

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Алтайский государственный технический университет им. И.И. Ползунова»

СОГЛАСОВАНО

Декан ФИТ

А.С. Авдеев

Рабочая программа дисциплины

Код и наименование дисциплины: **Б1.В.18 «Объектно-ориентированное программирование»**

Код и наименование направления подготовки (специальности): **09.03.04**

Программная инженерия

Направленность (профиль, специализация): **Разработка программно-информационных систем**

Статус дисциплины: **часть, формируемая участниками образовательных отношений (вариативная)**

Форма обучения: **очная**

Статус	Должность	И.О. Фамилия
Разработал	доцент	В.С. Троицкий
Согласовал	Зав. кафедрой «ПМ»	Е.Г. Боровцов
	руководитель направленности (профиля) программы	С.А. Кантор

г. Барнаул

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Код компетенции из УП и этап её формирования	Содержание компетенции	В результате изучения дисциплины обучающиеся должны:		
		знать	уметь	владеть
ПК-1	готовностью применять основные методы и инструменты разработки программного обеспечения	Знать основные методы и инструменты разработки программного обеспечения с использованием инструментальных средств, в том числе, особенности объектно-ориентированного программирования на языках C/C++,C#, Java.	Уметь использовать основные методы и инструменты разработки программного обеспечения, в том числе, разрабатывать программное обеспечение на языках C/C++,C#, Java с использованием объектно-ориентированного подхода.	Владеть основными методами и инструментами разработки программного обеспечения, в том числе, технологий объектно-ориентированного программирования на языках C/C++,C#, Java
ПК-15	способностью готовить презентации, оформлять научно-технические отчеты по результатам выполненной работы, публиковать результаты исследований в виде статей и докладов на научно-технических конференциях	Правила оформления научно-технических отчетов, статей, докладов и технической документации по разработанным программным продуктам	Уметь готовить презентации, оформлять научно-технические отчеты по результатам выполненной работы, публиковать результаты исследований в виде статей и докладов на научно-технических конференциях, в том числе, оформлять документацию по разработанным программным продуктам.	Навыками написания научно-технических отчетов.
ПК-3	владением навыками использования различных технологий разработки программного обеспечения	Знать технологии разработки программного обеспечения, в том числе, объектно-ориентированную технологию разработки программного обеспечения (АТД, наследование, полиморфизм и инкапсуляция).	Уметь использовать различные технологии разработки программного обеспечения, в том числе ООП, когда программы представляют из себя совокупность объектов, каждый из которых является	Владение навыками использования различных технологий разработки программного обеспечения, в том числе, объектно-ориентированной технологией.

Код компетенции из УП и этап её формирования	Содержание компетенции	В результате изучения дисциплины обучающиеся должны:		
		знать	уметь	владеть
		Разницу между процедурным и объектно-ориентированным программированием.	экземпляром определенного класса, а классы образуют иерархию наследования.	

2. Место дисциплины в структуре образовательной программы

Дисциплины (практики), предшествующие изучению дисциплины, результаты освоения которых необходимы для освоения данной дисциплины.	Архитектура вычислительных систем, Введение в алгоритмы и основы технологий разработки программ, Введение в информатику, Основы программирования, Теоретические основы информатики
Дисциплины (практики), для которых результаты освоения данной дисциплины будут необходимы, как входные знания, умения и владения для их изучения.	Архитектурное проектирование и паттерны программирования, Выпускная квалификационная работа, Научно-исследовательская работа, Преддипломная практика, Современные средства промышленной разработки программного обеспечения, Технология командной разработки программного обеспечения

3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося

Общий объем дисциплины в з.е. /час: 6 / 216

Форма промежуточной аттестации: Экзамен

Форма обучения	Виды занятий, их трудоемкость (час.)				Объем контактной работы обучающегося с преподавателем (час)
	Лекции	Лабораторные работы	Практические занятия	Самостоятельная работа	
очная	34	51	0	131	99

4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

Форма обучения: очная

Семестр: 3

Лекционные занятия (34ч.)

