МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ
Декан факультета экономики менеджмента и информационных технологий
_____ С.А.Баркалов
«29» июня 2018 г.

РАБОЧАЯ ПРОГРАММА

дисциплины

«Программная инженерия»

Направление подготовки 09.03.03 Прикладная информатика

Профиль Прикладная информатика в экономике

Квалификация выпускника бакалавр

Нормативный период обучения 4 года

Форма обучения очная

Год начала подготовки 2018

Автор программы _____ /Минакова О.В./

Заведующий кафедрой Информационных технологий и автоматизированного проектирования в строительстве _____ / Смольянинов А.В./

Руководитель ОПОП _____ / Аснина Н.Г./

Воронеж 2018

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

Целью данной дисциплины является изучение технологических процессов разработки программного обеспечения, методов и средств их организации.

1.2. Задачи освоения дисциплины

Задачами преподавания дисциплины являются:

– изложение основных положений технологии разработки программного обеспечения и типовых методов создания программных продуктов

– приобретение практических навыков работы в коллективе программистов, умения находить правильные технологические решения по выбору структуры программного проекта, методов тестирования и контроля исполнения с использованием современных инструментальных и методологических средств

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Программная инженерия» относится к дисциплинам базовой части блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Программная инженерия» направлен на формирование следующих компетенций:

ОПК-4 - Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью;

ОПК-7 - Способен разрабатывать алгоритмы и программы, пригодные для практического применения;

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ОПК-4	знать основные этапы и принципы создания программ
	уметь разрабатывать, согласовывать и выпускать все виды документации программных средств и оценивать качество ПО
	владеть методами и средствами представления данных, техниками извлечения требований и знаний о предметной области
ОПК-7	знать механизмы абстракции, рекурсии, типизации, повторного использования, обработки исключений
	уметь разрабатывать программы с графическим

	интерфейсом пользователя на объектно-ориентированном языке программирования
	владеть навыками работы с инструментальными средами разработки программного обеспечения

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Программная инженерия» составляет 8 з.е.

Распределение трудоемкости дисциплины по видам занятий **очная форма обучения**

Виды учебной работы	Всего часов	Семестры	
		5	6
Аудиторные занятия (всего)	108	36	72
В том числе:			
Лекции	36	18	18
Практические занятия (ПЗ)	36	-	36
Лабораторные работы (ЛР)	36	18	18
Самостоятельная работа	144	108	36
Курсовой проект	+	+	
Часы на контроль	36	-	36
Виды промежуточной аттестации - экзамен, зачет	+	+	+
Общая трудоемкость академические часы з.е.	288 8	144 4	144 4

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Прак зан.	Лаб. зан.	СРС	Всего, час
1	Основные этапы разработки ПО	Понятие технологии программирования как разработки надежного ПО: Проблемы разработки программного обеспечения. Понятие технологии и методологии. Связь технологии программирования и инженерии программного обеспечения. Жизненный цикл ПО: Стандарты в области программной инженерии. Основные и вспомогательные процессы	4	4	2	32	42

		разработки ПО. Этапы разработки ПО. Эволюция моделей разработки ПО: Каскадная и эволюционная модель. Прототипирование. Итерационная и инкрементная разработка. Спиральная модель и оценка рисков.					
2	Инженерия требований к программному обеспечению	Разработка требований к ПО: понятие требования к ПО, спецификация качества, функциональная спецификация. Анализ предметной области, опорные точки зрения и выявление требований. Спецификация требований: методы сбора требований: анкетирование, мозговой штурм, пользовательские истории, сценарии использования, построение диаграмм use-case. Написание технического задания и спецификации. Методы контроля внешнего описания. Характеристики качества ПО: Модель качества по ISO 9126. Характеристики и субхарактеристики качества программного средства. Метрики качества программного средства. Оценивание характеристик качества программных средств. Практика составления спецификации качества Проведение оценки качества.	8	6	4	24	42
3	Проектирование программ	Архитектурное проектирование: Архитектура с общим репозиторием, клиент-серверная, многоуровневая. Модели централизованного и событийного управления. Архитектура канала и фильтра. Декомпозиция программной архитектуры: Структурный подход к разработке ПО. Абстракция и декомпозиция. Модульное программирование. Характеристики программного модуля. Связанность и сцепление. Пошаговая детализация. Иерархия модулей/функций. Объектно-ориентированное проектирование: Объектная декомпозиция системной архитектуры. Объекты и классы, зависимости. Основные понятия ООП. Разработка и оценка архитектуры на основе	6	8	12	16	42

