

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ
Декан факультета _____ С.М. Пасмурнов
« » 2017 г.

РАБОЧАЯ ПРОГРАММА
дисциплины
«Практикум по программированию»

Направление подготовки 09.03.01 ИНФОРМАТИКА И
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Профиль Системы автоматизированного проектирования

Квалификация выпускника бакалавр

Нормативный период обучения 4 года

Форма обучения очная

Год начала подготовки 2017

Автор программы _____ /Иванов Д.В./

Заведующий кафедрой
Систем
автоматизированного
проектирования и
информационных систем _____ /Львович Я.Е./

Руководитель ОПОП _____ /Яскевич О.Г. /

Воронеж 2017

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

обеспечение знаний о современных технологиях построения программного обеспечения; изучение дисциплины должно способствовать пониманию основных современных сведений о принципах, методах, этапах проектирования ПО.

1.2. Задачи освоения дисциплины

- получение основных сведений о современных технологиях программирования;
- умение применять современные возможности средств программирования;
- приобретение навыков программирования основных конструкций языков программирования.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Практикум по программированию» относится к дисциплинам вариативной части блока ФТД.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Практикум по программированию» направлен на формирование следующих компетенций:

ПВК-1 - способностью разрабатывать компоненты программных комплексов и информационных систем, используя технологии программирования и инструментальные средства разработки

ПВК-5 - способностью проектировать и разрабатывать компоненты программного обеспечения для автоматизированных систем

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПВК-1	знать основные этапы и технологии программирования;
	уметь осуществлять сборку проекта программного средства;
	владеть навыками программирования основных конструкций языка;
ПВК-5	знать технические основы программирования;
	уметь применять возможности средств программирования;
	владеть навыками программирования базовых компонентов программного обеспечения.

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Практикум по программированию» составляет 2 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры
		2
Аудиторные занятия (всего)	36	36
В том числе:		
Лабораторные работы (ЛР)	36	36
Самостоятельная работа	36	36
Виды промежуточной аттестации - зачет	+	+
Общая трудоемкость:		
академические часы	72	72
зач.ед.	2	2

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лаб. зан.	СРС	Всего, час
1	Программирование основных конструкций языка	Компиляторы, трансляторы, интерпретаторы. Машинный код. Консольный ввод-вывод. Основные сведения о членах и типах. Константы и поля. Методы. Параметры. Свойства. Символы,	18	18	36
2	Программирование, сборка и реализация ПО	Массивы. Строки и обработка текста. Компоновка, упаковка, развертывание приложений и типов. Совместно используемые сборки и сборки с о строгим именем.	18	18	36
Итого			36	36	72

5.2 Перечень лабораторных работ

1. Программирование математических и логических конструкций в консольном режиме.
2. Программирование конструкций условного оператора и оператора выбора.
3. Программирование конструкций циклов.
4. Программирование классов, перегрузка методов.
5. Программирование свойств, перегрузка конструкторов.
6. Программирование массивов и строк.
7. Программирование абстрактных классов, полиморфизм.
8. Программирование классов для работы с файловой системой.

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины не предусматривает выполнение курсового проекта (работы) или контрольной работы.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ПВК-1	знать основные этапы и технологии программирования;	Знание теоретического материала для лабораторных работ № 1-4	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь осуществлять сборку проекта программного средства;	Выполнение лабораторных работ № 1-4	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть навыками программирования основных конструкций языка;	Выполнение лабораторных работ № 1-4	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ПВК-5	знать технические основы программирования;	Знание теоретического материала для лабораторных работ № 5-8	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь применять возможности средств программирования;	Выполнение лабораторных работ № 5-8	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть навыками программирования базовых компонентов программного обеспечения.	Выполнение лабораторных работ № 5-8	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 2 семестре для очной формы обучения по двухбалльной системе:

«зачтено»

«не зачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Не зачтено
ПВК-1	знать основные этапы и технологии программирования;	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь осуществлять сборку проекта программного средства;	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

	владеть навыками программирования основных конструкций языка;	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ПВК-5	знать технические основы программирования;	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь применять возможности средств программирования;	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть навыками программирования базовых компонентов программного обеспечения.	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

Вопрос 1: Что такое ENUM?

