

**Федеральное государственное автономное образовательное учреждение
высшего образования
"Национальный исследовательский университет
"Высшая школа экономики"**

Факультет компьютерных наук
Департамент программной инженерии

**Рабочая программа дисциплины
Программирование**

для образовательной программы «Программная инженерия»
направления подготовки 09.03.04 «Программная инженерия»
уровень - бакалавр

Разработчик программы
Подбельский В.В., д.т.н., профессор, vpodbelskiy@hse.ru

Одобрена на заседании департамента программной инженерии «___»_____ 2017 г.
Руководитель департамента Авдошин С.М. _____

Утверждена Академическим советом образовательной программы
«___»_____ 2017 г., № протокола _____

Академический руководитель образовательной программы
Шилов В.В. _____

Москва, 2017

*Настоящая программа не может быть использована другими подразделениями
университета и другими вузами без разрешения подразделения-разработчика программы.*



1. Область применения и нормативные ссылки

Настоящая программа учебной дисциплины "Программирование" устанавливает минимальные требования к знаниям и умениям студента и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих данную дисциплину, учебных ассистентов и студентов образовательной программы «Программная инженерия» направления подготовки 09.03.04 «Программная инженерия», изучающих дисциплину «Программирование».

Программа разработана в соответствии с:

- образовательным стандартом Федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский университет «Высшая школа экономики»;
- образовательной программой «Программная инженерия» направления подготовки 09.03.04 «Программная инженерия»;
- рабочим учебным планом по направлению 09.03.04 «Программная инженерия», утвержденным в 2017 г.

2. Цели освоения дисциплины

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.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины

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–технологии и возможности библиотек классов);
 - особенности применения средств платформы .NET (Windows Forms, WPF, XAML);
 - возможности интегрированных сред разработки консольных приложений и программ с графическим интерфейсом.

• **Уметь :**

- разрабатывать прикладные программы (настольные-, мобильные- и Web-приложения) 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.

• **Иметь навыки (приобрести опыт):**

- в решении типовых задач программирования с применением современного языка программирования и передовых инструментальных средств;
- проектирования и программирования консольных и оконных приложений и библиотек классов с использованием процедурного, объектного, объектно-ориентированного, декларативного и функционального подходов;
- в применении средств платформы .NET и свободно (открыто) распространяемых библиотек..

В результате освоения дисциплины студент осваивает следующие компетенции:

Компетенция	Код по ФГОС/ НИУ
Способен учиться, приобретать новые знания, умения, в том числе в области, отличной от профессиональной	УК-1/СК-Б1
Способен работать с информацией: находить, оценивать и использовать информацию из различных источников, необходимую для решения научных и профессиональных задач (в том числе, на основе системного подхода)	УК-5/СК-Б5
Способен критически оценивать и переосмысливать накопленный опыт (собственный и чужой), рефлексировать профессиональную и социальную деятельность	УК-9/СК-Б10
Способен осуществлять производственную или прикладную деятельность в международной среде	УК-10/СК-Б11
Способен применять основные концепции, принципы, теории и факты, связанные с информатикой при решении научно-исследовательских задач;	ПК-1/ИК-1
Способен к формализации в своей предметной области с учетом ограничений используемых методов исследования	ПК-2/ИК-2
Способен использовать методы и инструментальные средства исследования объектов профессиональной деятельности	ПК-3/ИК-3
Способен готовить презентации, оформлять научно-технические отчеты по результатам выполненной работы, публиковать результаты исследований в виде статей и докладов на научно-технических конференциях	ПК-5/ИК-5
Способен создавать программное обеспечение для компьютерных систем различной архитектуры	ПК-9/ИК-9
Способен проектировать, конструировать и тестировать программные продукты	ПК-10/ИК-10

Способен читать, понимать и выделять главную идею прочитанного исходного кода, документации	ПК-11/ИК-11
Способен моделировать, анализировать и использовать формальные методы конструирования программного обеспечения	ПК-12/ИК-12
Способен создавать программные интерфейсы	ПК-14/ИК-14
Способен использовать различные технологии разработки программного обеспечения	ПК-16/ИК-16
Способен применять основные методы и инструменты разработки программного обеспечения	ПК-17/ИК-17
Способен понимать стандарты и модели жизненного цикла	ПК-19/ИК-19
Способен использовать методы контроля проекта и версий при создании программного обеспечения	ПК-25/ИК-25
Способен гибко адаптироваться к различным профессиональным ситуациям, проявлять творческий подход, инициативу и настойчивость в достижении целей профессиональной деятельности и личных	ПК-35/СЛК-Б8

4. Место дисциплины в структуре образовательной программы

Настоящая дисциплина является базовой и относится к профессиональному циклу дисциплин.

