

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Саратовский государственный технический университет
имени Гагарина Ю.А.»

Кафедра «Информационная безопасность автоматизированных систем»

РАБОЧАЯ ПРОГРАММА

по дисциплине

С.1.1.16 «Языки программирования»

специальности подготовки

10.05.03 "Информационная безопасность автоматизированных систем"
Специализация №9 "Создание автоматизированных систем
в защищенном исполнении"

форма обучения – очная
курс – 1
семестр – 1,2
зачетных единиц – 4,5
часов в неделю – 4,5
всего часов – 324
в том числе:
лекции – 64
лабораторные занятия – 48
практические занятия - 32
самостоятельная работа – 180
экзамен – 1,2 семестр
ргр – 2 семестр

1. Цели и задачи дисциплины

Цель преподавания дисциплины: является изучение общих принципов построения и использования современных языков программирования высокого уровня; получение навыков разработки программ на языке программирования высокого уровня (объектно-ориентированное программирование)

Задачи изучения дисциплины:

- умение применять знания на практике
- улучшение способности учиться
- умение понять поставленную задачу
- умение формулировать результат
- основные информационные технологии, используемые в автоматизированных системах;
- общие принципы построения и использования современных языков программирования высокого уровня;
- язык программирования высокого уровня (объектно-ориентированное программирование);
- способы обработки исключительных ситуаций;
- основные структуры данных и способы их реализации на языке программирования;

2. Место дисциплины в структуре ООП ВО

Дисциплина «Языки программирования» относится к числу дисциплин базовой части профессионального цикла.

Для успешного усвоения данной дисциплины необходимо, чтобы студент владел знаниями, умениями и навыками, сформированными в процессе изучения дисциплин:

«Информатика» - знать формы и способы представления данных в персональном компьютере, классификацию современных компьютерных систем, типовые структуры и принципы организации компьютерных сетей; уметь применять типовые программные средства сервисного назначения (средства восстановления системы после сбоев, дефрагментации и очистки диска и т.п.), пользоваться сетевыми средствами и внешними носителями информации для обмена данными; владеть навыками обеспечения безопасности информации с помощью типовых программных средств, навыками поиска и обмена информацией в глобальной сети Интернет;

Дисциплина «Языки программирования» является предшествующей для изучения следующих базовых дисциплин: «Технологии и методы программирования», «Разработка и эксплуатация защищенных автоматизированных систем», «Программно-аппаратные средства обеспечения информационной безопасности», «Организация ЭВМ и вычислительных систем».

3. Требования к результатам освоения дисциплины

Изучение дисциплины направлено на формирование следующих компетенций:

способность использовать языки, системы и инструментальные средства программирования в профессиональной деятельности (ПК-3);

способностью к освоению новых образцов программных, технических средств и информационных технологий (ПК-8)

способностью применять современные методы исследования с использованием компьютерных технологий (ПК-10);

Студент должен знать:

- современные подходы к разработке программного обеспечения
- принципы работы приложений, и применять их для решения классических задач программирования

Студент должен уметь:

- самостоятельно выбирать наиболее подходящие инструменты для решения типовых задач, связанных с обработкой и хранением данных
- создавать программы, используя языки высокого уровня и соответствующие ожидаемому результату, полученному на основании анализа технического задания
- выявлять общие характеристики у разнотипных данных и обрабатывать их с применением принципов объектно-ориентированного проектирования
- Оценивать качество разработанного программного кода

Студент должен владеть:

- инструментами разработки приложений.