1. Введение. Инкапсуляция. Понятие классов и объектов в C++. {лекция с разбором конкретных ситуаций} (2ч.)[3,4] Технологии и методы разработки программного обеспечения с использованием инструментальных средств. Особенности объектно-ориентированного программирования. Обзор технологий разработки программного обеспечения, отличие процедурного и объектно-ориентированного подходов. Правила оформления технической документации по разработанным программным продуктам.

Запись структур в стиле C++

Замена структуры struct на класс class

Добавление методов в классы

Ограничение доступа (public: и private:)

Общепринятый способ размещения классов по файлам

Динамические объекты на C++

2. Язык Java. Инструменты разработки программного обеспечения на языке Java, основные сведения о языке Java. {лекция с разбором конкретных ситуаций} (2ч.)[5] Инструментальные средства.

Язык программирования Java.

Консольное приложение на языке Java.

Запись классов и объектов на Java.

eclipse и netbeans

3. Язык c#. Инструменты разработки программного обеспечения на языке c#, основные сведения о языке c#. Классы и объекты на c# {лекция с разбором конкретных ситуаций} (2ч.)[3,6] Инструментальные средства

Основные сведения о языке C# и NET Framework

Консольное приложение C#

Классы на C#

Структуры на C#

4. Объекты в языках C++, Java, C# (Часть1) {лекция с разбором конкретных ситуаций} (2ч.)[2,3] Объектно-ориентированная технология разработки программного обеспечения. Вызов метода у объекта класса и внутри класса

Дополнительные сведения о Java и C# (Константы в C++, Java, C#;Пакеты на Java;Массивы объектов на Java и C#;Передача параметров на C++, C# и Java)

Перегрузка методов

Указатель this

Дружественные функции на C++

Перегрузка операторов на C++ и C#

Работа со строками на C++, Java и C#

5. Объекты в языках C++, Java, C# (Часть2) {лекция с разбором конкретных ситуаций} (4ч.)[2,3] Объектно-ориентированная технология разработки

программного обеспечения. Статические поля и методы на C++, Java и C#

Конструкторы инициализации и деструктор на C++, Java и C#

Конструктор копирования на C++

"Мелкое"(shallow) и "глубокое"(deep)копирование на C++

Перегрузка оператора присваивания на C++

Исключения на C++, Java и C#

6. Наследование. Производные классы в C++, Java, C#. {лекция с разбором конкретных ситуаций} (4ч.)[2,3] Объектно-ориентированная технология разработки программного обеспечения. Производный класс, наследование полей и методов и добавление новых полей и методов на C++

Модификатор доступа protected

Модификатор в заголовке наследуемого класса

Множественное наследование

Наследование на языке java

Наследование в языке C#

Перегрузка методов и явное указание класса вызываемого метода

Конструкторы в производных классах

Перегрузка оператора присваивания для производного класса на C++

Библиотека ввода-вывода на C++

7. Полиморфизм. Абстрактные классы. Интерфейсы. {лекция с разбором конкретных ситуаций} (4ч.)[2,3] Объектно-ориентированная технология разработки программного обеспечения. Виртуальные методы на C++, Java, C#.

Полиморфизм

Вызов виртуальной функции на C++ и C# после присваивания указателей

Таблицы виртуальных функций для полиморфных классов. Принципы вызова виртуальных функций.

Абстрактные классы на C++, Java, C#. Чисто виртуальные функции C++, абстрактные функции на Java, C#

Интерфейсы на Java и C#.

Клонирование объектов на Java и C#. Встроенный интерфейс клонирования.

Мелкое и глубокое копирование на Java и C#

8. Шаблоны (templates). Обобщенное программирование(Generics). {лекция с разбором конкретных ситуаций} (2ч.)[2,3] Объектно-ориентированная технология разработки программного обеспечения. Шаблоны функций (template) на C++

Шаблоны классов на C++

Обобщенные методы и классы (generics) на Java

Обобщенные методы и классы (generics) на C#

Делегаты на C#

9. Контейнеры STL библиотеки. Коллекции на Java и C#. {лекция с разбором конкретных ситуаций} (2ч.)[2,3] Инструменты разработки ПО на C++, в частности, библиотека STL.