- (A) Используется для инициализации переменных
- (B) Используется для определения констант
- (C) Он используется для определения переменных
- (D) Нет верного ответа

Вопрос 2: Что делает метод Dispose?

- (A) Закрывает соединение
- (B) Временно избавиться от соединения
- (C) Удаляет его из памяти
- (D) Все вышеперечисленное

Вопрос 3: Особенность локальной переменной

- (A) Она должна быть объявлена в методе
- (B) представляет объект класса
- (C) Может использоваться в любом месте программы
- (D) Она должна использоваться в классе

Вопрос 4: В чем разница между Convert.ToInt32 и Int.Parse?

- (A) Оба одинаковы
- (B) Convert.ToInt32 Не может обработать нулевые значения, он выдаст ошибку ArgumentException.
- (C) Int. Анализ не может обработать нулевые значения, он выдает ошибку ArgumentException.
- (D) Оба могут обрабатывать нулевые значения

Вопрос 5: C# не поддерживает:

- (A) абстракция

- (Б) полиморфизм
- (С) множественное наследование
- (D) наследство

Вопрос 6: Какое из следующих утверждений верно в отношении управляемого кода?

- (A) Управляемый код, который выполняется поверх Windows.
- (B) Управляемый код CLR.
- (C) Управляемый код - это то место, где код является сборщиком мусора.
- (D) Управляемый код для компиляторов.

Вопрос 7: Можем ли мы переопределить частные виртуальные методы?

- (A) да
- (B) Нет

Вопрос 8: Какие из следующих сборок можно сохранить в Global Assembly Cache?

- (A) private
- (B) shared
- (C) protected
- (D) friend

Вопрос 9: Сборка - это

- (A) декларативный синтаксис Java
- (B) свойства C#
- (C) события C#
- (D) Коллекция файлов, которые представляются программисту в виде одной DLL или EXE.

Вопрос 10: Какие из следующих утверждений правильно определяют .NET Framework?

- (A) Это среда для разработки, создания, развертывания и выполнения настольных приложений, веб-приложений и веб-служб.
- (B) Это среда для разработки, создания, развертывания и выполнения распределенных приложений.
- (C) Это среда для разработки и исполнения приложений Windows.
- (D) Это среда для разработки, создания, развертывания и выполнения только веб-приложений.

7.2.2 Примерный перечень заданий для решения стандартных задач

Вопрос 1: Какое из следующих утверждений ИСТИННО в отношении .NET CLR?

1. Он обеспечивает нейтральную для языка среду разработки и исполнения.
2. Это гарантирует, что он не авторизован для доступа.

3. Предоставляет сервисы для запуска «управляемых» приложений.
 4. Ресурсы мусора.
 5. Предоставляет сервисы для запуска «неуправляемых» приложений.
- (A) только 1 и 2
(B) только 1, 2 и 4
(C) 1, 2, 3, 4
(D) только 4 и 5

Вопросы 2: Что можно использовать в считывателе данных до чтения?

- (A) getvalue
(B) getstring
(C) getnumber
(D) None

Вопрос 3: Может ли интерфейс быть создан непосредственно?

- (A) да
(B) нет

Вопрос 4: Какие из перечисленных ниже являются действительными счетчиками производительности .NET CLR JIT?

1. Общая память, используемая для компиляции JIT
 2. Средняя память, используемая для компиляции JIT
 3. Количество методов, которые не удалось скомпилировать со стандартным JIT
 4. Процент JIT-компиляции процессорного времени
 5. Процент памяти, выделенной для компиляции JIT
- (A) 1, 2
(B) 1, 5
(C) 3, 4
(D) 4, 5

Вопрос 5: Автоматическая страничная подкачка возможна в

- (A) datareader
(B) dataset
(C) datatable
(D) все варианты

Вопрос 6: Какой класс унаследован от класса?