		сценариев. UML. Виды диаграмм UML. Статические диаграммы. Динамические диаграммы.					
4	Конструирование программ	Основные приемы программирования: Механизмы абстракции. Типы и структуры данных. Управляющие конструкции. Рекурсия, структурная декомпозиция, функции и передача параметров. Обработка исключений и поиск ошибок. Построение пользовательского интерфейса: Психологические и физиологические факторы. Скоростные показатели деятельности человека. Внимание человека. Понятность. Разные категории пользователей. Факторы удобства использования и принципы создания удобного ПО. Виды интерфейсов пользователя. GUI и его компоненты. Событийное управление. Тестирование – как инструмент разработки ПО: Категории программных ошибок. Типы тестов. Тестирование на этапе планирования. Тестирование на этапе проектирования. Тестирование "белого ящика" на стадии кодирования. Способы тестирования базового пути, тестирования условий, циклов. Тестирование "черного ящика". Тестирование элементов, интеграционное, системное и проверка соответствия	6	4	10	22	42
5	Вспомогательные процессы разработки ПО	Особенности сопровождения программных продуктов. Реинжиниринг. Унаследованные системы и реверсная инженерия. Управление конфигурацией: Возникновение дисциплины. Основные термины управления конфигурацией. Технологии управления SVN, git. Обзор программных средств совместной разработки и управления конфигурацией. Документирование ПО: Стандарты разработки программной документации. Составление инструкции пользователя, рекламно-технических описаний, руководства по сопровождению. Практика использования JavaDocs.	8	8	4	22	42

6	Управление разработкой программного средства	Управление программными проектами: Понятие проекта. Характеристика основных этапов программного проекта. Организация группы и принятие решений. Роли и ответственность в команде разработчиков. Построение сетевых графиков. Диаграмма Гранта. Обзор ПО поддержки управления программным проектом. Слежение за состоянием проекта. Техники проверки соответствия, оценки качества, просмотра кода. Анализ рисков. Модели оценки надежности ПО. Современные методологии разработки ПО и CASE-средства: Тяжеловесные и легковесные процессы. RUP, MFC. XP-программирование. Angle-техники: Scrum, Kanban. Понятие среды разработки (IDE, SDK) и CASE-средства. Механизмы интеграции инструментальных средств..	4	6	4	28	42
		Итого	36	**36**	**36**	**144**	**252**

5.2 Перечень лабораторных работ

1. Знакомство со средой разработки ПО и разработка простого GUI приложения с одной кнопкой
2. Создание простого GUI приложения с вводом данных
3. Реализация приложения с панелью для рисования
4. Реализация обработчиков событий мыши
5. Обработка событий клавиатуры
6. Обработка событий таймера
7. Разработка приложения для рисования графических объектов
8. Работа с массивами
9. Работа со связанным списком
10. Работа со стеком
11. Работа с файлами
12. Разработка многопоточного приложения
13. Реализация модели состояния на примере светофора
14. Разработка игрового приложения
15. Реализация графического интерфейса к БД
16. Разработка графического растрового редактора
17. Разработка графического векторного редактора
18. Разработка html-редактора
19. Разработка клиент-серверного приложения
20. Разработка приложения для анализа данных

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ)

И **КОНТРОЛЬНЫХ РАБОТ**

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсового проекта в 5 семестре для очной формы обучения.

Примерная тематика курсового проекта:

1. «Проектирование программного средства для учета семейных расходов»
2. «Проектирование программного средства для бухгалтерского учета компании авиаперевозок»
3. «Проектирование программного средства для учета списанных продукций организации по продаже офисных товаров»

Задачи, решаемые при выполнении курсового проекта:

• Анализ предметной области и существующих программных решений, формулировка требований к ПО, написание технического задания;

• Проектирование программной системы, моделирование последователи взаимодействия пользователя с системой;

• Реализация пользовательского интерфейса в соотвествии с выявленными требованиями.

