

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Смоленский государственный университет»
Кафедра информатики

«Утверждаю»
Проректор по учебно-
методической работе
_____ Ю.А. Устименко
«02» сентября 2020 г.

**Рабочая программа дисциплины
Б1.В.05 Технологии разработки программного обеспечения**

Направление подготовки: **44.03.05 Педагогическое образование (с двумя профилями подготовки)**

Направленность (профиль): **Математика, Информатика**

Форма обучения: очная

Курс – 2

Семестр – 4

Всего зачетных единиц – 2, часов – 72

Форма отчетности: зачет – 4 семестр

Программу разработал
кандидат педагогических наук, доцент Козлов С.В.

Одобрена на заседании кафедры
«26» августа 2020 г., протокол № 1

Заведующий кафедрой _____ Е.П. Емельченков

Смоленск
2020

1. Место дисциплины в структуре ОП

Дисциплина «Технологии разработки программного обеспечения» относится к части дисциплин, формируемых участниками образовательных отношений. Она изучается в 4 семестре.

При изучении данной дисциплины необходимы компетенции студентов, сформированные при изучении такой дисциплины, как «Основы информатики и вычислительной техники». Курс построен так, чтобы сформировать у студентов целостное представление о классических и инновационных технологиях разработки программного обеспечения.

Изучение курса основано на традиционных методах высшей школы, тесной взаимосвязи со смежными курсами, а также на использовании современных систем программирования.

2. Планируемые результаты обучения по дисциплине

Компетенция	Индикаторы достижения
ПК-6. Способен использовать научные знания в предметной области (информатика) в процессе формирования предметной компетенции обучающихся в рамках реализации основной общеобразовательной программы	Знать: назначение, структуру и содержание курса информатики, современное состояние и перспективы развития информатики как учебной дисциплины, ее место и роль, фундаментальное ядро современного школьного курса информатики, принципы построения методической системы обучения информатике, ее основных компонентов. Уметь: анализировать цели и содержание школьного курса информатики, проектировать образовательный процесс, использовать дидактический потенциал средств информационных технологий в реализации образовательного процесса по курсу информатики; Владеть: основными видами профессиональной деятельности учителя информатики, профессиональными навыками реализации методики обучения основным разделам курса информатики, современными информационно-коммуникационными средствами для эффективного осуществления профессиональной деятельности.
ПК-8. Способен использовать современные системные программные средства, разрабатывать моделирующие алгоритмы и реализовывать их на базе языков и пакетов прикладных программ моделирования	Знать: основные принципы и методики создания алгоритмов и программ для решения прикладных задач, основные среды для разработки программного обеспечения, базовые информационные технологии программные средства; Уметь: корректно использовать современные информационные технологии и программные средства, внедрять и адаптировать прикладное программное обеспечение; Владеть: навыками решения прикладных задач с применением современных программных средств, владеть современными языками программирования и методиками разработки и внедрения прикладного программного обеспечения.

3. Содержание дисциплины

1. **Введение в технологии разработки программных средств.** Основные понятия и определения. Жизненный цикл программных средств.
2. **Стратегии разработки программных средств и систем и реализующие их модели жизненного цикла.** Базовые стратегии разработки программных средств и систем. Каскадная стратегия разработки программных средств и систем. Инкрементная стратегия разработки программных средств и систем. Эволюционная стратегия разработки программных средств и систем. Общие сведения о каскадных моделях. Классическая каскадная модель. Каскадная модель с обратными связями. V-образная модель. Модели быстрой разработки приложений. Базовая RAD-модель. RAD-модель, основанная на моделировании предметной области. RAD-модель параллельной разработки приложений. Достоинства, недостатки и области использования RAD-моделей. Общие сведения об инкрементных моделях. Инкрементная модель с уточнением требований на начальных этапах разработки. Инкрементная модель экстремального программирования. Общие сведения об эволюционных моделях. Структурная эволюционная модель быстрого прототипирования. Спиральная модель Бозма. Упрощенные варианты спиральной модели.
3. **Выбор модели жизненного цикла для конкретного проекта.** Классификация проектов по разработке программных средств и систем. Процедура выбора модели жизненного цикла программных средств и систем. Адаптация модели жизненного цикла разработки программных средств и систем к условиям конкретного проекта.
4. **Классические методологии разработки программных средств.** Структурное программирование. Основные положения структурного программирования. Реализация основ структурного программирования в языках программирования. Графическое представление структурированных схем алгоритмов. Модульное проектирование программных средств. Методы нисходящего проектирования. Пошаговое уточнение. Проектирование программных средств с помощью псевдокода и управляющих конструкций структурного программирования. Использование комментариев для описания обработки данных. Анализ сообщений. Методы восходящего проектирования. Методы расширения ядра. Метод JSP Джексона. Основные конструкции данных. Построение структур данных. Проектирование структур программ. Этапы проектирования программного средства. Оценка структурного разбиения программы на модули. Связность модуля. Сцепление модулей.
5. **Структурное программирование в среде C#.** Раздел описаний. Раздел операторов. Библиотека Math. Логические выражения. Ветвление, безусловный переход. Циклы с условиями. Циклы с параметром. Массивы данных. Строки.
6. **Модульное программирование в среде C#.** Методы в языке C#. Способы обработки данных. Оконный ввод и вывод. Использование сложных типов данных. Создание и использование библиотек. Текстовые файлы. Типизированные файлы. Куча.
7. **Объектно-ориентированное программирование в среде C#.** Объекты. Простейшие классы. Графика на экране и графика на битовой матрице. Графика на форме. Использование цвета, режимов рисования. Шаблон заливки. Движение на экране. Компоненты вывода. Программирование объектов. Динамическое создание компонент. Наследование объектов.
8. **Визуальное программирование в среде C#.** Методы визуального программирования. Разработка графического интерфейса. Разработка меню. Стек. Очередь. Однонаправленный список. Структуры данных в динамической памяти. Двухнаправленный список. Графы. Деревья. Пути и циклы.