- (A) Sealed
(B) Gather
(C) Static
(D) Constru

Вопрос 7: Особенности переменных только для чтения

- (A) Декларация и инициализация разделены

- (B) Распределяется во время компиляции
- (C) Распределяется во время выполнения
- (D) все вышеперечисленное

Вопрос 8: Событие имеет _____ в качестве типа возврата по умолчанию

- (A) Нет типа возврата для событий
- (B) String
- (C) Double
- (D) Integer

Вопрос 9: Код, предназначенный для общезыковой среды выполнения, известен как

- (A) Наследие
- (B) Управляемый код
- (C) Неуправляемый
- (D) родной код

Вопрос 10: Какой метод вы вызываете для объекта SqlDataAdapter для загрузки вашего набора данных с данными?

- (A) Обновить
- (B) Заполнить
- (C) Собрать
- (D) Загрузить

7.2.3 Примерный перечень заданий для решения прикладных задач

Вопрос 1: Могут ли частные переменные уровня класса наследоваться?

- (A) Да, и мы можем получить к ним доступ
- (B) Нет, и мы не можем получить к ним доступ
- (C) Да, но мы не можем получить к ним доступ
- (D) Все вышеперечисленное неверно

Вопросы 2: Есть ли ошибки в этом конструкторе EmployeeMgmt: `Public int EmployeeMgmt {emp_id = 100; }`

- (A) Тип возврата
- (B) Формальные параметры
- (C) ошибок нет
- (D) Имя

Вопрос 3: Что верно в отношении интерфейса и абстрактных методов?

- (A) Мы можем написать только один абстрактный метод внутри интерфейса.
- (B) Ни один метод не является абстрактным внутри интерфейса
- (C) В абстрактном методе.
- (D) Ничего из вышеперечисленного

Вопрос 4: Какой объект вы можете использовать?

- (A) Объект сеанса
- (B) Объект сервера
- (C) Объект ответа
- (D) Объект приложения

Вопрос 5: Два метода с одинаковым именем, но с разными параметрами.

- (A) перегрузка
- (B) мультиплексирование
- (C) дуплекс
- (D) Загрузка

Вопрос 6: В чем разница между методом `Convert.ToString (str)` и `str.ToString ()`?

- (A) Функция `Convert.ToString (str)` обрабатывает `NULL`, а функция `str.ToString ()` - нет. Будет выброшено исключение `NULL`.
- (B) функция `str.ToString ()` обрабатывает `NULL`, а `Convert.ToString (str)` - нет. Будет выброшено исключение `NULL`.
- (C) Оба могут обрабатывать `NULL`
- (D) Никто не может обращаться с `NULL`

Вопрос 7: Какое из этих определений строк предотвратит обратную косую черту в C#?

- (A) строка `s = #.n Тестовая строка .;`
- (B) строка `s = @.n Тестовая строка .;`
- (C) строка `s = ..n тестовая строка .;`
- (D) строка `s = .n тестовая строка .;`

Вопрос 8: Разница между `Convert.ToString ()` и `ToString ()`

- (A) `Convert.ToString ()` обрабатывает нулевые значения, но `ToString ()` нет
- (B) `Convert.ToString ()` обрабатывает только нулевые значения
- (C) `ToString ()` обрабатывает нулевые значения, но `Convert.ToString ()` нет
- (D) Вывод `ToString ()` в соответствии с предоставленным форматом

Вопрос 9: Как мы можем отсортировать элементы массива в порядке убывания?

- (A) вызывая метод `Sortdescending()`
- (B) вызывая метод `Sort()`
- (C) Вызывая методы `Sort()` и затем `Reverse()`
- (D) вызывая метод `Sortreverse()`

Вопрос 10: Что из следующего составляет .NET Framework?