Курсовой проект включат в себя создание прототипа программного продукта и оформление расчетно-пояснительной записки.

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».

Компе-тенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ОПК-4	знать основные этапы и принципы создания программ	Устный опрос на практическом занятии	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь разрабатывать, согласовывать и выпускать все виды документации программных средств и оценивать качество ПО	Разработка системной спецификации к ПС с использованием стандартных нотаций	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть методами и средствами представления данных, техниками извлечения требований и знаний о предметной	Реализация ПС в соответствии со спецификацией	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

	области			
ОПК-7	знать механизмы абстракции, рекурсии, типизации, повторного использования, обработки исключений	Устный опрос на практическом занятии	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь разрабатывать программы с графическим интерфейсом пользователя на объектно-ориентированном языке программирования	Разработка системной спецификации к ПС с использованием стандартных нотаций	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть навыками работы с инструментальными средами разработки программного обеспечения	Реализация ПС в соответствии со спецификацией	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 5, 6 семестре для очной формы обучения по двух/четырехбалльной системе:

«зачтено»

«не зачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Не зачтено
ОПК-4	знать основные этапы и принципы создания программ	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь разрабатывать, согласовывать и выпускать все виды документации программных средств и оценивать качество ПО	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть методами и средствами представления данных, техниками извлечения требований и знаний о предметной области	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ОПК-7	знать механизмы абстракции, рекурсии, типизации, повторного использования, обработки исключений	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь разрабатывать программы с графическим интерфейсом пользователя на объектно-ориентированном языке программирования	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

	владеть навыками работы с инструментальными средами разработки программного обеспечения	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

или
«отлично»;
«хорошо»;
«удовлетворительно»;
«неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.
ОПК-4	знать основные этапы и принципы создания программ	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь разрабатывать, согласовывать и выпускать все виды документации программных средств и оценивать качество ПО	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть методами и средствами представления данных, техниками извлечения требований и знаний о предметной области	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ОПК-7	знать механизмы абстракции, рекурсии, типизации, повторного использования, обработки исключений	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь разрабатывать программы с графическим интерфейсом пользователя на объектно-ориентированном языке программирования	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть навыками работы с инструментальными	Решение прикладных задач в конкретной	Задачи решены в полном	Продемонстрирован верный ход	Продемонстрирован верный ход решения в	Задачи не решены

	средами разработки программного обеспечения	предметной области	объеме и получены верные ответы	решения всех, но не получен верный ответ во всех задачах	большинстве задач	

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

1. Выберите правильное утверждение

a) *Программный модуль – это любой фрагмент кода, который программируется, компилируется и отлаживается отдельно*
b) *Программные модули основа объектно-ориентированного программирования*
c) *Программный модуль состоит из одного объекта или функции*
d) *Каждая функция программы состоит из отдельных модулей*

2. Укажите оптимальный тестовый набор для значения x лежащего в диапазоне от 0 до 10

a) *-1, 5, 11*
b) *0, 1, 2, ...,10*
c) *0, 10*
d) *2,8, 11*
e) *17*

3. Принцип абстрагирования заключается в

a) *выделение существенных аспектов системы и отвлечение от несущественных;*
b) *сложных проблемы путем их разбивки на множество меньших независимых задач*
c) *организации составных частей проблемы в иерархические древовидные структуры*
d) *в обоснованности и согласованности элементов;*

4. Методы класса определяют:

a) *какие операции можно выполнять с объектами данного класса*
b) *какие типы значений могут принимать данные-члены класса*
c) *каким образом реализуется защита данных-членов класса*
d) *как изменяется сущность*

5. Структура данных, обеспечивающая быструю вставку и очень быстрый доступ, если известен индекс

a) *Массив*
b) *Стек*
c) *Очередь*
d) *Хеш-таблица*

6. Из представленных языковых конструкций укажите объявление переменной:

a) *int a;*
b) *a=5;*

c) *return;*
d) *double countIMT (int a);*

8. Принцип «иерархического упорядочивания» заключается

a) *в том, что данные должны быть структурированы и иерархически организованы*
b) *организации составных частей проблемы в иерархические древовидные структуры*
c) *выделение существенных аспектов системы и отвлечение от несущественных;*
d) *в обоснованности и согласованности элементов*