4. Тематический план

№ п/п	Разделы и темы	Всего часов	Формы занятий			
			лекции	практические занятия	лабораторные занятия	самостоятельная работа
1	Введение в технологии разработки программных средств	6	2	–	–	4

2	Стратегии разработки программных средств и систем и реализующие их модели жизненного цикла	6	2	–	–	4
3	Выбор модели жизненного цикла для конкретного проекта	6	2	–	–	4
4	Классические методологии разработки программных средств	8	2	–	2	4
5	Структурное программирование в среде С#	12	2	–	4	6
6	Модульное программирование в среде С#	12	2	–	4	6
7	Объектно-ориентированное программирование в среде С#	12	2	–	4	6
8	Визуальное программирование в среде С#	10	2	–	2	6
ИТОГО		72	16	–	16	40

5. Виды образовательной деятельности

Занятия лекционного типа

1. **Введение в технологии разработки программных средств.** Основные понятия и определения. Жизненный цикл программных средств.

2. **Стратегии разработки программных средств и систем и реализующие их модели жизненного цикла.** Стратегии разработки программных средств и систем. Модели жизненного цикла, реализующие каскадную стратегию разработки программных средств и систем. Модели быстрой разработки приложений. Модели жизненного цикла, реализующие инкрементную стратегию разработки программных средств и систем. Модели жизненного цикла, реализующие эволюционную стратегию разработки программных средств и систем.

3. **Выбор модели жизненного цикла для конкретного проекта.** Классификация проектов по разработке программных средств и систем. Процедура выбора модели жизненного цикла программных средств и систем. Адаптация модели жизненного цикла разработки программных средств и систем к условиям конкретного проекта.

4. **Классические методологии разработки программных средств.** Структурное программирование. Модульное проектирование программных средств. Методы нисходящего и восходящего проектирования. Методы расширения ядра. Метод JSP Джексона. Оценка структурного разбиения программы на модули.

5. **Структурное программирование в среде С#.** Раздел описаний. Раздел операторов. Библиотеки. Логические выражения. Ветвление, безусловный переход. Циклы с условиями. Циклы с параметром. Массивы данных. Строки.

6. **Модульное программирование в среде С#.** Методы в языке С#. Способы обработки данных. Оконный ввод и вывод. Использование сложных типов данных. Создание и использование библиотек. Текстовые файлы. Типизированные файлы. Куча.

7. **Объектно-ориентированное программирование в среде C#.** Объекты. Простейшие классы. Графика в среде C#. Программирование объектов. Динамическое создание компонент. Наследование объектов.

8. **Визуальное программирование в среде C#.** Методы визуального программирования. Разработка графического интерфейса. Разработка меню. Структуры данных в динамической памяти.

Занятия семинарского типа

Не предусмотрены.

Практические занятия

Не предусмотрены.

Лабораторные занятия

Лабораторное занятие №1.

Классические методологии разработки программных средств

Модель «Делать, пока не будет сделано».

1. Даны попарно координаты концов двух отрезков x_1, x_2 и x_3, x_4 .