Изучение данной дисциплины базируется на знаниях студентами математики, основ информатики и алгоритмизации в рамках учебной программы средней школы, умении применять математический аппарат при выборе метода решения задачи.

Дисциплина является основой для последующего изучения дисциплин: Конструирование программного обеспечения, Алгоритмы и структуры данных, Базы данных, Распределенные вычисления, Компьютерная графика.

5. Тематический план учебной дисциплины

№	Название темы	Всего часов по дисциплине	Аудиторные часы		Самостоятельная работа
			Лекции	Семинары и практические занятия	
1	Введение. Императивная и декларативная парадигмы программирования. Эволюция языков программирования	2	1	0	1
2	Тема 1. Принципы программирования на основе платформы .Net Framework	2	1	0	1
3	Тема 2. Основные элементы программ (данные, выражения, операторы)	12	4	2	6
4	Тема 3. Массивы и строки	10	2	2	6
5	Тема 4. Методы как основа процедурного программирования	12	4	2	6

6	Тема 5. Класс как контейнер статических членов	12	4	2	6
7	Тема 6. Класс как тип. Объекты и их члены	10	4	2	8
8	Тема 7. Исключения	10	2	2	4
9	Тема 8. Регулярные выражения	10	2	0	6
10	Тема 9. Перечисления и структуры – типы значений	10	2	2	6
	Модуль 1	90	26	14	50
11	Тема 10. Визуальное проектирование приложений Windows Forms	8	4	4	8
12	Тема 11. Основы графики Windows Forms	8	4	4	8
13	Тема 12. Отношения между классами	14	6	6	12
14	Тема 13. Делегаты, анонимные методы, лямбда-выражения, события и таймеры	14	6	6	12
15	Тема 14. Интерфейсы	8	4	4	10
	Модуль 2	98	24	24	50
16	Тема 15. Обобщенное программирование	8	2	2	8
17	Тема 16. Коллекции и итераторы	8	4	4	8
18	Тема 17. Поточковый ввод-вывод	14	4	4	10
19	Тема 18. XAML и технология WPF.	52	14	14	24
	Модуль 3	98	24	24	50
20	Тема 19. Язык интегрированных запросов LINQ и элементы функционального программирования	24	8	8	15
21	Тема 20. Асинхронные методы и параллельные программы	24	6	6	15
22	Тема 21. Небезопасный код и препроцессорные директивы C#	16	4	4	10
23	Тема 22. Мобильные приложения и Web-приложения	16	4	4	10
	Модуль 4	80	22	22	50
	Итого по дисциплине	380	96	84	200

6. Формы контроля знаний студентов

Тип	Форма контроля	1 год	Параметры
-----	----------------	-------	-----------

контроля		1	2	3	4	
Текущий (неделя)	Контрольная работа	5-я неделя	5-я неделя		6-я неделя	Работа на компьютере, 70-80 минут
	Текущие контрольные мероприятия	недели : 2-7	недели : 1-8	недели : 2-9	недели : 1-9	Самостоятельная работа на компьютере, 10-15 минут
	КДЗ		6-я неделя	6-я неделя		Программа с оформлением документации
Промежуточный	Экзамен		*			Реализация предложенного задания на компьютере, продолжительность 70-80 минут, экзаменационный компьютерный тест, продолжительность 40-50 минут
Итоговый	Экзамен				*	Реализация предложенного задания на компьютере, продолжительность 70-80 минут, экзаменационный компьютерный тест продолжительность 40-50 минут

6.1. Критерии оценки знаний, навыков

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. 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 70-80 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.