9. Отладка - это

a) *Исправление ошибок в коде*
b) *Поиск ошибок*
c) *Проверка спецификации*
d) *Выполнение теста*

10. Событие OnClick определяет реакцию программы на:

a) *щелчок указателя мыши над компонентом*
b) *двойной щелчок указателя мыши над компонентом*
c) *перемещение указателя мыши над компонентом*

получение фокуса ввода

7.2.2 Примерный перечень заданий для решения стандартных задач

1. Для программной системы «Метеостанция» составьте иерархию заинтересованных лиц и их ролей, используя метод опорных точек.

2. Для программной системы «Библиотека ВУЗа» смоделируйте для каждого выделенного действующего лица варианты использования.

3. Составьте перечень пользовательских и функциональных требований к *программе «Графический редактор».*

4. Составьте лист оценки характеристик качества программных средств построения UML-диаграмм и проведение экспертную оценки 3-4 программных продуктов.

5. Для любой ранее разработанной программы создайте механизм защиты полей ввода, включающий:

- *Контроль границ допустимых значений;*
- *«Эхо»;*
- *Обработка исключений.*

6. Оформите программный код с использованием JavaDoc

7. Выполните анализ кода программы в соответствии с рекомендациями Code Conventions for the Java TM Programming Language

8. Подготовить кейс-тесты для проверки функций программы анализа текста, со следующими функциональными требованиями:

Программа осуществляет чтение файла input.txt и выводит в файл output.txt следующую информацию:

- кол-во букв (1), знаков препинания (2) и цифр(3) в заданном тексте; - кол-во слов (4) в заданном тесте; - кол-во предложений (5) в заданном тесте*

в *виде числовых значений на разных строках в указанном порядке (в скобках - позиция в файле)*

9. Разработать программную документацию на разработанный программный продукт

10. Для класса:class Complex {

```
private final double re;
private final double im;

public Complex(double re, double im) {
    this.re = re;
    this.im = im;
}
Complex divide(Complex c) {
    double d = c.re * c.re + c.im * c.im;
    return new Complex(
        (re * c.re + im * c.im) / d,
        (im * c.re - re * c.im) / d
    );
}
```

a) Предложите дисциплинированную обработку ошибок (не обработка исключений)

b) Представьте пример использования обработки исключений в реализации одного из методов

c) Добавьте возможность выбрасывание собственного исключения в этот класс

d) Придумайте примеры реализации определения этого класса для трех видов гарантий безопасности.

7.2.3 Примерный перечень заданий для решения прикладных задач

1. Разработать и реализовать класс Student: Фамилия, Имя, Отчество, Дата рождения, Адрес, Телефон, Факультет, Курс.

Создать массив объектов.

Вывести:

а) список студентов заданного факультета;

б) списки студентов для каждого факультета и курса;

в) список студентов, родившихся после заданного года

2. Разработать и реализовать класс Abiturient: Фамилия, Имя, Отчество, Адрес, Баллы по предметам.

Создать массив объектов.

Вывести:

а) список абитуриентов, имеющих неудовлетворительные оценки;

б) список абитуриентов, сумма баллов у которых не меньше заданной;

в) выбрать N абитуриентов, имеющих самую высокую сумму баллов, и список абитуриентов, имеющих полупроходной балл.

3. Разработать и реализовать класс Apartment: Адрес, Этаж, Количество комнат, Площадь.

Создать массив объектов.

Вывести:

а) список квартир, имеющих заданное число комнат;

б) список квартир, имеющих заданное число комнат и расположенных на этаже, который находится в определенном промежутке;

в) список квартир, имеющих площадь, превосходящую заданную.

4. Определить объект файл с поддержкой операций создания, копирования, перемещения, удаления файла, наполнения содержимым. Унаследовать от него класс «Зашифрованный файл» с поддержкой функций шифрации – дешифрации содержимого. В программе продемонстрировать функциональность разработанных классов.