4. Распределение трудоемкости (час.) дисциплины по темам и видам занятий

№ мо- ду- ля	№ неде- ли	№ те- мы	Наименование темы	Часы/ Из них в интерактивной форме				
				Всего	Лек- ции	Лабора- торные	Прак- тичес- кие	СРС
1	2	3	4	5	6	7	8	9
1 семестр								
1	1	1	Что такое программирование	2	2	-	-	-
1	2	2	Синтаксические конструкции	12	6	4	2	-
1	5	3	Типы данных. Условные операторы	4	2	-	2	-
1	6	4	Оператор варианта. Циклы	6	4	-	2	-
1	8	5	Массивы	8	4	2	2	-
2	10	6	Функции	8	4	2	2	-
2	12	7	Указатели	28	2	4	2	20
2	13	8	Память	28	4	2	2	20
2	15	9	Пользовательские типы данных	26	2	2	2	20
2	16	10	Сборка программы	22	2	-	-	20
2 семестр								
1	1	1	Концепции ООП.	4	4	-	-	-
1	3	2	Классы	36	6	8	2	20
1	6	3	Наследование	56	6	8	2	40
1	9	4	Шаблоны	12	2	8	2	-
2	10	5	Исключения	4	2	-	2	-
1	11	6	Динамические структуры данных	54	4	8	2	40
2	13	7	Строки	4	2	-	2	-
2	14	8	STL	6	4	-	2	-
2	16	9	Лямбда-выражения	4	2	-	2	-
Всего				324	64	48	32	180

5. Содержание лекционного курса

№ темы	Всего часов	№ лекции	Тема лекции. Вопросы, отрабатываемые на лекции	Учебно-методическое обеспечение
1	2	3	4	5
1 семестр				
1	2	1	Тема 1. Что такое программирование. Цели и задачи курса. Содержание дисциплины. Рекомендуемая литература.	3,6,8
2	2	2	Тема 2. Синтаксические конструкции. Идентификаторы, ключевые слова, типы данных.	1,4,10,11,12
2	2	3	Объявление переменных, Арифметические операции,	1,4,10,11,12
2	2	4	Логические операции, операции сравнения, битовые операции, операции присваивания.	1,4,10,11,12
3	2	5	Тема 3. Типы данных. Условные операторы, преобразование типов, структура программы, основы ввода/вывода	1,4,10,11,12
4	2	6	Тема 4. Оператор варианта. Циклы, множество меток, оператор switch	1,4,10,11,12
4	2	7	Оператор for, Оператор while, Оператор do while, операторы break и continue	1,4,10,11,12
5	2	8	Тема 5. Массивы. Устройство массивов. Одномерные массивы. Многомерные массивы. Алгоритмы сортировки	1,4,10,11,12
5	2	9	Массивы, виды (статический, переменный, динамический). Массивы, принцип хранения (статический и динамический, одномерный и многомерный). Динамические массивы (стандартное выделение памяти и выделение непрерывного блока). Указатели и массивы.	1,4,10,11,12
6	2	10	Тема 6. Функции. Объявление и определение функций. Передача параметров. Возвращаемое значение. Рекурсия.	1,4,10,11,12
6	2	11	Функции и static. Указатели на функцию. Функции с переменным количеством параметров.	1,4,10,11,12
7	2	12	Тема 7. Что такое указатели. Адресная арифметика. Указатели и const	1,4,10,11,12
8	2	13	Тема 8. Память. Виды памяти (автоматические, внешние, статические и регистровые переменные). Стек и куча.	1,4,10,11,12
8	2	14	Выделение и освобождение памяти. Утечка памяти.	1,4,10,11,12
9	2	15	Тема 9. Пользовательские типы данных. Структуры. Объединения.	1,4,10,11,12
10	2	16	Тема 10. Препроцессор. Компилятор. Линковщик. Команды препроцессора. Многофайловые проекты.	2,7
2 семестр				
1	2	1	Тема 1. Объектно-ориентированное программирование. Инкапсуляция, наследование, полиморфизм	1,4,9

1	2	2	Класс как понятие. Класс и объект	1,4,10,11,12
2	2	3	Тема 2. Классы. Поля и методы. Модификаторы доступа	1,4,10,11,12
2	2	4	Конструкторы и деструкторы. Указатель this	1,4,10,11,12
2	2	5	Друзья классов. Перегрузка операций. const, static	1,4,10,11,12
3	2	6	Тема 3. Наследование классов. Модификаторы доступа	2,7
3	2	7	Виртуальные функции. Абстрактные классы	1,4,10,11,12
3	2	8	Множественное наследование Вложенные классы	1,4,10,11,12
4	2	9	Тема 4. Шаблоны	2,7
5	2	10	Тема 5. Исключения	1,4,10,11,12
6	2	11	Тема 6. Динамические структуры данных. Последовательные структуры.	1,4,10,11,12
6	2	12	Ассоциативные структуры.	1,4,10,11,12
7	2	13	Тема 7. Строки	1,4,10,11,12
8	2	14	Тема 8. STL. Контейнеры. Итераторы	1,4,10,11,12
8	2	15	Алгоритмы. Функторы	1,4,10,11,12
9	2	16	Тема 19. Лямбда-выражения	1,4,10,11,12

6. Содержание коллоквиумов

Коллоквиумы учебным планом не предусмотрены.