Промежуточный контроль в форме экзамена предусматривает решение на компьютере задач по всем пройденным темам и тестирование на компьютере. В тест промежуточного экзамена входят вопросы по теоретическому и практическому материалу 1-го и 2-го модулей. Оценки по решению задач (в соответствии с методикой формирования оценки за программную реализацию задачи) 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 (в соответствии с методикой формирования оценки за программную реализацию задачи) и тесту выставляются по 10-ти балльной шкале.

7. Содержание 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.

Литература по теме: Вирт Н. стр. 14-73; Шилдт Г. стр. 31-40; Нейгел К. стр. 36-46.

Тема 1. Принципы программирования на основе платформы .Net Framework

Основные компоненты платформы .Net Framework: общезыковая исполняющая среда (CLR) и библиотека классов (FCL). Управляемый код. Общая система типов (CTS) и общезыковая 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. Кодирование, трансляция, синтаксический анализ, отладка и исполнение простой программы.

Литература по теме: Рихтер Дж. стр. 1-30; Подбельский В.В. стр. 5-18; Шилдт Г. стр. 45-56; Нейгел К. стр. 438-472.

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 C# 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.

Литература по теме: Подбельский В.В. стр. 19-96; Шилдт Г. стр. 67-145.

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().

Литература по теме: Подбельский В.В. стр. 97-144; Шилдт Г. стр. 177-208; Албахари Дж. стр. 302-309.

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().

Литература по теме: Подбельский В.В. стр. 145-173; Шилдт Г. стр. 218-260; Скит Дж. стр. 380-393.

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.

Литература по теме: Подбельский В.В. стр. 174-189; Шилдт Г. стр. 260-268; Скит Дж. стр. 206-208.

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**.

Литература по теме: Подбельский В.В. стр. 190-235; Шилдт Г. стр. 303-328.

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.

Литература по теме: Подбельский В.В. стр. 319-340; Шилдт Г. стр. 403-430; Нейгел К. стр. 415-435; Рихтер Дж. стр. 334-381.

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.

Литература по теме: Албахари Дж. стр. 981-999; Нейгел К. стр. 263-271; Нэш Т. стр. 253.

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.

Литература по теме: Подбельский В.В. стр. 292-318; Шилдт Г. стр. 391-402; Албахари Дж. стр. 127-129.

Тема 10. Визуальное проектирование приложений Windows Forms

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.

Литература по теме: Нейгел К. стр. 1053-1090; Фролов А.В. стр. 20-90, 143-265.

Тема 11. Основы графики Windows Forms

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.

Литература по теме: Нейгел К. стр. 1030-1183; Фролов А.В. стр. 288-349.

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.

Литература по теме: Подбельский В.В. стр. 236-266; Шилдт Г. стр. 329-374; Скит Дж. стр. 273-281.

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. Объявление события. Генерация и обработка событий. Типовая структура приложения, управляемого событиями.

Литература по теме: Подбельский В.В. стр. 341-369; Шилдт Г. стр. 473-510; Албахари Дж. стр. 145-164; Нейгел К. стр. 226-250, 347-348.

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.

Литература по теме: Подбельский В.В. стр. 267-291; Шилдт Г. стр. 375-391; Нейгел К. стр. 151-171; Рихтер Дж. стр. 278-289; Албахари Дж. стр. 104-111.

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.

Литература по теме: Подбельский В.В. стр. 370-396; Шилдт Г. стр. 575-635; Нейгел К. стр. 270-293.

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.

Литература по теме: Шилдт Г. стр. 923-1010; Албахари Дж. стр. 291-337; Нейгел К. стр. 294-337; Скит Дж. стр. 181-189.

Тема 17. Поточковый 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).

Литература по теме: Нейгел К. стр. 849-862; Шилдт Г. стр. 431-472; Албахари Дж. стр. 609-630.

Тема 18. XAML и Технология WPF

XAML (XML for Application Markup Language — XML для языка разметки приложений). Элементы и атрибуты в XAML. Свойства как атрибуты и как элементы. Зависимые и присоединенные свойства (dependency property). Расширения разметки (markup extensions). Иерархия классов WPF. Фигуры (shapes) — центральные элементы WPF. Элементы управления WPF. Компоновки элементов управления. Обработка событий в WPF. Стили, шаблоны и ресурсы. Триггеры свойств и событий. Привязка данных; команды; анимация; 3-D графика; интеграция WPF с Windows Forms.

