного студентом программного решения**

После проверки экзаменационной письменной работы преподавателем может быть принято решение о необходимости «защиты» выполненной работы.

Защита представленного студентом решения задачи предусматривает:

1. объяснение студентом всех использованных в коде конструкций (их назначение в программе, их синтаксис, их семантика…);
2. внесение в код программы предложенных преподавателем изменений и демонстрация работы измененного варианта программы;
3. рассмотрение возможных альтернативных вариантов решения задачи.

В ходе защиты студент должен продемонстрировать навыки программирования, знание теоретического материала курса, и владение терминологией дисциплины.

На основе приведенных выше критериев и, возможно, с учетом результатов защиты выставляется десятибалльная оценка за письменную работу (ЭК).

Если в задании предложено решить две задачи, то выставляется оценка по каждой из них (ЭК1и ЭК2), и общая оценка вычисляется как средневзвешенная:

ЭК = 0,5\*ЭК1 + 0,5\*ЭК2.

**Методика проведения и оценки экзаменационной работы**

Экзамен по дисциплине «информатика и программирование» предусматривает:

1. письменную работу (программирование на ЭВМ);
2. тест по теоретическому материалу дисциплины.

В письменной работе студенту необходимо решить на компьютере одну или две задачи по темам, проработанным в процессе изучения дисциплины. Так как программная реализация задач выполняется в системе программирования Visual Studio, в которую включена мощная справочная система, то никакими другими материалами и пособиями пользоваться при выполнении письменной работы нельзя.

1. **РЕСУРСЫ**
	1. **Основная литература**
* Подбельский В.В. Язык C#. Базовый курс. 2-е изд., перераб. и доп. – М.: Финансы и статистика, 2013. – 408 с.
* Шилдт Г. С# 4.0. Полное руководство. – М.: Издательский дом "Вильямс", 2013. – 1056 с.
* Скит Дж. C#: Программирование для профессионалов. 2-е изд., - М.: «Издательский дом "Вильямс"». 2011. -544 с.
* Албахари Дж., Албахари Б. C# 5.0. Полное описание языка. – М.: ООО "И. Д. Вильямс", 2013. – 1008 с.

**5.2. Дополнительная литература**

* Вирт Н. Алгоритмы + структуры данных = программы. – М.: Мир. 1986. – 406 с.
* Зиборов В.В. Visual C# 2012 на примерах. СПб.: БХВ-Петербург, 2013. – 480 с.

**5.3. Программное обеспечение**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование** | **Условия доступа** |
| 1. |  Microsoft Windows 7 Professional RUS | *Из внутренней сети университета (договор)* |
| 2. | Microsoft Visual Studio 2015 Community | *Из внутренней сети университета (договор)* |

**5.4. Профессиональные базы данных, информационные справочные системы,
интернет-ресурсы (электронные образовательные ресурсы)**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование** | **Условия доступа** |
|  | ***Интернет-ресурсы (электронные образовательные ресурсы)*** |
| 1. | MSDN – сеть разработчиков Microsoft | URL: https://msdn.microsoft.com/ru-ru/ |

* 1. **Материально-техническое обеспечение дисциплины**

Учебные аудитории для лекционных занятий по дисциплине обеспечивают использование и демонстрацию тематических иллюстраций, соответствующих программе дисциплины в составе:

ПЭВМ с доступом в Интернет (операционная система Microsoft Windows 7 Professional RUS и среда разработки Microsoft Visual Studio 2015 Community);

мультимедийный проектор с дистанционным управлением.

Учебные аудитории для лабораторных и самостоятельных занятий по дисциплине оснащены ПЭВМ (операционная система Microsoft Windows 7 Professional RUS и среда разработки Microsoft Visual Studio 2015 Community), с возможностью подключения к сети Интернет и доступом к электронной информационно-образовательной среде НИУ ВШЭ.