7. Перечень практических занятий

№ темы	Всего часов	№ занятия	Тема практического занятия. Задания, вопросы, отрабатываемые на практическом занятии	Учебно-методическое обеспечение
1	2	3	4	5
1 Семестр				
2	2	1	Синтаксические конструкции	3,6,8
3	2	2	Типы данных. Условные операторы	1,4,10,11,12
4	2	3	Оператор варианта. Циклы	1,4,10,11,12
5	2	4	Массивы	1,4,10,11,12
6	2	5	Функции	1,4,10,11,12
7	2	6	Указатели	1,4,10,11,12
8	2	7	Память	1,4,10,11,12
9	2	8	Пользовательские типы данных	1,4,10,11,12
2 Семестр				
1	2	1	Классы	1,4,9
2	2	2	Наследование	1,4,10,11,12
3	2	3	Шаблоны	2,7
4	2	4	Исключения	1,4,10,11,12
5	2	5	Динамические структуры данных	2,7
6	2	6	Строки	1,4,10,11,12
7	2	7	STL	1,4,10,11,12
8	2	8	Лямбда-выражения	1,4,10,11,12

8. Перечень лабораторных работ

№ темы	Всего часов	Наименование лабораторной работы. Задания, вопросы, отработываемые на лабораторном занятии	Учебно-методическое обеспечение
1	2	4	3
1 семестр			
2	6	Типы данных. Условные операторы. Циклы	1,4,10,11,12
5	6	Массивы	1,4,10,11,12
6	4	Функции	1,4,10,11,12
7	6	Указатели	2,7
8	6	Память	2,7
9	4	Пользовательские типы данных	2,7
2 семестр			
2	8	Классы	1,4,10,11,12
3	8	Наследование	1,4,10,11,12
4	8	Шаблоны.	1,4,10,11,12
6	8	Динамические структуры данных	2,7

8. Задания для самостоятельной работы студентов

№ темы	Всего Часов	Задания, вопросы, для самостоятельного изучения (задания)	Учебно-методическое обеспечение
1	2	3	4
1 Семестр			
7	20	Указатели	1,4,10,11,12
8	20	Память	1,4,10,11,12
9	20	Пользовательские типы данных	1,4,10,11,12
10	20	Сборка программы	1,4,10,11,12
2 Семестр			
2	20	Классы	2,7
3	40	Наследование	2,7
6	40	Динамические структуры данных	2,7

Виды, график контроля СРС, (по решению кафедры УМКС/УМКН).

№ темы	Вид СРС	Вид контроля СРС	График контроля (№ недели)
1 семестр			
1-5	Работа с печатными источниками, разбор типовых заданий	Рубежный контроль, промежуточный контроль, самоконтроль	8 (промежуточная аттестация), экзамен
6-10	Работа с печатными источниками, разбор	Рубежный контроль, промежуточный контроль,	Экзамен

	типовых заданий	самоконтроль	
2 семестр			
1-5	Работа с печатными источниками, разбор типовых заданий	Рубежный контроль, промежуточный контроль, самоконтроль	8 (промежуточная аттестация), экзамен
6-11	Работа с печатными источниками, разбор типовых заданий	Рубежный контроль, промежуточный контроль, самоконтроль	Экзамен

10. Расчетно-графическая работа

2 Семестр

Задание 1 Определение кратчайшего пути (простая сложность)

Дан файл, описывающий набор городов и карту дорог между ними. Возможный формат файла описан ниже (комментарии в реальном файле будут отсутствовать):

N // число городов

Название1 X1 Y1 // Название и координаты на карте

Название2 X2 Y2

...

НазваниеN XN YN

M // число дорог

N1 K1 Length1 // порядковые номера городов и расстояние между ними.