Литература по теме: К. Watson, и др. 381-468 pp., Solis Daniel. Illustrated WPF, pp. ; Петцольд Ч. Microsoft Windows Presentation Foundation. Мак Дональд WPF

Тема 19. 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.

Литература по теме: Шилдт Г. стр. 637-680; Албахари Дж. стр. 337-387; Нейгел К. стр. 339-368; Скит Дж. стр. 296-335.

Тема 20. 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.

Литература по теме: Шилдт Г. стр. 885-922; ; Албахари Дж. стр. 553-608; 907-946. Девис А. стр. 13-90.

Тема 21. Небезопасный код и препроцессорные директивы C#

Небезопасный код. Указатели и адресная арифметика. Препроцессорная обработка. Препроцессорные директивы.
Литература по теме: Шилдт Г. стр. 528-535, 681-715; Албахари Дж. стр. 198-204.

Тема 22. Мобильные приложения и Web-приложения

Windows Store Apps.

Браузерное приложение WPF (XAML Browser Applications).

Литература по теме: К. Watson, и др. 469-512 pp., Solis Daniel. Illustrated WPF, pp. ; Петцольд Ч. Программирование для Microsoft Windows 8. 6-е изд.

8. Образовательные технологии

На каждом практическом занятии проводится разбор кодов конкретных программ, написанных на современном языке программирования. По существу, каждое занятие является мастер-классом по соответствующей теме дисциплины.

По пройденному материалу проводится контрольная проверка, результаты которой входят в накопленную оценку модуля.

Задания в тестовой форме применяются для обучения студентов и проведения промежуточных и итогового контролей.

9. Оценочные средства для текущего контроля и аттестации студента

9.1. 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 строк;
- обработка исключений;
- объекты классов, определяемых программистом-пользователем;
- отношения между классами;
- событийное программирование;
- обработка массива объектов классов определяемых пользователем;
- обобщенное программирование;
- интерфейсы;
- классы, производные от библиотечного класса Form, и элементы управления;
- работа с потоками ввода-вывода, сериализация;
- основы компьютерной графики Windows Forms;
- автономные XAML-файлы;
- WPF технология;
- работа с коллекциями, язык LINQ и элементы функционального программирования;
- асинхронные методы и параллельные программы.

9.2. Демонстрационные варианты заданий

9.2.1. Демонстрационный вариант задания для проведения текущего контроля.

В отдельном классе отдельного файла проекта реализовать метод, принимающий на вход двумерный массив и возвращающий одномерный массив длиной M, каждый элемент которого равен сумме элементов в соответствующем столбце исходного массива. В основной программе запросить у пользователя размеры двумерного массива (N и M), а затем создать двумерный массив целых

чисел размером $N \times M$ и заполнить его случайными целыми числами из диапазона $[-5; 5]$. Вывести на экран исходный массив и получившийся после применения метода одномерный массив (элементы в строках при выводе разделять тремя пробелами). Предусмотреть контроль входных данных и реализовать повторение решения.

9.2.2. Демонстрационный вариант контрольной работы.

В библиотеке определить класс Triangle - «треугольник на плоскости» с длинами сторон A, B, C и свойствами для периметра P и площади S треугольника. Дополнительные члены класса - по вашему выбору.

В первой программе создать в каталоге вашего решения текстовый файл "Triangles.txt", разместив на каждой из N строк файла значения длин сторон одного треугольника. Значения сторон выбираются случайно из интервала от 1 до M (включительно) и разделены пробелами. Число строк в файле N и значение M вводятся пользователем.

Во второй программе вашего решения определить класс Register, объект которого представляет созданный первой программой текстовый файл как нумерованную последовательность объектов класса Triangle.

В методе Main() второй программы, применяя оператор **foreach** к объекту класса Register, вывести из последовательности информацию об объектах класса Triangle. Затем, используя LINQ-запрос, получить последовательность, в которой элементы упорядочены по периметрам треугольников. Вывести элементы полученной последовательности. Используя второй LINQ-запрос, получить последовательность треугольников, для которых отношение площади к периметру больше значения, заданного пользователем. Вывести элементы полученной последовательности.

9.2.3. Демонстрационный вариант тестового задания.