- 1. Приложения ASP.NET
- 2. CLR

- 3. Framework Class Library
- 4. Приложения WinForm
- 5. Службы Windows

- (A) 2, 5
- (Б) 2, 1
- (C) 2, 3
- (D) 3, 4

7.2.4 Примерный перечень вопросов для подготовки к зачету

1. Алгоритмы с ветвлением. Пример алгоритма.
2. Алгоритм цикла с предусловием. Пример алгоритма.
3. Алгоритм цикла с постусловием. Пример алгоритма.
4. Алгоритм цикла с управляющей переменной. Пример алгоритма.
5. Целый и вещественный типы данных. Операции с переменными этого типа.
6. Логический тип данных. Символьный тип данных. Операции с переменными этого типа.
7. Общие принципы разработки ПО.
8. Организация ввода/вывода в консольном приложении.
9. Основные типы данных языков программирования.
10. Структура приложения на языке C#.
11. Организация ввода/вывода на языке C#.
12. Операторы выбора и условного перехода в языке C#.
13. Операторы цикла в языке C#.
14. Преобразование типов на языке C#.
15. Работа с массивами на языке C#.
16. Работа с строками на языке C#.
17. Работа с объектами на языке C#.
18. Работа с файлами на языке C#.
19. Общие принципы разработки ПО.
20. Основные понятия объектно-ориентированного программирования.
21. Определение классов: основные понятия, работа с визуальными компонентами.
22. Реализация инкапсуляции в визуальном приложении.
23. Реализация наследования в визуальном приложении.
24. Реализация полиморфизма в визуальном приложении.
25. Организация областей видимости в визуальном приложении.
26. Работа с файловой системой.
27. Работа с графическими изображениями.
28. Типы приложений развертывания Windows-приложений.
29. Классы файлового ввода информации.
30. Классы файлового вывода информации.

7.2.5 Примерный перечень заданий для решения прикладных задач

Не предусмотрено учебным планом

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Зачет проводится по тест-билетам, каждый из которых содержит 10 вопросов и задачу. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом, задача оценивается в 10 баллов (5 баллов верное решение и 5 баллов за верный ответ). Максимальное количество набранных баллов – 20.

1. Оценка «Неудовлетворительно» ставится в случае, если студент набрал менее 6 баллов.

2. Оценка «Удовлетворительно» ставится в случае, если студент набрал от 6 до 10 баллов

3. Оценка «Хорошо» ставится в случае, если студент набрал от 11 до 15 баллов.

4. Оценка «Отлично» ставится, если студент набрал от 16 до 20 баллов.

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Программирование основных конструкций языка	ПВК-1	Защита лабораторных работ № 1-4
2	Программирование, сборка и реализация ПО	ПВК-5	Тест, лабораторных работ № 5-8

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

1. Иванов Д.В., Тишуков Б.Н. Программирование на языках высокого уровня [Электронный ресурс] / Учебно-методическое пособие. - Электрон. текстовые, граф. дан. (1,03 Мб). - Воронеж : ФГБОУ ВО "Воронежский государственный технический университет", 2016.

2. Снетков В.М. Практикум прикладного программирования на C# в

среде VS.NET 2008 [Электронный ресурс]/ Снетков В.М.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 1691 с.— Режим доступа: <http://www.iprbookshop.ru/62823.html>.— ЭБС «IPRbooks».

3. Ганцева Е.А. Программирование на C# : Учеб. пособие. - Воронеж : ФГБОУ ВПО "Воронежский государственный технический университет", 2012. - 135с.

4. Сергеев М.Ю. Программирование и основы алгоритмизации : Учеб. пособие. - Воронеж : ГОУВПО "Воронежский государственный технический университет", 2010. - 231 с. - 76-53; 250 экз.

8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:

MS Windows, MS Visual Studio

habr.com

docs.microsoft.com

eLibrary.ru

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Лекционная аудитория, учебные лаборатории, оснащенные компьютерными программами для проведения лабораторного практикума.

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Практикум по программированию» проводятся лабораторные работы.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Вид учебных занятий	Деятельность студента
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоения учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов;

	- работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед зачетом три дня эффективнее всего использовать для повторения и систематизации материала.