...

NM KM LengthM // порядковые номера городов и расстояние между ними.

Написать программу, которая для заданного файла рисует на экране карту (возможными искривлениями дорог при отображении пренебречь) и определяет кратчайший путь между любыми двумя выбранными вершинами (найденный путь должен подсвечиваться на карте). При загрузке файла программа должна проверять его на корректность (правильное число строк, длина дороги между пунктами a и b не может быть меньше фактического расстояния между их координатами и т.д.)

Задание 2. Вращение фигур (простая)

Дан файл с описанием скелета 3D фигуры. Возможный формат файла описан ниже (комментарии в реальном файле будут отсутствовать):

N // число вершин

X1 Y1 Z1 // координаты вершины

X2 Y2 Z2

...

XN YN ZN

M // число ребер

N1 K1 // порядковые номера соединяемых вершин

N2 K2

...

NM KM

Написать программу, которая читает данную фигуру из файла, показывает ее на экране и позволяет выполнять с ней простейшие манипуляции (поворот, приближение/удаление и т.д.). При загрузке файла программа должна проверять его на корректность.

Задание 3. 2D редактор (сложная, или средняя)

Написать программу, которая представляет собой простейший 2D редактор. Пользователь может создать новую фигуру-многоугольник (последовательно кликнув мышью в нескольких точках экрана), выделять фигуры, добавлять и удалять вершины. Результат работы должен сохраняться (и потом загружаться) в файл, возможный формат которого описан ниже:

```
N // число фигур
M1 // число вершин в первой фигуре
X1_1 Y1_1 // вершины фигуры
X1_2 Y1_2
...
X1_M1 Y1_M1
P1 // число ребер
N1_1 K1_1 // порядковые номера соединяемых вершин
N1_2 K1_2
...
N1_P1 K1_P1
... // аналогично для остальных фигур
```

Задание 4. Задача коммивояжера (сложная)

Дана схематичная карта кварталов города (файл, структура которого имеет примерно следующий вид):

```
00000000000000000000
02110111011101110
01000000011111120
01000000011111110
00011112011000000
01000011001121000
03011111011101110
0000000000000000000
```

Цифрами на карте обозначены:

- 0 – дорога
- 1 – здание
- 2 – магазин
- 3 – склад

Написать программу, которая по введенной из файла карте строит кратчайший маршрут, позволяющий развезти все товары со склада по магазинам и вернуться обратно.

Программа должна строить карту города и отображать на ней рассчитанный маршрут.

Задание 5. Сети, определение ключевых узлов, которые нужно разрушить для разделения сети на сегменты (сложная, или средняя)

Дан файл с описанием карты локальной компьютерной сети. Возможный вариант структуры файла приведен ниже (комментарии в реальном файле будут отсутствовать):

```
N – число вершин.
X1 Y1 // координаты вершины
```

X2 Y2

...

XN YN

M – число связей.

N1 K1 // какие вершины связываются данной связью

N2 K2

...

NM KM

Написать программу, которая рисует на экране заданную компьютерную сеть и определяет, какое минимальное число вершин нужно уничтожить, чтобы разорвать связь между компьютерами с номерами M и N (задаются пользователем).

Задание 6. Определение максимальных потоков в сети (средняя)

Дан файл с описанием карты локальной компьютерной сети. Каждая связь имеет вес – объем информации, который можно прокачать через данное соединение за единицу времени. Возможный вариант структуры файла приведен ниже (комментарии в реальном файле будут отсутствовать):

N – число вершин.

X1 Y1 // координаты вершины

X2 Y2

...

XN YN

M – число связей.

N1 K1 P1 // какие вершины связаны и какой объем данных может быть передан

N2 K2 P2

...

NM KM PM

Написать программу, которая рисует на экране заданную компьютерную сеть и определяет, с какой максимальной скоростью можно передавать между компьютерами с номерами M и N (задаются пользователем).

Задание 7. Лабиринт, поиск выхода (простая)

Дана карта лабиринта (файл, структура которого имеет примерно следующий вид):

000000000000000000

02111111011101110

01100000011111110

01001010011111110

00011111111000000

01010011001111000

01011111011101130

00000000000000000

Здесь обозначены:

0 – стена

1 – проход

2 – начальное положение игрока

3 – выход

Написать программу, которая строит на экране изображение лабиринта и определяет кратчайший маршрут от начального положения пользователя к выходу.