Задание с открытым ответом

В результате выполнения следующей программы:

```
using System;
class Program {
    delegate char[] Del(int n);
    static void Main() {
        Del del_1 = new Del(convert);
        char[] temp = del_1(567);
        Array.Sort(temp, comp);
        foreach (char ch in temp) Console.Write(ch);
    }
    static char[] convert(int num) {
        string str = num.ToString();
        return str.ToCharArray();
    }
    static int comp(char d, char e) {
        if (d == e) return 0;
        if (d % 2 == 0 && e % 2 != 0) return 1;
        return -1;
    }
}
```

на экран будет выведено: _____.

Задание с выбором нескольких верных ответов

Верно, что класс **Graphics**

- 1) является абстрактным;
- 2) не может быть унаследован;
- 3) располагается в пространстве имен **System.Drawing**
- 4) предоставляет прототипы методов для рисования объектов;
- 5) предоставляет методы для рисования объектов.

10. Порядок формирования оценок по дисциплине

По всем видам работ выставляется 10-балльная оценка.

Итоговая десятибалльная оценка семестра в конце 2-го модуля (1-го семестра):

$$ИО = 0,8*ЭО + 0,2*НО.$$

ЭО – экзаменационная оценка:

$$ЭО = 0,8*\min(ЭК; ЭТ) + 0,2*\max(ЭК; ЭТ).$$

Где:

ЭК – экзаменационная контрольная работа;

ЭТ – экзаменационный тест;

НО – накопленная оценка двух модулей семестра:

$$НО = 0,2*НО1 + 0,8*НО2,$$

где

НО1 = 0,2*ТП1 + 0,8*КР1 - накопленная оценка 1-го модуля

НО2 = 0,3*ТП2 + 0,35*КДЗ + 0,35*КР2 - накопленная оценка 2-го модуля

ТП1 – оценка текущих проверок 1-го модуля;

ТП2 – оценка текущих проверок 2-го модуля;

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

КР1 – оценка контрольной 1-го модуля;

КР2 – оценка контрольной 2-го модуля.

КДЗ – оценка контрольного домашнего задания.

НО3 = 0,5*ТП3 + 0,5*КДЗ - накопленная оценка 3-го модуля

НО4 = 0,2*ТП4 + 0,8*КР4 - накопленная оценка 4-го модуля

Итоговая 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 передаче экзамена

$$ИО = 0,8*ЭО + 0,2*НО$$

ИО – итоговая оценка;

ЭО – экзаменационная оценка передачи;

НО – накопленная оценка двух модулей семестра

$$ЭО = 0,8*\min(ЭК; ЭТ) + 0,2*\max(ЭК; ЭТ).$$

ЭК – экзаменационная контрольная передачи;

ЭТ – экзаменационный тест передачи.

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

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

Если $ИО \geq 8$, то $ИО = 5$ (“отлично”);

Если $6 \leq Оит < 8$, то $ИО = 4$ (“хорошо”);

Если $4 \leq Оит < 6$, то $ИО = 3$ (“удовлетворительно”);

Если $ИО < 4$, то $ИО = 2$ (“неудовлетворительно”).

Итоговая оценка за второй семестр выставляется в диплом.

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 в комментариях обосновал принятые конструктивные решения.
3. Исходный текст документирован. Присутствуют сведения о назначении используемых переменных, параметров, методов, классов, объектов.
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 соответствует некоторым дополнительным критериям.

10 баллов

1. Программа соответствует критериям получения оценки 8 баллов.
2. Программа полностью соответствует дополнительным критериям.
3. Студент отразил в комментариях возможность альтернативных вариантов решения задачи.

“ХОРОШО”:

6 баллов

1. Программа решает поставленную задачу и соответствует спецификации. Отклонения от спецификации допущены при реализации второстепенных подзадач.
2. 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. 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 (не компилируется).

2 балла:

1. Программа не решает поставленную задачу или не соответствует спецификации.

2. В программе обнаруживаются не обработанные исключения.
3. Программа завершается аварийно хотя бы при некоторых вариантах исходных данных.

3 балла:

1. Программа не решает поставленную задачу при некоторых вариантах исходных данных.