5. Определить абстрактный класс Currency для работы с денежными суммами. Определить в нем методы перевода в рубли и вывода на экран. На его основе реализовать классы Dollar, Euro и Pound (фунт стерлингов) с возможностью пересчета в центы и пенсы соответственно и указанием текущего курса.. Создать класс Purse (кошелек), содержащий массив объектов этих классов в динамической памяти. Предусмотреть возможность случайного наполнения кошелька купюрами различного типа и подсчета общей суммы кошелька при изменении курса валют. В программе продемонстрировать функциональность разработанных классов.

6. Определить абстрактный класс Function (функция) с виртуальными методами вычисления значения функции y = f(x) в заданной точке x и вывода результата на экран, поиска минимума и максимума функции на заданном интервале.. Унаследовать от класса Function классы Hiperbola и Parabola, Exponenta, в которых уточняется функция f(x) и задаются коэффициенты соответствующих функций. Продемонстрировать функциональные возможности этих классов (получением значений, вычислением максимума или минимума).

7. Разработайте и реализуйте графический интерфейс пользователя для следующей системы:

1) Обработка информации по оказанию услуг ЖКХ населению.

2) Обработка информации по списанию основных средств.

3) Формирование цены продажи продукции.

4) Обработка информации по договорам с контрагентами.

5) Учет выплаты стипендии студентам.

8. Спроектируйте графический интерфейс и реализуйете компьютерную игру Реверси

9. Постройте UML-диаграммы для программной реализации программной системы «Обработка информации по расчетам с поставщиками»

10. Обоснуйте выбор программной архитектуры и разработайте проектную документацию для программной системы «Расчет рейтинга студентов».

7.2.4 Примерный перечень вопросов для подготовки к зачету

1. Понятие программа, программное средство, программное

обеспечение, программный продукт

2. Ключевые различия функционального, императивного (процедурного) и объектно-ориентированного программирования.

3. Основные этапы разработки программ

4. Инструментальные средства разработки ПО

5. Основные концепции ООП. Абстракция. Инкапсуляция. Наследование. Полиморфизм

6. UML диаграмма классов

7. Типы отношений между классами. Агрегация. Ассоциация. Композиция

8. Объявление классов. Атрибуты. Модификаторы доступа. Сигнатура метода. Конструкторы объектов и инициализаторы.

9. Объект с точки зрения ООП. Идентичность и жизненный цикл объектов. Отношения между объектами

10. Абстрактные классы и Интерфейсы. Наследование. Множественное наследование

11. Управление доступом к компонентам класса. Область действия класса и доступ к компонентам класса. Статические переменные и методы

12. Понятие типы данных. Значение типизации в программировании.

13. Структуры данных, виды. Значение структур данных.

14. Статические и динамические массивы

15. Связанный список.

16. Понятие ошибки в ПО. Виды дисциплинированной их обработки

17. Механизм обработки исключения.

18. Использование ключевых слов try, catch, finally.

19. Основные принципы разработки пользовательских интерфейсов

20. Обработка событий

21. Элементы GUI: контейнеры и элементы управления, менеджеры компоновки

22. Варианты организации взаимодействия с пользователем.

7.2.5 Примерный перечень вопросов для подготовки к экзамену

1. Цели и задачи технологий разработки ПО. Особенности современных крупных проектов разработки ПО.

2. Понятие программная инженерия. Основные, вспомогательные и организационные процессы программной инженерии.

3. Структурный подход к проектированию ПО. Сущность структурного подхода.

4. Объектно-ориентированнаяразработкапрограмм. Объектно-ориентированные языки программирования. Объектно-ориентированные методологии разработки программных систем.

5. Каскадная модель жизненного цикла ПС: содержание этапов, область применения, достоинства и недостатки.

6. Эволюционная модель жизненного цикла ПС: последовательность действий, область применения, достоинства и недостатки.

7. Спиральная модель разработки ПО: содержание этапов создания ПС,

область применения, достоинства и недостатки.

8. Инкрементальная модель разработки ПО. Развитие инкрементального подхода. XP-процессы.

9. Понятие программного проекта. Управление программным проектом. План и содержание его разделов. Составление сетевого графика работ.