Верно ли, что:

1) отрезки пересекаются (отрезки не пересекаются);

2) отрезки совпадают (отрезки не совпадают);

3) первый отрезок вложен во второй (второй отрезок вложен в первый);

4) отрезки имеют единственную общую точку (не имеют единственной точки пересечения).

Разработать программную документацию тестирования программы (для этой и каждой последующей задачи занятия 1). Документация должна быть представлена в виде текстового файла и содержать:

- варианты тестирования программы при устранении ее недоработок (описание синтаксической или логической

ошибки и как она была устранена, в папке проекта должны быть записаны все версии программы при устранении логических ошибок);

- блок-схему программы, основанную на трех алгоритмических конструкциях (следовании, ветвлении и цикле);

- варианты результатов работы программы при вводе различных входных данных.

Лабораторное занятие №2.

Структурное программирование в среде C#

1. Разложите натуральное число N на простые множители и определите:

1) сколько раз встречается множитель 3 в произведении;

2) сколько различных множителей в произведении;

3) наименьший множитель произведения;

4) наибольший множитель произведения.

Лабораторное занятие №3.

Структурное программирование в среде C#

1. Среди трёхзначных (N -значных) полных квадратов найти:

1) последний, у которого совпадают крайние цифры;

2) первый, у которого совпадают две первые цифры;

3) последний, у которого совпадают две последние цифры;

4) первый, у которого какая то крайняя цифра равна 5.

Лабораторное занятие №4.

Модульное программирование в среде C#

1. Сформировать случайным образом целочисленный массив из 10 натуральных двузначных чисел.

Найти среднее значение S и:

- 1) число элементов, меньших S;
- 2) элемент, наиболее близкий к S;
- 3) элемент, наиболее удалённый от S;
- 4) два соседних элемента, между которыми находится S.

Лабораторное занятие №5.

Модульное программирование в среде C#

1) Два автомата последовательно анализируют 2-значные числа 10..99, написанные на упаковках изделий. Один из них увеличивает счетчик, если сумма цифр числа оказывается больше 12, другой - увеличивает тот же счетчик, если замечает среди цифр восьмёрку, но при условии, что первый автомат не изменил показания счётчика. Какое число будет написано на упаковке, когда значение счетчика станет равным N?

2) Дети играют в «хлопки». Они по очереди считают от 1 и хлопают в ладоши, если число делится на 3 или кончается на 3. При каком счёте прозвучит хлопок N?

3) По конвейеру движутся пронумерованные от 10 до 99 пакеты с изделиями. Для проверки качества извлекается каждый 9-й и те, что содержат цифру 5. Какой двузначный номер будет иметь пакет, извлечённый N-ым по-порядку?

4) Каждого третьего по списку ученика направляют на уборку территории, а каждого девятого из списка делают звеньевым соответствующей уборочной бригады. Сколько учеников списка не будут участвовать в уборке, когда назначат N-го звеньевого?

Лабораторное занятие №6.

Объектно-ориентированное программирование в среде C#

Работа с входными данными.

1. Разработать проект обработки дат символьного потока. На вход подается 20 строк. Каждая строка представляет собой «правильную» или «неправильную» дату, записанную в одном из общепринятых форматов (вспомните форматы в базах данных, например, в MS Access). Необходимо отделить правильные даты от неправильных, отсортировать правильные даты, выявить зимние и летние даты и вывести результаты на экран.

В отчете при создании схемы алгоритма отразить его укрупненные единицы: отделение дат, их сортировку, деление на летние и зимние месяцы и вывод на экран. Для этих блоков создать собственные блок-схемы.

Лабораторное занятие №7.

Объектно-ориентированное программирование в среде C#

Проект «Игра на шахматном поле».

1. В некоторых клетках квадратной доски 4x4 находятся четырёхгранные игральные кости. Кость снимается с поля, если количество соседних по стороне непустых клеток совпадает с числом, выпавшим на кости. Все кости, для которых выполняется это свойство, снимаются с поля одновременно. Если после снятия костей появляются новые кости, которые можно снять, то они снимаются по тем же правилам. Нужно переставить одну кость так, чтобы снялось максимальное количество костей.

На вход программе даётся четыре строки по четыре числа в каждом, разделённые одним пробелом. Все числа – целые неотрицательные, не превышающие 4. Ноль означает отсутствие кости, любое положительное – наличие кости, на которой выпало указанное число. Гарантируется, что на поле есть, по крайней мере, одна кость и одна свободная клетка.