Задание 8. Разводка печатной платы (сложная, или средняя)

Дана карта печатной платы (файл, структура которого имеет примерно следующий вид):

```
00500000000200000
00100003000000000
00000000000200000
000000000000000100
00000003000000000
005000000000040040
00000000000000000
```

Здесь обозначены:

0 – проходимые области

1...1, 2...2 и т.д. – пары точек, которые нужно соединить

Написать программу, которая строит схему дорожек печатной платы. По возможности стараться найти кратчайшие маршруты.

Задание 9. Определение маршрута с учетом загрузки сети (сложная, или средняя).

Дан файл с описанием карты города. Каждая связь имеет два веса – максимальная пропускная способность дороги и ее текущий уровень загрузки в процентах. Возможный вариант структуры файла приведен ниже (комментарии в реальном файле будут отсутствовать):

N – число вершин.

X1 Y1 // координаты вершины

X2 Y2

...

XN YN

M – число связей.

N1 K1 P1 V1 // какие вершины связаны, пропускная способность и загрузка

N2 K2 P2 V2

...

NM KM PM VM

С учетом пробок проложить наиболее быстрый маршрут из пункта А в пункт В.

Задание 10. Построитель пейзажей (сложная, или очень сложная)

Дан файл с картой высот. По заданной карте построить скелетный пейзаж в 3D.

Обеспечить возможность облета пейзажа камерой.

Задание 11. Игра Digger (сложная, или очень сложная)

На прямоугольной карте, представляющей собой «шахту» перемещается диггер. Он может как двигаться по уже существующим ходам, так и прокапывать новые. Цель игры – собрать все разбросанные по карте сокровища. Игроку активно мешают монстры, которые также перемещаются по карте. Столкновение с монстром означает «гибель» игрока.

Монстры могут преследовать игрока.

<http://ru.wikipedia.org/wiki/Digger>

Возможные расширения: несколько уровней, бонусы с различным эффектом, разные по силе и характеристикам противники, возможность «отбиваться» от монстров.

Задание 12. Игра «Tower defense» (очень сложная)

На прямоугольной карте из «логова» к защищаемому игроком строению двигаются монстры. Игрок может противодействовать им, строя на карте защитные башни, стреляющие по монстрам. Цель игры: не допустить проникновения монстров в

защищаемое строение. Компьютер должен автоматически пересчитывать маршрут атаки на более оптимальный.

http://ru.wikipedia.org/wiki/Tower_Defense

Возможные расширения:

1. разные типы монстров,
2. upgrade башен,
3. выдача за каждого уничтоженного монстра игроку некоторого количества «золота», которое можно использовать на постройку башен и т.д.

Задание 13. Игра «Пятнашки» (средняя, или сложная)

Написать программу, которая позволяет пользователю играть в игру «пятнашки» (<http://ru.wikipedia.org/wiki/Пятнашки>). Для новой игры программа должна предлагать пользователю случайно сгенерированную заведомо корректную расстановку. **Возможное расширение:** программа сама играет в пятнашки.

Задание 14. Ремонт дорог (средняя, или сложная)

Дан файл, описывающий карту дорог в городе. Возможный формат файла описан ниже (комментарии в реальном файле будут отсутствовать):

```
N // число перекрестков
X1 Y1 // координаты перекрестка
X2 Y2
...
XN YN
M // число дорог
N1 K1 // порядковые номера соединяемых перекрестков
...
NM KM // порядковые номера соединяемых перекрестков
```

Написать программу, которая для заданного файла рисует на экране карту (возможными искривлениями дорог при отображении пренебречь). По выбору пользователя принимается решение о закрытии дороги (или дорог) на ремонт. Программа должна определять, можно ли закрыть дорогу. Дорогу закрывать нельзя, если при ее закрытии нарушится связь между любыми двумя частями города.

11. Курсовая работа

Курсовая работа учебным планом не предусмотрена.

12. Курсовой проект

Курсовой проект учебным планом не предусмотрен.

13. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)

Формирование профессиональных компетенций по дисциплине производится на лабораторных и лекционных занятиях (75%); закрепление достигается при проведении промежуточной аттестации (10%) и сдаче экзамена (15%).

Перечень компетенций с указанием этапов их формирования и критериев оценивания

№ п/п	Наименование компетенции	Части компонентов	Технологии формирования	Средства и технологии оценки
1	2	3	4	5
1	ПК-3: способность использовать языки, системы и инструментальные средства программирования в профессиональной деятельности	Знает: - Объектно-ориентированное программирование. - Основы алгоритмов	Лекции Самостоятельная работа	Экзамен
		Умеет: - проектировать классы с использованием принципов объектно-ориентированного проектирования - самостоятельно выбирать наиболее подходящие инструменты для решения типовых задач, связанных с обработкой и хранением данных	Лабораторные работы с использованием активных и интерактивных приемов обучения. Самостоятельная работа	Экзамен
		Владет: - инструментами разработки приложений	Лекции Лабораторные занятия с использованием активных и интерактивных приемов обучения. Самостоятельная работа	Экзамен
2	ПК-8: способность к освоению новых образцов программных, технических средств и информационных технологий	Знает: - принципы работы приложений на языке высокого уровня, и применять их для решения классических задач программирования	Лекции Самостоятельная работа	Экзамен
		Умеет: - создавать программы, на языке высокого уровня и соответствующие ожидаемому результату, полученному на основании анализа технического задания	Лабораторные работы с использованием активных и интерактивных приемов обучения. Самостоятельная работа	Экзамен
		Владет: - инструментами разработки приложений	Лекции Лабораторные занятия с использованием активных и интерактивных приемов обучения. Самостоятельная работа	Экзамен
3	ПК-10: способность применять современные методы исследования с использованием компьютерных технологий	Знает: - принципы работы приложений на языке высокого уровня, и применять их для решения классических задач программирования	Лекции Самостоятельная работа	Экзамен
		Умеет: - выявлять общие характеристики у разнотипных данных и обрабатывать их с применением принципов	Лабораторные работы с использованием активных и интерактивных приемов обучения. Самостоятельная	Экзамен

		объектно-ориентированного проектирования - Оценивать качество разработанного программного кода	работа	
		Владеет: - инструментами разработки приложений	Лекции Лабораторные занятия с использованием активных и интерактивных приемов обучения. Самостоятельная работа	Экзамен

При выставлении экзаменационных оценок преподаватель руководствуется следующим:

- оценки «отлично» заслуживает студент, показавший всестороннее, систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой, усвоивший основную литературу и знакомый с дополнительной литературой, рекомендованной программой, продемонстрировавший умения и навыки в рамках формируемых компетенций на высоком уровне освоения. Как правило, оценка "отлично" выставляется студентам, усвоившим взаимосвязь основных понятий дисциплины и их значение для приобретаемой профессии, проявившим творческие способности в понимании, изложении и использовании учебно-программного материала;

- оценки «хорошо» заслуживает студент, обнаруживший полное знание учебно-программного материала, успешно выполняющий предусмотренные в программе задания, усвоивший основную литературу, рекомендованную в программе, продемонстрировавший умения и навыки в рамках формируемых компетенций на хорошем уровне освоения, способный к самостоятельному пополнению знания в ходе дальнейшей учебной работы и профессиональной деятельности;

- оценки «удовлетворительно» заслуживает студент, обнаруживший знания основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, освоившийся с выполнением заданий, предусмотренных программой, знакомый с основной литературой, рекомендованной программой, продемонстрировавший умения и навыки в рамках формируемых компетенций на достаточном уровне освоения. Как правило, оценка «удовлетворительно» выставляется студенту, допустившему неточность в ответе на экзамене;

- оценка «неудовлетворительно» выставляется студенту, обнаружившему пробелы в знаниях основного учебно-программного материала, допустившего принципиальные ошибки в выполнении предусмотренных программой заданий, не освоивший умений и навыков в рамках формируемых компетенций на достаточном уровне освоения. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение или приступить к профессиональной деятельности по

окончании вуза без дополнительных занятий по соответствующей дисциплине.