Краткие сведения о контейнере list, adapter, functor

Алгоритмы

Контейнер vector библиотеки STL с объектами базового и производного классов
Контейнер map библиотеки STL

Инструменты разработки ПО на Java и C# - Коллекции, Обобщенные коллекции

10. Примеры ООП. Объектно-ориентированная технология разработки простейших GUI приложений. {лекция с разбором конкретных ситуаций} (2ч.)[2,3] Простое GUI приложение на Qt.

Простое приложение Windows Forms на C++ и C#

Простое GUI приложение на Java

Простое приложение для Android

11. Примеры ООП. Объектно-ориентированная технология разработки GUI приложений. {лекция с разбором конкретных ситуаций} (2ч.)[2,3] Все примеры рассматриваются на Windows Forms (C++ и C#), Qt (C++), Java (swing), Android (java).

Несколько форм.

Меню.

Окна диалогов.

Чтение и запись файлов.

Сериализация.

Списки и комбинированные списки, таблицы и пр.

12. Примеры ООП. Объектно-ориентированная технология разработки GUI приложений (Растровая графика). {лекция с разбором конкретных ситуаций} (2ч.)[2,3] Все примеры рассматриваются на Windows Forms (C++ и C#), Qt (C++), Java (swing), Android (java).

Классы для работы с графикой.

Загрузка изображений.

Обработка событий мыши / событий касания экрана.

Преобразование фрагмента изображения

по отдельным точкам.

Алгоритмы масштабирования изображений

13. Примеры ООП. Объектно-ориентированная технология разработки GUI приложений (Векторная графика). {лекция с разбором конкретных ситуаций} (2ч.)[2,3] Все примеры рассматриваются на Windows Forms (C++ и C#), Qt (C++), Java (swing), Android (java).

Рисование геометрических фигур, закрашивание.

Использование таймера для анимации изображений

14. Примеры ООП. Объектно-ориентированная технология разработки GUI приложений (Обработка текста). {лекция с разбором конкретных ситуаций} (2ч.)[2,3] Все примеры рассматриваются на Windows Forms (C++ и C#), Qt (C++), Java (swing), Android (java).

Классы для обработки текста.

Выбор и установка шрифта.

Операции над текстом.

Добавка, удаление фрагмента текста.

Установка другого шрифта на фрагменте текста.

Выравнивание строки.

Загрузка текста из файла и сохранение текста.

Обработка события нажатия клавиш.

Лабораторные работы (51ч.)

1. Абстрактные типы данных. Инкапсуляция. {работа в малых группах} (2ч.)[2,3] Изучить различные технологии разработки ПО, в том числе отличие процедурного и объектно-ориентированного подходов.

1) Разработать тип данных и набор функций для работы с ним (не использовать конструкции ООП)

Во всех вариантах должен быть разработан комплексный тип данных и реализован набор функций для работы с ним.

- Процедура инициализации (Init);

- Процедура ввода с клавиатуры (Read);

- Процедура вывода на экран (Display);

- Процедура сложения (Add), ее два аргумента и возвращаемое значение вашего типа данных. Что понимается под "сложением" поясняется в варианте задания;

- Процедура по вашему варианту задания.

В main продемонстрировать работу с переменными данного типа.

2) Придумать свой вариант задания и реализовать его.

3) Переделать любой из вариантов (из задания 1 или который вы придумали в задании 2) в стиле ООП.

Язык программирования C++.

2. Поля класса могут быть объектами другого класса {творческое задание} (2ч.)[2,3] Разработать ПО на языке C++ с использованием объектно-ориентированного подхода, а именно:

1. Реализовать вспомогательный класс по варианту задания с указанными полями и методами. Должны быть также реализованы методы Init и Display. Продемонстрировать работу всех методов в main.

2. Создать основной класс по варианту задания, полями которого являются объекты вспомогательного класса. Реализовать указанные методы. В main продемонстрировать использование объектов основного класса. В основном классе должно быть строковое поле названия в виде массива char.

3. Придумать свой вариант задания (только придумать и написать, его реализовывать не нужно).