Пример ввода 1:

0 0 1 0

0 0 0 0
0 0 1 0
0 0 0 0

Пример ввода 2:

0 0 0 1
0 2 1 0
0 1 2 1
0 0 0 0

В виде ответа нужно вывести одно целое число – количество костей, которое можно снять с доски.

Пример вывода 1:

2

Пример вывода 2:

6

Лабораторное занятие №8.

Визуальное программирование в среде C#

Проект с методами.

Разработать проект, содержащий главное меню. Пункты меню должны быть названы «Задача 1», ..., «Задача 5».

Каждый из пунктов должен содержать два подпункта «Условие задачи» и «Решение». При выборе пункта меню «Условие» условие задачи должно отражаться на главной форме проекта (или в отдельном окне). При выборе пункта «Решение» должна загружаться форма, соответствующая решению, выбранной задачи. В качестве задач взять задачи занятия 1.

При этом модифицировать решение предыдущих задач таким образом, чтобы они содержали не только главный метод для запуска программы и методы, относящиеся к нажатию кнопок, а содержали собственные методы. Например, метод разложения на множители или метод вывода матрицы или метод нахождения элементов в массиве и т.д.

Задания для лабораторных работ по дисциплине «Методологии разработки программного обеспечения» предоставляется студентам на занятиях в электронном виде.

Самостоятельная работа

Текущая самостоятельная работа направлена на углубление и закрепление знаний студентов и развитие их практических умений. Она заключается в работе с лекционными материалами, поиске и обзоре литературы и электронных источников, информации по заданным темам курса, опережающей самостоятельной работе, в изучении тем, вынесенных на самостоятельную проработку, подготовке к лабораторным занятиям.

Самостоятельная внеаудиторная работа студентов состоит в:

- проработке лекционного материала, составлении конспекта лекций по темам, вынесенным на самостоятельное изучение;
- выполнении домашних заданий.

Темы для самостоятельного изучения

1. Использование структурного программирования при решении прикладных задач.
2. Обработка данных в одномерных и двумерных массивах.
3. Особенности обработки текстовых файлов.
4. Особенности обработки типизированных файлов.
5. Обработка данных в куче.
6. Организация обработки графической информации.
7. Организация движения на форме.

8. Организация стека. Задачи на стек.
9. Организация очереди. Задачи на очередь.
10. Формирование однонаправленных списков и примеры их использования.
11. Двухнаправленные списки и примеры их использования.
12. Понятие дерева. Обход дерева.
13. Задача о достижимости вершин.
14. Понятие дерева. Задачи, приводящие к деревьям.
15. Пути и циклы в графе.

В качестве индивидуальных домашних работ по визуальному программированию студентам предлагается выполнить следующие задания с использованием методологии структурного, модульного и объектно-ориентированного подходов:

- 1) Используя компонент Mask Edit, составить контролируемую программу, проверяющую правильность вставки:
 - a) пропущенного слова предложения;
 - b) пропущенных букв слова;
 - c) пропущенных цифр числового равенства;
 - d) пропущенных знаков препинания предложения.
- 2) Оформить лист-шпаргалку по одному из разделов:
 - a) механики;
 - b) оптики;
 - c) электричества;
 - d) молекулярной теории.

Лист должен содержать список формул и ключевых слов, при выборе которых данная программа выдаёт краткие справки.

- 3) На основе редактора Rich Edit составить программу – тест, в которой пользователь произвольно выбирает вопросы и ответы к ним, получая в конце итоговый результат. Темами могут служить проблемы:
 - a) психологии;
 - b) педагогики;
 - c) методики;
 - d) социологии.
- 4) Разработать удобный текстовый редактор для размещения сведений:
 - a) о телефонах;
 - b) об адресах;
 - c) о днях рождения;
 - d) о месте работы или учёбы.
- 5) Создать графический конструктор.

Консультирование студентов осуществляется в индивидуальном порядке на занятиях и во внеурочное время. Выполнение самостоятельной работы оценивается по электронным материалам, подготовленным студентами. Результаты деятельности накапливаются в индивидуальных портфолио студентов.

6. Критерии оценивания результатов освоения дисциплины (модуля)

6.1. Оценочные средства и критерии оценивания для текущей аттестации

Теоретические вопросы

1. Что подразумевается под технологией разработки ПО?
2. Что является целью структурных методов проектирования ПС?
3. Дайте определение программного продукта.
4. Дайте определение системы.
5. Назовите базовый стандарт в области ЖЦ ПС и систем.
6. Определите понятие ЖЦ программного средства или системы.