Вопросы для зачета

Зачет учебным планом не предусмотрен.

Вопросы для экзамена

1 семестр

1. Идентификаторы языка. Правила именования. Ключевые слова.
2. Представление информации в двоичном виде. Таблица истинности.
3. Переменные. Базовые типы данных.
4. Переменные. Объявление, инициализация и применение.
5. Переменные. Область видимости.
6. Переменные. Ключевое слово `const`.
7. Переменные. Автоматические, внешние, статические и регистровые.
8. Условные операторы.
9. Операторы цикла.
10. Операторы безусловной передачи управления.
11. Функции. Объявление, определение и вызов функции.
12. Функции. Передача параметров по ссылке и по значению.
13. Функции. Возвращаемое значение.
14. Функции. Рекурсия.
15. Функции. Указатели как параметры функции.
16. Функции. Массивы как параметры функции.
17. Функции. Ключевые слова `const` и `static`.
18. Массивы. Устройство массивов. Объявление, инициализация и применение.
19. Массивы. Одномерные и многомерные массивы.
20. Массивы. Статические, переменные и динамические.
21. Массивы. Выделение памяти под динамический массив. Одномерный и многомерный.
22. Массивы. Связь указателей и массивов.
23. Указатель. Объявление, инициализация и применение.
24. Указатель. Взятие адреса и разыменование.
25. Указатель. Адресная арифметика.
26. Указатель. Ключевое слово `const`.
27. Память. Стек и куча.
28. Память. Выделение и освобождение памяти.
29. Память. Утечка памяти.
30. Строки. Объявление, инициализация и применение.
31. Работа с файлами.
32. Ввод/вывод. Форматированный.
33. Ввод/вывод. Файловый.
34. Структуры. Объявление, инициализация и применение.
35. Объединения. Объявление, инициализация и применение.
36. Перечисления. Объявление, инициализация и применение.
37. Сборка программы. Препроцессор, компилятор и линковщик.
38. Сборка программы. Команды препроцессора.
39. Сборка программы. Многофайловые проекты.

40. Стандартная библиотека. Состав.

2 семестр

1. ООП. Объектно-ориентированное программирование, поясните суть парадигмы, чем отличается от других. Расскажите про главные постулаты ООП.
2. Класс и объект. Что такое класс и объект, чем отличаются. Как объявляется класс и создается экземпляр класса.
3. Класс и объект. Поля и методы, параметры по умолчанию.
4. Класс и объект. Модификаторы доступа.
5. Класс и объект. Конструктор. По умолчанию, с параметрами и копирования.
6. Класс и объект. Конструктор. Список инициализаторов и вызов конструктора.
7. Класс и объект. Конструктор. Спецификаторы default и delete.
8. Класс и объект. Деструктор. Как объявляется и для чего нужен.
9. Класс и объект. Указатель this.
10. Класс и объект. Перегрузка операций.
11. Класс и объект. Статические поля и методы, особенности работы.
12. Класс и объект. Модификатор const для полей и методов, особенности работы.
13. Наследование. Базовый класс и потомок. Спецификатор final.
14. Наследование. Модификаторы доступа.
15. Наследование. Не наследуемые методы.
16. Наследование. Перекрытие методов.
17. Наследование. Конструкторы, деструкторы и перегруженные операторы
18. Наследование. Статические поля и методы, Спецификатор final
19. Множественное наследование. Прямой и косвенный базовый класс.
Ключевое слово virtual
20. Абстрактные классы. Полиморфизм.
21. Абстрактные классы. Виртуальные методы, чистые виртуальные методы и спецификатор override.
22. Абстрактные классы. Раннее и позднее связывание.
23. Шаблоны функций.
24. Шаблоны классов.
25. Исключения. Ключевые слова try, throw, catch.
26. Исключения. Передача параметров. Порядок обработки исключений
27. Лямбда-выражения.
28. Стандартная библиотека. Состав и назначение.
29. STL. Контейнеры.
30. STL. Итераторы.
31. STL. Алгоритмы.
32. STL. Функции.

Тестовые задания по дисциплине

Тестовые задания учебным планом не предусмотрен.