Язык программирования C++

3. Создание динамических объектов {творческое задание} (2ч.)[2,3] Разработать ПО на языке C++ с использованием объектно-ориентированного подхода, а именно:

В функции main ЛР1 продемонстрировать использование одного динамического объекта, а также динамического массива объектов. Использовать new, delete, delete[] . При обращении к методам использовать (*a) и ->.

В ЛР2 заменить поля объектов вспомогательного класса в основном классе на массив объектов с изменением методов основного класса.

4. Знакомство с языком Java {творческое задание} (2ч.)[2,3] Разработать ПО на языке Java с использованием объектно-ориентированного подхода, а именно: Выполнить программу как консольное приложение (compile.bat, run.bat) и в среде (Eclipse или NetBeans). Использовать ввод-вывод (printf, scanner), использовать при выводе русские буквы.

Программа должна вычислять значение выражения, в зависимости от введенных данных, после проверки соответствующих условий. Проверку следует оптимизировать, чтобы количество условий , && , || было минимальным.

5. Несколько классов на Java {творческое задание} (2ч.)[2,3] Разработать ПО на языке Java с использованием объектно-ориентированного подхода и инструмента разработки eclipse, а именно:

1.

Реализовать на java в среде eclipse класс с двумя полями first и second по вариантам. Реализовать методы: Init, GetFirst и GetSecond, Display (вывод всех полей), Read и метод, указанный в варианте. Привести пример работы.

2.

Добавить основной класс, в котором имеются 2 поля вспомогательного класса, одно или два дополнительных поля и дополнительный метод.

3.

в методе main добавить массив объектов из 5 элементов ч 1 и проинициализировать его поэлементно случайными значениями. Кроме этого, добавить массивы int или double в зависимости от варианта (поля first, second целые или вещественные). В эти массивы записать в цикле результаты вызовов метода, указанного в ч. 1 и затем в новом цикле вывести значения элементов этого массива и максимальное значение через printf. Также через Display вывести значения полей всех элементов массива объектов.

6. Знакомство с языком C# {творческое задание} (2ч.)[2,3] Разработать ПО на языке C# с использованием объектно-ориентированного подхода, а именно: Написать консольное приложение на c# по вводу небольшого массива с консоли и созданию нового целочисленного или вещественного массива по варианту задания. Ввод выполняется в одну строку, причем пользователь может вводить через несколько пробелов. При преобразовании в число проверить правильность ввода и при необходимости вывести сообщение о повторном вводе. вывод произвольный (в строку, столбец).

7. Несколько классов на C# {творческое задание} (2ч.)[2,3] Разработать ПО на языке C# с использованием объектно-ориентированного подхода, а именно:

1.

Реализовать на c# в среде Visual Studio класс по варианту задания. Для полей

добавить свойства. Использовать методы Read, Display для ввода и вывода значений всех полей. Для задания/получения значений одного из полей использовать свойства, остальные через Init. Добавить метод add с двумя аргументами объектов класса, который в качестве результата дает объект того же класса, являющийся "смесью" аргументов по варианту задания. В main создать массив объектов и выполнить вычисления от элементов массива по варианту задания.

2.

В новом проекте создать структуру (struct) с полями и методами класса из ч.1 . Продемонстрировать различие между присваиванием объектов в классе и структуре.

3.

Добавить основной класс, в котором имеются 3 поля вспомогательного класса, одно дополнительное поле и дополнительные методы.

8. Перегрузка методов, операторов, способы передачи параметров и возврата результата и др. {творческое задание} (4ч.)[2,3] Разработать ПО на языках C++, Java и C# с использованием объектно-ориентированного подхода, а именно:

1. Поля из объектов вспомогательного класса в ЛР3, ЛР5, ЛР7 заменить на массив объектов

2. Метод в ЛР1, дополнить двумя методами, возвращающие значения через указатель (*) и через ссылку (&).

3. Метод в ЛР5, дополнить методом, возвращающий значение через вспомогательный класс.

4. Метод в ЛР7, дополнить методом, возвращающий значение через ref и out.

5. В ЛР1, 5, 7 добавить метод использующий this.