7. Определите понятие модели ЖЦ программного средства или системы.
8. Назовите основные стороны, участвующие в ЖЦ ПС и систем.
9. Назовите базовые стратегии разработки ПС и систем.
10. Охарактеризуйте сущность каскадной стратегии разработки ПС и систем, перечислите достоинства, недостатки и области применения данной стратегии.
11. Охарактеризуйте сущность инкрементной стратегии разработки ПС и систем, перечислите достоинства, недостатки и области применения данной стратегии.
12. Охарактеризуйте сущность эволюционной стратегии разработки ПС и систем, перечислите достоинства, недостатки и области применения данной стратегии.
13. Дайте сравнительную характеристику каскадной, инкрементной и эволюционной стратегий разработки ПС и систем.
14. Назовите общие черты каскадных моделей жизненного цикла.
15. Изобразите и охарактеризуйте классическую каскадную модель ЖЦ.
16. Изобразите и охарактеризуйте каскадную модель ЖЦ с обратными связями.
17. В чем заключаются ее преимущества и недостатки по сравнению с классической каскадной моделью?
18. Изобразите и охарактеризуйте V-образную модель ЖЦ. В чем заключаются ее отличия, преимущества и недостатки по сравнению с классической каскадной моделью?
19. Изобразите и охарактеризуйте V-образную модель ЖЦ с обратными связями. В чем заключаются ее преимущества и недостатки по сравнению с V-образной моделью без обратных связей?
20. Назовите основные черты RAD-моделей ЖЦ.
21. Изобразите и охарактеризуйте базовую RAD-модель ЖЦ. В чем заключаются ее отличия, преимущества и недостатки по сравнению с классической каскадной моделью?
22. Изобразите и охарактеризуйте RAD-модель ЖЦ, основанную на моделировании предметной области. В чем заключаются ее отличия, преимущества и недостатки по сравнению с базовой RAD-моделью?
23. Изобразите и охарактеризуйте RAD-модель параллельной разработки приложений. В чем заключаются ее особенности по сравнению с базовой RAD-моделью?
24. Перечислите основные достоинства, недостатки и области использования RAD-моделей.
25. Назовите общие черты инкрементных моделей ЖЦ.
26. Изобразите и охарактеризуйте инкрементную модель ЖЦ с уточнением требований на начальных этапах разработки. В чем заключаются ее особенности по сравнению с классической каскадной моделью?
27. Изобразите и охарактеризуйте инкрементную модель ЖЦ, рекомендованную ГОСТ Р ИСО/МЭК ТО 15271–2002. В чем заключаются ее особенности по сравнению с классической каскадной моделью?
28. Изобразите и охарактеризуйте инкрементную модель экстремального программирования. В чем заключаются ее особенности по сравнению с классической каскадной моделью?
29. Назовите общие черты эволюционных моделей ЖЦ.
30. Изобразите и охарактеризуйте структурную эволюционную модель быстрого прототипирования. В чем заключаются ее особенности по сравнению с другими эволюционными моделями жизненного цикла?
31. Изобразите и охарактеризуйте спиральную модель Боэма. Перечислите фазы и квадранты данной модели. Назовите достоинства и недостатки данной модели ЖЦ.
32. Изобразите и охарактеризуйте спиральную модель Института качества SQI. В чем заключаются ее особенности, достоинства и недостатки по сравнению с базовой спиральной моделью ЖЦ Боэма?
33. Изобразите и охарактеризуйте спиральную модель «win-win». В чем заключаются ее особенности, достоинства и недостатки по сравнению с базовой спиральной моделью ЖЦ Боэма?