11. Состав и структура коллектива разработчиков программного продукт, их функции. Составление расписания (PERT-диаграммы)

12. Управление документацией разработки программного продукта.

13. Рациональный Унифицированный Процесс. Динамические аспекты процессов: структура ЖЦ, стадии, итерации и контрольные точки.

14. Рациональный Унифицированный Процесс. Статическое содержание процесса: виды деятельности (технологические операции), рабочие продукты, исполнители и дисциплины (технологические процессы).

15. Внешнее описание программного средства и спецификация. Виды требований к ПО: системные, функциональные, характеристики качества.

16. Методы определения и формализация требований к ПО.

17. Понятие качества ПО и его многоуровневая модель. Характеристики и атрибуты качества.

18. Разработка требований к ПО: формирование и анализ, документирование, аттестация. Управление.

19. Алгоритмическая декомпозиция. Модульное программирование. Характеристики программного модуля.

20. Модели архитектур с различными способами обмена данными: репозиторий, «клиент-сервер».

21. Архитектуры с различными моделями управления.

22. Событийно-управляемые архитектуры.

23. Модели архитектур с различными подходами к обработке данных: непрерывная обработка, каналы и фильтры.

24. Объектно-ориентированная декомпозиция. Общая характеристика объектов. Виды отношений между объектами. Агрегация.

25. Абстрагирование. Общая характеристика классов. Виды отношений между классами. Ассоциации классов. Наследование. Полиморфизм. Агрегация.

26. Повторное использование компонентов. Инкапсуляция. Интерфейсы. Компонентная объектная модель (COM).

27. Принципы проектирования пользовательского интерфейса.

28. Структурное тестирование. Покрытие операторов, ветвей, условий.

29. Функциональное тестирование. Метод эквивалентного разбиения, граничных значений, причинно-следственных (функциональных) диаграмм.

30.Тестирование интеграции компонентов ПО: нисходящее и восходящее. Понятие драйвер и заглушка. Стохастическое тестирование.

31. Разработка программной документации. С-документация и П-документация.

32.Отладка ПО: цели и методы.

33. Управление конфигурацией ПО. Системы контроля версий.

Регрессионное тестирование.

34. Аттестация ПО. Оценка качества ПО.

35. Инструментальные средства разработки ПО. Автоматизация разработки ПО. CASE-средства.

36. Сопровождение ПО. Основные подходы: с целью исправления ошибок, адаптации и изменения функциональных возможностей. Решение проблемы эволюции ПО – рефакторинг, реинженерия, реверсная инженерия

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Экзамен проводится по билетам, каждый из которых содержит 2 вопроса и задачу. Каждый правильный ответ на вопрос оценивается от 1 до 5 баллов, задача оценивается в 10 баллов (5 баллов – обоснование и корректность использования выбранного метода решения и 5 баллов за полученный результат. 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	Основные этапы разработки ПО	ОПК-4, ОПК-7	Тест, практическое задание, защита лабораторных работ, требования к курсовому проекту
2	Инженерия требований к программному обеспечению	ОПК-4, ОПК-7	Тест, практическое задание, защита лабораторных работ, требования к курсовому проекту
3	Проектирование программ	ОПК-4, ОПК-7	Тест, практическое задание, защита лабораторных работ, требования к курсовому проекту
4	Конструирование программ	ОПК-4, ОПК-7	Тест, практическое задание, защита лабораторных работ, требования к курсовому

			проекту
5	Вспомогательные процессы разработки ПО	ОПК-4, ОПК-7	Тест, практическое задание, защита лабораторных работ, требования к курсовому проекту
6	Управление разработкой программного средства	ОПК-4, ОПК-7	Тест, практическое задание, защита лабораторных работ, требования к курсовому проекту

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 тест-заданий из 10 вопросов в презентации лекции. Время тестирования 15 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка «зачтено», если на 6 и более вопросов был дан правильный ответ.

Решение стандартных задач осуществляется по индивидуальным вариантам самостоятельно в течение 2 недель. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется в ходе лабораторных занятий и самостоятельно в течение 1 недели. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Защита курсовой работы, курсового проекта или отчета по всем видам практик осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного студента составляет 20 мин.