14. Образовательные технологии

Для реализации компетентного подхода в соответствии с требованиями ФГОС ВПО в рамках учебного курса предусмотрены активные

и интерактивные формы проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

В связи с этим предусмотрено применение мультимедийных средств и презентаций, обсуждение докладов студентов, лекции с элементами деловых игр, тестирование, консультации, решение ситуационных задач, дискуссии.

Общее количество занятий, проводимых в интерактивных формах, не менее 24 часов.

15. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

1. Биллиг В.А. Основы объектного программирования на C# (C# 3.0, Visual Studio 2008) [Электронный ресурс]/ Биллиг В.А.— Электрон. текстовые данные.— М.: БИНОМ. Лаборатория знаний, Интернет-Университет Информационных Технологий (ИНТУИТ), 2010.— 582 с.— Режим доступа: <http://www.iprbookshop.ru/16092>.— ЭБС «IPRbooks», по паролю
2. Комлева Н.В. Структуры и алгоритмы компьютерной обработки данных [Электронный ресурс]: учебное пособие/ Комлева Н.В.— Электрон. текстовые данные.— М.: Евразийский открытый институт, Московский государственный университет экономики, статистики и информатики, 2004.— 140 с.— Режим доступа: <http://www.iprbookshop.ru/10898>.— ЭБС «IPRbooks», по паролю
3. Шень А. Программирование [Электронный ресурс]: теоремы и задачи/ Шень А.— Электрон. текстовые данные.— М.: МЦНМО, 2004.— 296 с.— Режим доступа: <http://www.iprbookshop.ru/11944>.— ЭБС «IPRbooks», по паролю

Дополнительные издания

1. Сеницын С.В. Программирование на языке высокого уровня / С.В. Сеницын, А. С. Михайлов, О. И. Хлытчиев. - М. : ИЦ "Академия", 2010. - 400 с., 17 экземпляров
2. Шульга Т.Э. Объектно-ориентированное программирование на языке C# : учеб. пособие для направлений подгот. бакалавров "Программная инженерия" и "Прикладная информатика" / Т. Э. Шульга, Н. Б. Фролова ; Саратовский гос. техн. ун-т. - Саратов : СГТУ, 2014. - 108 с. : табл., 40 экземпляров
3. Могилев А.В. Информатика : учеб. пособие / А. В. Могилев, Н. И. Пак, Е. К. Хеннер ; под ред. Е. К. Хеннера. - М.: ИЦ "Академия", 2008. - 848 с., 30 экземпляров
4. Демидович, Е. М. Основы алгоритмизации и программирования. Язык Си [Текст] : учеб. пособие / Е. М. Демидович. - СПб. : БХВ-Петербург, 2006. - 440 с., 10 экземпляров
5. Сырецкий, Г. А. Информатика. Фундаментальный курс [Текст] : учебник / Г. А. Сырецкий. Основы информационной и вычислительной техники. - 2005. - 832 с., 10 экземпляров

6. Буч, Г. UML / Г. Буч, А. Якобсон, Дж . Рамбо = The unified modeling language reference manual / J. Rumbaugh, I. Jacobson, G. Booch. - 2-е изд. = second edition. - М. [и др.] : Питер, 2006. - 736 с., 10 экземпляров
7. Интернет-ресурсы
8. Справочник C#. <https://msdn.microsoft.com/ru-ru/library/6a71f45d.aspx#>
9. C# 5.0 и платформа .NET 4.5.
http://professorweb.ru/my/csharp/charp_theory/level1/infocsharp.php
10. Полное руководство по языку программирования C# 6.0 и платформе .NET 4.6. <http://metanit.com/sharp/tutorial/>

16. Материально-техническое обеспечение дисциплины.

Для проведения лекционных занятий требуется комплект технических средств обучения в составе:

- персональный компьютер (в конфигурации не хуже: процессор Intel Core 2 Duo, 2 Гбайта ОЗУ, 500 Гбайт НЖМД);
- проектор (разрешение не менее 1280x1024);
- экран для проектора.

Для проведения практических занятий требуется компьютерный класс, оборудованный ПЭВМ в конфигурации не худшей чем: процессор Pentium IV 3 ГГц, ОЗУ 2 Гбайта, НЖМД 200 Гбайт.