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. обсуждение 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 (ЭК1 и ЭК2), и общая оценка вычисляется как:

$$ЭК = 0,8 * \min(ЭК1; ЭК2) + 0,2 * \max(ЭК1; ЭК2).$$

Методика проведения и оценки экзаменационной работы

Экзамен по дисциплине 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. Программная реализация задач выполняется в системе программирования Visual Studio. Никакими материалами и пособиями пользоваться при выполнении письменной работы нельзя.

11. Учебно-методическое и информационное обеспечение дисциплины

11.1. Базовые учебники:

- Подбельский В.В. Язык С#. 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.
- Подбельский В.В. Язык С#. Решение задач. – М.: Финансы и статистика, 2014. – 296 с.
- Solis Daniel. Illustrated C# 2010. –USA: - Apress. – 712 pp.
- Solis Daniel. Illustrated WPF. –USA: - Apress. – 530 pp.

11.2. Основная литература:

- Скит Дж. С#: Программирование для профессионалов. 2-е изд., - М.: «Издательский дом "Вильямс"». 2011. -544 с.
- Фролов А.В., Фролов Г.В. Визуальное проектирование приложений С#. – М.: КУДИЦ-ОБРАЗ, 2003. – 512 с.
- Нэш Т. С# 2008. Ускоренный курс для профессионалов. – М.: ООО Издательский дом "Вильямс", 2008. – 576 с.
- Албахари Дж., Албахари Б. С# 6.0. Справочник. Полное описание языка. – М.: ООО "И. Д. Вильямс", 2016. – 1040 с.
- К. Watson, J.V. Hammer, J.D. Reid, M. Skinner, D. Kemper, C. Nagel, *Beginning Visual C# 2012. Programming.* 2012. - 910 pp.
- Farrell J., *Microsoft Visual C# 2010: An introduction to object-oriented programming.* 4-th edition, 2011. - 826 pp.
- Петцольд Ч. Microsoft Windows Presentation Foundation. М.: - Русская редакция; СПб.: Питер, 2008. – 944 с.
- Петцольд Ч. Программирование для Microsoft Windows 8. 6-е изд. – СПб.: Питер, 2014. – 1008 с.
- Мак-Дональд М. WPF: Windows Presentation Foundation в .NET 4.5 с примерами на С# 5.0 для профессионалов. 4-е изд. - М.: Вильямс, 2013. – 1024 с.
- Девис Алекс. Асинхронное программирование в С#. М.: ДМК Пресс. 2015. -120 с.

11.3 Дополнительная литература и источники

- Вирт Н. Алгоритмы + структуры данных = программы. – М.: Мир. 1986. – 406 с.
- С#. Language Specification. Version 4.0. : Microsoft Corporation. . 2010. – 525 pp.
- Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке С#. Мастер класс. 2-е изд. исправ. . – М. : Русская Редакция; СПб.: Питер, 2008. – 656 с.
- Нейгел К. и др. С# 4.0 и платформа .NET 4.0 для профессионалов. – М. : ООО "И.Д. Вильямс", 2011. – 1440 с.
- Петцольд Ч. Программирование для Microsoft Windows на С#. В 2-х томах. – М.: Издательско-торговый дом «Русская Редакция». 2002. Том 1 – 624 с. Том 2 – 576 с.
- Зиборов В.В. Visual C# 2012 на примерах. СПб.: БХВ-Петербург, 2013. – 480 с.
- Antonopoulos Nikos, Gillam Lee. *Cloud Computing: Principles, Systems and Applications.* – L.: Springer, 2012. – 382 p. ISBN 97811447125808.
- Троелсен, Э. Язык программирования С# 5.0 и платформа .NET 4.5, 6-е изд.: Пер. с англ. – М.: ООО "И.Д. Вильямс", 2013. – 1312 с

11.4. Справочники, словари, энциклопедии

MSDN.

11.5. Программные средства

Для успешного освоения дисциплины, студент использует следующие программные средства:

Microsoft Visual Studio 2015

Visual C# 2015 Express Edition

11.6. Дистанционная поддержка дисциплины

Для ознакомления с оценками контроля и методическими материалами по дисциплине предусмотрен раздел в LMS.

Облачные вычисления в режиме PaaS (Platform as a Service).

12. Материально-техническое обеспечение дисциплины

Проектор для лекций и семинаров, классы для семинаров с компьютерами, на которых установлена интегрированная среда Microsoft Visual Studio 2015