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

1. Терехов, А. Н. Технология программирования [Электронный ресурс] : учебное пособие / А. Н. Терехов. — Электрон. текстовые данные. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. — 152 c. — 978-5-4487-0070-5. — Режим доступа: http://www.iprbookshop.ru/67370.html

2. Орлов С. А. Теория и практика языков программирования [Текст] : учебник. – Москва ; Санкт-Петербург ; Нижний Новгород [и др.] : Питер, 2014

(Чехов : Первая Образцовая тип., фил. "Чеховский Печатный Двор", 2014). - 688 с.

3. Кулямин, В. В. Технологии программирования. Компонентный подход [Электронный ресурс] / В. В. Кулямин. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 590 с. — 5-9556-0067-1. — Режим доступа: http://www.iprbookshop.ru/73733.html

4. Практикум «Паттерны проектирования»/О.В.Минакова – Электронный ресурс https://sites.google.com/view/study-pattern

5. Ковалевская Е.В. Методы программирования [Электронный ресурс]:учебноепособие/КовалевскаяЕ.В.,КомлеваН.В.— Электрон.текстовые данные.— М.: Евразийский открытый институт, 2011.— 320 с.— Режим доступа: http://www.iprbookshop.ru/10784

6. Липаев, В. В. Документирование сложных программных комплексов [Электронный ресурс] : электронное дополнение к учебному пособию «Программная инженерия сложных заказных программных продуктов» (для бакалавров) / В. В. Липаев. — Электрон. текстовые данные. — Саратов : Вузовское образование, 2015. — 115 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/27294.html

7. Леоненков, А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose. Курс лекций [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по специальностям в области информационных технологий / А. В. Леоненков. — Электрон. текстовые данные. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. — 318 с. — 978-5-4487-0081-1. — Режим доступа: http://www.iprbookshop.ru/67388.html.

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:

Программное обеспечение

Среда разработки IntelliJ IDEA 2016

Информационно-справочные системы

1. Сайт ixbt.com www.ixbt.com Полная оперативная и объективная информация о персональных компьютерах, их компонентах и периферийных устройствах

2. Сайт CITForum www.citforum.ru Библиотека технических материалов по информационным технологиям

3. Сайты поддержки разработчиков ПО www.eclipse.com, www.java.com Справочная техническая документация по среде разработки Eclipseи поддержки языка программирования Java

4. Комитет по стандартизации в области радиоэлектроники и вычислительной техники www.ieee.org Нормативно-справочная

документация по вычислительной технике

5. Программная инженерия http://www.software-engin.com/ http://www.cs.st-andrews.ac.uk Авторские обзоры по современным тенденциям в инженерии ПО, обновление глав учебника «Программная инженерия»

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Компьютерный класс с предустановленной средой разработки приложений.

Мультимедийные средства: наборы файлов презентаций по темам лекционных занятий, комплект видеороликов по инсталляции, настройке и примерам использования инструментальных средств технологии программирования.

Средства мониторинга – программа тестирования по модулям дисциплины с базами тестовых вопросов.

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

По дисциплине «Программная инженерия» проводятся практические занятия и лабораторные работы, выполняется курсовой проект.

Практические занятия направлены на приобретение практических навыков разработки проектной документации. Занятия проводятся путем выполнения конкретных практических заданий в аудитории.

Лабораторные работы выполняются с использованием инструментальных средств разработки программного обеспечения с целью реализации программы соответствии с методиками, приведенными в указаниях к выполнению работ.

Методика выполнения курсовой работы изложена в учебно-методическом пособии. Выполнять этапы курсовой работы должны своевременно и в установленные сроки.

Контроль усвоения материала дисциплины производится на экзамене и защитой курсового проекта

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удается разобраться в материале, необходимо

	сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Практическое занятие	Конспектирование рекомендуемых источников. Работа с конспектом лекций, подготовка ответов к контрольным вопросам, просмотр рекомендуемой литературы. Прослушивание аудио- и видеозаписей по заданной теме, выполнение расчетно-графических заданий, решение задач по алгоритму.
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, ознакомится 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, экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.