6. В ЛР1 добавить дружественную функцию

9. Обработка строк {творческое задание} (2ч.)[2,3] Разработать ПО на языках C++, Java и C# с использованием объектно-ориентированного подхода, а именно:

1. В ЛР2 массив char заменить на строку и продемонстрировать обработку строк текста.

2. Доработать ЛР5,7, продемонстрировать обработку строк текста.

10. Ключевое слово static, конструкторы инициализации и конструктор копии, оператор присваивания и исключения {творческое задание} (4ч.)[2,3]

Разработать ПО на языках C++, Java и C# с использованием объектно-ориентированного подхода, а именно:

1. В ЛР1,5,7 добавить статический метод.

2. Метод Init в ЛР1,5,7 заменить на конструктор со всеми параметрами, добавить конструктор с одним параметром и конструктор без параметров.

3. В ЛР2 добавить конструктор инициализации и конструктор копии, перегрузить оператор присваивания.

4. В ЛР1,5,7 реализовать 2 исключения.

11. Массивы {творческое задание} (2ч.)[2,3] Разработать ПО на языках C++, Java и C# с использованием объектно-ориентированного подхода, а именно:

В ЛР1,5,7 реализовать статический метод для создания и обработки двумерного массива по варианту задания.

12. Наследование {творческое задание} (3ч.)[2,3] Разработать ПО на языках C++, Java и C# с использованием объектно-ориентированного подхода, а именно: В ЛР1,5,7 создать производный класс, продемонстрировать применение private, public, protected, перегрузку методов, доступ к полям базового класса, вызов методов базового класса, работу конструкторов, перегрузку оператора присваивания (C++), создание статических и динамических объектов базового и производного классов, заменить метод Display на операции << для C++ и toString для Java и C#.

13. Наследование 2 {творческое задание} (2ч.)[2,3] Разработать ПО на языках C++, Java и C# с использованием объектно-ориентированного подхода, а именно: В ЛР2, ЛР5, ЛР7, ЛР10 создать производные классы к вспомогательному классу. Доработать основной класс для работы с объектами как базового вспомогательного, так и производного вспомогательного классов.

14. Наследование 3. Полиморфизм. {творческое задание} (4ч.)[2,3] Разработать ПО на языках C++, Java и C# с использованием объектно-ориентированного подхода, а именно:

Для программ на C++ и C# из ЛР12 задать виртуальную функцию. Продемонстрировать изменение работы программы, если определяемая функция не виртуальная.

Основной класс в ЛР13 для C++, Java, C# сделать абстрактным. Основной класс в ЛР13 для C++, Java, C# заменить на интерфейсы. Основной класс в ЛР5 и в ЛР7 сделать допускающим клонирование. Продемонстрировать варианты с мелким и глубоким клонированием.

15. Шаблоны {творческое задание} (2ч.)[2,3] Разработать ПО на языках C++, Java и C# с использованием объектно-ориентированного подхода, а именно:

Продемонстрировать использование шаблонов на C++, Java и C#.

Создать делегат для метода и продемонстрировать работу экземпляров делегата.

16. Библиотека STL {творческое задание} (2ч.)[2,3] Разработать ПО на языках C++, Java и C# с использованием объектно-ориентированного подхода и дополнительных инструментальных средств, а именно:

Чтение данных из текстового файла с занесением их в контейнер, сортировка контейнера, вывод содержимого контейнера после сортировки на консоль и в файл. Поиск элементов в контейнере. Содержимое контейнера это объекты базового и производного классов из предыдущих ЛР.

17. Использование коллекций Java, C# {творческое задание} (2ч.)[2,3] Разработать ПО на языках C++, Java и C# с использованием объектно-ориентированного подхода и дополнительных инструментальных средств, а именно: Продемонстрировать работу с ArrayList в ЛР7,12. Перегрузить метод compareTo интерфейса Comparable и выполнить сортировку.

18. Примеры ООП. Создание простейших GUI приложений. {творческое задание} (2ч.)[2,3] Разработать ПО на языках C++, Java и C# с использованием объектно-ориентированного подхода и дополнительных инструментальных

средств, а именно:

Создать приложения с графическим пользовательским интерфейсом для вычисления значения функции на языках C++ (QT и VC++), Java (JDK, Android), C#. Итого 5 программ. Использовать радиокнопки, TextBox, Label и Button.