34. Изобразите и охарактеризуйте спиральную модель Консорциума по вопросам разработки программного обеспечения. В чем заключаются ее особенности, достоинства и недостатки по сравнению с базовой спиральной моделью ЖЦ Бозма?
35. Изобразите и охарактеризуйте компонентно-ориентированную спиральную модель ЖЦ. В чем заключаются ее особенности, достоинства и недостатки по сравнению с другими спиральными моделями?
36. Охарактеризуйте схему классификации проектов по разработке ПС и систем, предложенную Институтом качества программного обеспечения SQI для выбора модели ЖЦ.
37. Перечислите категории критериев, положенных в основу схемы классификации проектов Института SQI.
38. Перечислите шаги процедуры выбора модели ЖЦ ПС и систем, предложенной Институтом SQI.
39. Назовите критерии категории характеристик требований к проекту.
40. Назовите критерии категории характеристик команды разработчиков.
41. Назовите критерии категории характеристик пользователей (заказчиков).
42. Назовите критерии категории характеристик типов проектов и рисков.
43. Перечислите характеристики проекта, влияющие на адаптацию выбранной модели ЖЦ к потребностям данного проекта.
44. Перечислите типы программных продуктов, влияющие на адаптацию выбранной модели ЖЦ к потребностям конкретного проекта.
45. Назовите основные достоинства структурного программирования.
46. Перечислите теоретические основы структурного программирования.
47. В чем заключается принцип Бомы–Джакопини при реализации структурированных программ?
48. Какие преобразования называются преобразованиями Бомы–Джакопини? Поясните назначение данных преобразований.
49. Каким образом теоретические основы структурного программирования реализуются в языках программирования?
50. Назовите основные методы графического представления структурированных схем алгоритмов.
51. Представьте графически конструкции структурного программирования, используемые в методе Дамке.
52. Нарисуйте схему алгоритма решения некоторой конкретной задачи, представленную по методу Дамке.
53. Представьте графически конструкции структурного программирования, используемые в схемах Насси–Шнейдермана.
54. Нарисуйте алгоритм решения некоторой конкретной задачи, представленный с помощью схем Насси–Шнейдермана.
55. Дайте определение идеальной модульной программы.
56. Перечислите признаки модульности программ.
57. Назовите основные достоинства и недостатки модульного проектирования.
58. Дайте классификацию классических методов структурного проектирования модульных программных средств.
59. Поясните сущность методов нисходящего проектирования.
60. Перечислите основные классические стратегии реализации нисходящего проектирования.
61. Поясните сущность и назовите способы реализации стратегии пошагового уточнения.
62. Поясните сущность проектирования программ с помощью псевдокода и управляющих конструкций структурного программирования.
63. Поясните сущность проектирования программ с помощью использования комментариев для описания обработки данных.
64. Поясните сущность стратегии анализа сообщений.
65. Поясните сущность методов восходящего проектирования. Назовите случаи, когда применение данных методов является целесообразным.

66. Перечислите и охарактеризуйте способы сочетания методов нисходящего и восходящего проектирования.
67. Сформулируйте базовое положение метода JSP Джексона.
68. Назовите и изобразите графически основные конструкции построения структур данных, используемые в методе JSP Джексона.
69. Приведите пример иерархической структуры данных, представленной в нотации Джексона.
70. Приведите пример сетевой структуры данных, представленной в нотации Джексона.
71. Приведите пример реляционной структуры данных, представленной в нотации Джексона.
72. Перечислите этапы проектирования программы по методу JSP Джексона.
73. Назовите первый этап метода JSP Джексона и на конкретном примере поясните правила его выполнения.
74. Назовите второй этап метода JSP Джексона и на конкретном примере поясните правила его выполнения.
75. Назовите третий этап метода JSP Джексона и на конкретном примере поясните правила его выполнения.
76. Назовите четвертый этап метода JSP Джексона и на конкретном примере поясните правила его выполнения.
77. Назовите пятый этап метода JSP Джексона и на конкретном примере поясните правила его выполнения.
78. Что такое связность модуля?
79. Назовите и охарактеризуйте типы и силы связности модулей.
80. Что такое сцепление модулей?
81. Назовите и охарактеризуйте типы и степени сцепления модулей.

Критерии оценивания теоретических вопросов

1. Нормы оценивания ответов на теоретические вопросы

№ п/п	Теоретический вопрос	Количество баллов (*)
1	Дан краткий ответ на поставленный вопрос	1 балл
2	Дан развернутый ответ на вопрос с анализом результатов	2 балла

(*) Возможна градация в 0,25 балла.

2. Шкала оценивания. Оценка «зачтено» за ответы на теоретические вопросы выставляется, если набрано не менее 3 баллов при ответе на три вопроса, в противном случае выставляется «не зачтено».

Задания для лабораторных занятий

Задачи по темам курса предложены к каждому лабораторному занятию.

Задания для лабораторных и самостоятельной работ, образцы решений основных типовых задач практики также размещены в системе дистанционного обучения СмолГУ (www.moodle.smolgu.ru).

Критерии оценивания выполнения лабораторных работ

1. Нормы оценивания каждой лабораторной работы:

№п/п	Структурная часть работы	Количество баллов (*)
1	Ответ на теоретические вопросы по теме лабораторной работы	1 балл
2	Демонстрация выполнения конкретного задания, предложенного для самостоятельного решения к лабораторной работе	2 балла

(*) с возможностью градации до 0,25 балла.

2. Шкала оценивания. Оценка «зачтено» за лабораторную работу выставляется, если набрано не менее 2 баллов, в противном случае за работу выставляется «не зачтено».

6.2. Оценочные средства и критерии оценивания для промежуточной аттестации

Зачетная контрольная работа

1. Разработать проект «Вычисление НОД и НОК» для двух заданных натуральных чисел.
2. Создать блок-схему алгоритма программы.
3. Подготовить документацию по проекту, содержащую варианты тестирования и отладки программы.

Критерии оценивания зачетной контрольной работы

1. Нормы оценивания работы

№ п/п	Структурная часть контрольной работы	Количество баллов (*)
1	Правильно реализован каждый метод решения	1 балл
2	Анализ результатов	2 балла

(*) Возможна градация в 0,25 балла.

2. Шкала оценивания работы:

п/п	Оценка	Количество баллов
1	Отлично	4,75-5
2	Хорошо	3,75-4,5
3	Удовлетворительно	3-3,5
4	Неудовлетворительно	менее 3

Критерий получения зачета

Зачет выставляется по результатам работы студента в течение семестра согласно Положению о текущем контроле успеваемости и промежуточной аттестации обучающихся в федеральном государственном бюджетном образовательном учреждении высшего профессионального образования «Смоленский государственный университет» (утверждено приказом ректора № 01-113 от 26.09.2019 г.; внесены дополнения приказом ректора № 01-48 от 30.04.2020).

Для получения зачета студент должен:

- уметь отвечать на теоретические вопросы, рассмотренные на лекциях;
- уметь решать задачи, предложенные на лабораторных занятиях.

7. Перечень основной и дополнительной учебной литературы

7.1. Основная литература

1. Гниденко И. Г. Технологии и методы программирования: учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. – Москва: Издательство Юрайт, 2020. – 235 с. – (Высшее образование). – ISBN 978-5-534-02816-4. – URL: <https://urait.ru/bcode/450999>
2. Зыков С. В. Программирование: учебник и практикум для вузов / С. В. Зыков. – Москва: Издательство Юрайт, 2020. – 320 с. – (Высшее образование). – ISBN 978-5-534-02444-9. – URL: <https://urait.ru/bcode/450832>
3. Крупский В. Н. Теория алгоритмов. Введение в сложность вычислений: учебное пособие для вузов / В. Н. Крупский. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2020. – 117 с. – (Высшее образование). – ISBN 978-5-534-04817-9. – URL: <https://urait.ru/bcode/454121>
4. Кувшинов Д. Р. Основы программирования: учебное пособие для вузов / Д. Р. Кувшинов. – Москва: Издательство Юрайт, 2020. – 104 с. – (Высшее образование). – ISBN 978-5-534-07559-5. – URL: <https://urait.ru/bcode/454667>
5. Лаврищева Е. М. Программная инженерия и технологии программирования сложных систем: учебник для вузов / Е. М. Лаврищева. – 2-е изд., испр. и доп. – Москва:

- Издательство Юрайт, 2020. – 432 с. – (Высшее образование). – ISBN 978-5-534-07604-2. – URL: <https://urait.ru/bcode/452137>
6. Лаврищева Е. М. Программная инженерия. Парадигмы, технологии и CASE-средства: учебник для вузов / Е. М. Лаврищева. – 2-е изд., испр. – Москва: Издательство Юрайт, 2020. – 280 с. – (Высшее образование). – ISBN 978-5-534-01056-5. – URL: <https://urait.ru/bcode/452156>
 7. Малявко А. А. Формальные языки и компиляторы: учебное пособие для вузов / А. А. Малявко. – Москва: Издательство Юрайт, 2020. – 429 с. – (Высшее образование). – ISBN 978-5-534-04288-7. – URL: <https://urait.ru/bcode/453250>
 8. Методы оптимизации: теория и алгоритмы: учебное пособие для вузов / А. А. Черняк, Ж. А. Черняк, Ю. М. Метельский, С. А. Богданович. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2020. – 357 с. – (Высшее образование). – ISBN 978-5-534-04103-3. – URL: <https://urait.ru/bcode/453567>
 9. Тузовский А. Ф. Объектно-ориентированное программирование: учебное пособие для вузов / А. Ф. Тузовский. – Москва: Издательство Юрайт, 2020. – 206 с. – (Высшее образование). – ISBN 978-5-534-00849-4. – URL: <https://urait.ru/bcode/451429>
 10. Черткова Е. А. Программная инженерия. Визуальное моделирование программных систем: учебник для вузов / Е. А. Черткова. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2020. – 147 с. – (Высшее образование). – ISBN 978-5-534-09172-4. – URL: <https://urait.ru/bcode/452749>