19. Примеры ООП. Создание GUI приложений. {творческое задание} (2ч.)[2,3] Разработать ПО на языках C++, Java и C# с использованием объектно-ориентированного подхода и дополнительных инструментальных средств, а именно:

Разработать 5 программ с графическим пользовательским интерфейсом на Qt, Visual Studio C++, Java SE, C# (Visual studio), Android Studio.

Продемонстрировать использование меню, списков выбора (ListWidget, listBox, JList, ListView), таблиц (QTableWidget, listView, JTable, GridView), ComboBox, Spinner и диалогов выбора имени файла.

20. Примеры ООП. Создание GUI приложений (Растровая графика). {творческое задание} (2ч.)[2,3] Разработать ПО на языках C++, Java и C# с использованием объектно-ориентированного подхода и дополнительных инструментальных средств, а именно:

Разработать 5 программ: C++ Qt, Visual Studio C++, Java Se, C#, Android

Продемонстрировать работу с растровыми изображениями (загрузка и сохранение, копирование части, выделение мышкой части изображения и его преобразование отдельно для R, G, B каналов).

21. Примеры ООП. Создание GUI приложений (Векторная графика). {творческое задание} (2ч.)[2,3] Разработать ПО на языках C++, Java и C# с использованием объектно-ориентированного подхода и дополнительных инструментальных средств, а именно:

Разработать 5 программ на C++ Qt, Java Se, C#, Android

Продемонстрировать масштабирование фрагмента изображения, выравнивание изображения по гистограмме, формирование векторного рисунка поверх фонового изображения и его анимацию.

22. Примеры ООП. Создание GUI приложений (Обработка текста). {творческое задание} (2ч.)[2,3] Разработать ПО на языках C++, Java и C# с использованием объектно-ориентированного подхода и дополнительных инструментальных средств, а именно:

Разработать 5 программ для Qt, Visual Studio C++, Java SE, Visual Studio C#, Android

Продемонстрировать работу с текстом (выбрать шрифт и его параметры + поиск и замена) и обработку ввода с клавиатуры.

Самостоятельная работа (131ч.)

1. Изучение теоретического материала(16ч.)[2,3] Изучение лекционного материала, запуск демонстрационных примеров из курса лекций.

2. Подготовка к лабораторным работам и их защите(34ч.)[2,3] Доработка и отладка ПО

Оформление научно-технического отчета по результатам выполненной работы
Подготовка к защите отчета

3. Разработка курсового проекта {разработка проекта} (45ч.)[1,2,3] 1) Выбор темы работы (Примеры тем: Разработка на основе объектно-ориентированного подхода программного обеспечения для создания и обработки данных агентства по перевозке грузов; Разработка компьютерной игры «Аквамир: от малька до акулы»)

2) Разработка соответствующего ПО с использованием ООП

3) Оформление научно-технического отчета по выполненной работе (Оформление пояснительной записки)

4) Защита КП

4. Подготовка к экзамену(36ч.)[2,3,4,5,6,7,8,9,10] Подготовка к экзамену

5. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине

Для каждого обучающегося обеспечен индивидуальный неограниченный доступ к электронно-библиотечным системам: Лань, Университетская библиотека он-лайн, электронной библиотеке АлтГТУ и к электронной информационно-образовательной среде:

1. Потупчик, А. И. Оформление пояснительной записки к курсовому проекту (курсовой работе) : методические указания к курсовому проектированию для бакалавров / А. И. Потупчик ; АлтГТУ им. И. И. Ползунова. – Барнаул, Изд-во АлтГТУ, 2015 . – 26 с. Прямая ссылка: <http://elib.altstu.ru/eum/download/pm/oformlenie-kp.pdf>

6. Перечень учебной литературы

6.1. Основная литература

2. Ашарина, И.В. Язык С++ и объектно-ориентированное программирование в С++. Лабораторный практикум [Электронный ресурс] : учеб. пособие / И.В. Ашарина, Ж.Ф. Крупская. — Электрон. дан. — Москва : Горячая линия-Телеком, 2016. — 232 с. — Режим доступа: <https://e.lanbook.com/book/107633>. — Загл. с экрана.