7.2. Дополнительная литература

1. Демин А. Ю. Информатика. Лабораторный практикум: учебное пособие для вузов / А. Ю. Демин, В. А. Дорофеев. – Москва: Издательство Юрайт, 2020. – 131 с. – (Высшее образование). – ISBN 978-5-534-08366-8. – URL: <https://urait.ru/bcode/451395>
2. Зыков С. В. Программирование. Объектно-ориентированный подход: учебник и практикум для вузов / С. В. Зыков. – Москва: Издательство Юрайт, 2020. – 155 с. – (Высшее образование). – ISBN 978-5-534-00850-0. – URL : <https://urait.ru/bcode/451488>
3. Казанский А. А. Программирование на Visual C#: учебное пособие для вузов / А. А. Казанский. – 2-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2020. – 192 с. – (Высшее образование). – ISBN 978-5-534-12338-8. – URL: <https://urait.ru/bcode/451467>
4. Лобанова Н. М. Эффективность информационных технологий: учебник и практикум для вузов / Н. М. Лобанова, Н. Ф. Алтухова. – Москва: Издательство Юрайт, 2020. – 237 с. – (Высшее образование). – ISBN 978-5-534-00222-5. – URL: <https://urait.ru/bcode/450399>
5. Мойзес О. Е. Информатика. Углубленный курс: учебное пособие для вузов / О. Е. Мойзес, Е. А. Кузьменко. – Москва: Издательство Юрайт, 2020. – 157 с. – (Высшее образование). – ISBN 978-5-9916-7051-7. – URL: <https://urait.ru/bcode/451401>
6. Токарев В. В. Методы оптимизации: учебное пособие для вузов / В. В. Токарев. – Москва: Издательство Юрайт, 2020. – 440 с. – (Высшее образование). – ISBN 978-5-534-04712-7. – URL: <https://urait.ru/bcode/454017>
7. Сысолетин Е. Г. Разработка интернет-приложений: учебное пособие для вузов / Е. Г. Сысолетин, С. Д. Ростунцев. – Москва: Издательство Юрайт, 2020. – 90 с. – (Высшее образование). – ISBN 978-5-9916-9975-4. – URL: <https://urait.ru/bcode/453345>
8. Трофимов В. В. Алгоритмизация и программирование: учебник для вузов / В. В. Трофимов, Т. А. Павловская; под редакцией В. В. Трофимова. – Москва: Издательство Юрайт, 2020. – 137 с. – (Высшее образование). – ISBN 978-5-534-07834-3. – URL: <https://urait.ru/bcode/452333>
9. Тузовский А. Ф. Проектирование и разработка web-приложений: учебное пособие для вузов / А. Ф. Тузовский. – Москва: Издательство Юрайт, 2020. – 218 с. – (Высшее образование). – ISBN 978-5-534-00515-8. – URL: <https://urait.ru/bcode/451207>
10. Черпаков И. В. Основы программирования: учебник и практикум для вузов / И. В. Черпаков. – Москва: Издательство Юрайт, 2020. – 219 с. – (Высшее образование). – ISBN 978-5-9916-9983-9. – URL: <https://urait.ru/bcode/450823>

7.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Система дистанционного обучения СмолГУ (moodle.smolgu.ru).
2. Национальный открытый университет (intuit.ru).
3. Национальная платформа открытого образования (opened.ru)

8. Материально-техническое обеспечение

Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий, обеспечивающие тематические иллюстрации, соответствующие программе дисциплины (модулей), учебная ауд. 224 на 12 посадочных мест.

Перечень материально-технического обеспечения, необходимого для реализации курса, включает в себя лабораторию, оснащенную персональными компьютерами, объединенные в сеть с выходом в Интернет, проектором и интерактивной доской, ауд.224 на 12 посадочных мест и 6 парт (12 посадочных мест).

Помещение для самостоятельной работы обучающихся оснащено компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду университета, ауд.224 на 12 посадочных мест и 6 парт (12 посадочных мест).

9. Программное обеспечение

1. Операционная система MS Windows XP, Linux.
2. Система программирования MS Visual Studio 17 (язык программирования C#).
3. Поисковые системы сети Интернет.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 03B6A3C600B7ADA9B742A1E041DE7D81B0
Владелец: Артеменков Михаил Николаевич
Действителен: с 04.10.2021 до 07.10.2022