3. Бабушкина, И.А. Практикум по объектно-ориентированному программированию [Электронный ресурс] : учебное пособие / И.А. Бабушкина, С.М. Окулов. — Электрон. дан. — Москва : Издательство "Лаборатория знаний", 2015. — 369 с. — Режим доступа: <https://e.lanbook.com/book/66121>. — Загл. с экрана.

6.2. Дополнительная литература

4. Виссер, Д. Разработка обслуживаемых программ на языке Java [Электронный ресурс] / Д. Виссер ; пер. с англ. Р. Н. Рагимова. — Электрон. дан.

— Москва : ДМК Пресс, 2017. — 182 с. — Режим доступа: <https://e.lanbook.com/book/105834>. — Загл. с экрана.

5. Виссер, Д. Разработка обслуживаемых программ на языке С# [Электронный ресурс] : руководство / Д. Виссер ; пер. с англ. Рагимова Р.Н.. — Электрон. дан. — Москва : ДМК Пресс, 2017. — 192 с. — Режим доступа: <https://e.lanbook.com/book/93279>. — Загл. с экрана.

6. Дэвид, Х. Разработка приложений Java EE 7 в NetBeans 8 [Электронный ресурс] : руководство / Х. Дэвид ; пер. с англ. Киселев А.Н.. — Электрон. дан. — Москва : ДМК Пресс, 2016. — 348 с. — Режим доступа: <https://e.lanbook.com/book/97342>. — Загл. с экрана.

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

7. Официальный сайт Qt с открытым исходным кодом
<https://www.qt.io/developers/>

8. Официальная страница Android Studio
<https://developer.android.com/studio/intro/>

9. Официальный сайт JAVA
<https://www.java.com>

10. Visual Studio
<https://visualstudio.microsoft.com>

8. Фонд оценочных материалов для проведения текущего контроля успеваемости и промежуточной аттестации

Содержание промежуточной аттестации раскрывается в комплекте контролирующих материалов, предназначенных для проверки соответствия уровня подготовки по дисциплине требованиям ФГОС, которые хранятся на кафедре-разработчике РПД в печатном виде и в ЭИОС.

Фонд оценочных материалов (ФОМ) по дисциплине представлен в приложении А.

9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

Для успешного освоения дисциплины используются ресурсы электронной информационно-образовательной среды, образовательные интернет-порталы, глобальная компьютерная сеть Интернет. В процессе изучения дисциплины происходит интерактивное взаимодействие обучающегося с преподавателем через личный кабинет студента.

№пп	Используемое программное обеспечение
1	Visual Studio
2	NetBeans IDE

№пп	Используемое программное обеспечение
3	Qt Creator Open Source
4	Windows
5	LibreOffice
6	Linux
7	Антивирус Kaspersky

№пп	Используемые профессиональные базы данных и информационные справочные системы
1	Бесплатная электронная библиотека онлайн "Единое окно к образовательным ресурсам" для студентов и преподавателей; каталог ссылок на образовательные интернет-ресурсы (http://Window.edu.ru)
2	Национальная электронная библиотека (НЭБ) — свободный доступ читателей к фондам российских библиотек. Содержит коллекции оцифрованных документов (как открытого доступа, так и ограниченных авторским правом), а также каталог изданий, хранящихся в библиотеках России. (http://нэб.рф/)

10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Наименование специальных помещений и помещений для самостоятельной работы
учебные аудитории для проведения занятий лекционного типа
лаборатории
помещения для самостоятельной работы
учебные аудитории для проведения групповых и индивидуальных консультаций
учебные аудитории для проведения курсового проектирования (выполнения курсовых работ)
учебные аудитории для проведения текущего контроля и промежуточной аттестации

Материально-техническое обеспечение и организация образовательного процесса по дисциплине для инвалидов и лиц с ограниченными возможностями здоровья осуществляется в соответствии с «Положением об обучении инвалидов и лиц с ограниченными возможностями здоровья».