

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ СОЦИАЛЬНЫЙ ИНСТИТУТ»

Утверждаю
Декан факультета
Ж.В. Игнатенко
« 24 » 10 20 20 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Языки программирования

Направление подготовки: 09.03.02 Информационные системы и технологии

Направленность (профиль) программы: Информационные технологии в управлении предприятием

Квалификация выпускника: бакалавр

Форма обучения: очная, заочная

год начала подготовки – 2019

Разработана
канд.эконом.наук., доцент, доцент
А.Ю. Орлова

Согласована
зав. выпускающей кафедры
А.Ю. Орлова

Рекомендована
на заседании кафедры
от « 24 » 10 20 20 г.
протокол № 2
Зав. кафедрой Ж.В. Игнатенко

Одобрена
на заседании учебно-методической
комиссии факультета
от « 24 » 10 20 20 г.
протокол № 2
Председатель УМК Ж.В. Игнатенко

Ставрополь, 2020 г.

Содержание

1. Цели освоения дисциплины	3
2. Место дисциплины в структуре ОПОП	3
3. Планируемые результаты обучения по дисциплине.....	4
4. Объем дисциплины и виды учебной работы.....	5
5. Содержание и структура дисциплины	7
5.1. Содержание дисциплины	7
5.2. Структура дисциплины.....	7
5.3. Занятия семинарского типа	9
5.4. Курсовой проект (курсовая работа, расчетно-графическая работа, реферат, контрольная работа).....	9
5.5. Самостоятельная работа	9
6. Образовательные технологии	11
7. Фонд оценочных средств для текущего контроля успеваемости, промежуточной аттестации	11
8. Учебно-методическое и информационное обеспечение дисциплины.....	11
8.1. Основная литература	11
8.2. Дополнительная литература.....	12
8.3. Программное обеспечение	12
8.4. Профессиональные базы данных.....	12
8.5. Информационные справочные системы	12
8.6. Интернет-ресурсы	12
8.7. Методические указания по освоению дисциплины.....	12
9. Материально-техническое обеспечение дисциплины	15
10. Особенности освоения дисциплины лицами с ограниченными возможностями здоровья.....	16
Приложение 1	17

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью дисциплины «Языки программирования» является формирование профессиональных компетенций будущего бакалавра направления 09.03.02 «Информационные системы и технологии».

Задачи дисциплины:

Основными задачами освоения дисциплины являются:

- Знакомство студентов с несколькими распространенными современными языками программирования, их сравнительный анализ.
- Сравнение организации ссылочной и размерной объектной модели в различных языках.
- Изучение стандартных библиотек языков C++, Java.
- Знакомство с функциональной парадигмой программирования.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина (Б.1.В.3) «Языки программирования» входит в часть, формируемую участниками образовательных отношений, – обязательные дисциплины Блока 1 «Дисциплины (модули)» и находится в логической и содержательно-методической связи с другими дисциплинами.

Предшествующие дисциплины (курсы, модули, практики)	Последующие дисциплины (курсы, модули, практики)
Информатика и программирование	Технологии программирования
Операционные системы	Визуальное программирование
Информационные системы и технологии	Методы и средства проектирования информационных систем и технологий

Требования к «входным» знаниям, умениям и навыкам обучающегося, необходимым при освоении данной дисциплины

Знать:

- различные подходы к определению понятия «информация»;
- методы измерения количества информации: вероятностный и алфавитный; единицы измерения информации;
- назначение наиболее распространенных средств автоматизации информационной деятельности (текстовых редакторов, текстовых процессоров, графических редакторов, электронных таблиц, баз данных, компьютерных сетей);
- назначение и виды информационных моделей, описывающих реальные объекты или процессы;
- использование алгоритма как способа автоматизации деятельности;
- назначение и функции операционных систем;
- назначение, роль, принципы построения, задачи и классификацию современных информационных систем (ИС) и информационных технологий, в том числе ИС и технологий отечественного производства для решения задач в профессиональной деятельности.

Уметь:

- оценивать достоверность информации, сопоставляя различные источники;
- распознавать информационные процессы в различных системах;
- использовать готовые информационные модели, оценивать их соответствие реальному объекту и целям моделирования;
- осуществлять выбор способа представления информации в соответствии с поставленной задачей;

- иллюстрировать учебные работы с использованием средств информационных технологий;
- создавать информационные объекты сложной структуры, в том числе гипертекстовые;
- просматривать, создавать, редактировать, сохранять записи в базах данных;
- осуществлять поиск информации в базах данных, компьютерных сетях и пр.;
- представлять числовую информацию различными способами (таблица, массив, график, диаграмма и пр.);
- анализировать, оценивать и работать с аппаратным и программным обеспечением современных ИС на основе современных информационных технологий, в том числе отечественного производства при решении задач в профессиональной деятельности.

Владеть:

- стандартными приемами написания на алгоритмическом языке программы для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ;
- компьютерными средствами представления и анализа данных;
- навыками анализа и работы с аппаратным и программным обеспечением установленным на АРМ пользователя ИС.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Формируемые компетенции (код компетенции, уровень)	Планируемые результаты обучения на данном этапе формирования компетенции
ПК-3 Способен анализировать требования к программному обеспечению, разрабатывать технические спецификации на программные компоненты и их взаимодействие	Знать: принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения, типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения, методы и средства проектирования программного обеспечения, баз данных и программных интерфейсов.
	Уметь: использовать существующие типовые решения и шаблоны проектирования программного обеспечения, применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.
	Владеть навыками: проведения анализа исполнения требований к ПО, выработки вариантов реализации требований к ПО, оценки и обоснования рекомендуемых решений по ПО, выбора средств и вариантов реализации требований к программному обеспечению.
ПК-4 Способен проектировать программное обеспечение	Знать: правила настройки и эксплуатации устанавливаемого системного программного обеспечения, включая лицензионные требования; стандарты информационного взаимодействия систем; регламенты обеспечения информационной безопасности.
	Уметь: идентифицировать инциденты при работе системного программного обеспечения; применять

	<p>специализированные программно-аппаратные средства для локализации инцидентов при работе системного программного обеспечения; выполнять настройку системного программного обеспечения в соответствии с регламентами обеспечения информационной безопасности; проводить авторизацию пользователей, имеющих доступ к настройке системного программного обеспечения инфокоммуникационной системы организации; применять программные и программно-аппаратные средства защиты информации.</p> <p>Владеть навыками разработки, изменения и согласования архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения, проектирования структуры данных, базы данных и программного интерфейса, оценивания и согласования сроков выполнения задач проектирования.</p>
ПК-7 Способен организовывать и выполнять проектирование и дизайн ИС, разрабатывать базы данных ИС	<p>Знает: языки программирования и работы с базами данных, инструменты и методы проектирования и дизайна ИС, верификации структуры программного кода, инструменты и методы проектирования и верификации структуры баз данных.</p>
	<p>Умеет: кодировать на языках программирования, верифицировать структуру программного кода, разрабатывать и верифицировать структуру баз данных.</p>
	<p>Владеет навыками: разработки и верификации структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС, разработки структуры баз данных ИС в соответствии с архитектурной спецификацией, верификации структуры баз данных ИС относительно архитектуры ИС и требований заказчика к ИС, устранения обнаруженных несоответствий ИС требованиям.</p>

4. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Общий объем дисциплины составляет 4 зачетных единиц, 144 академических часа.

Очная форма обучения.

Вид учебной работы	Всего часов	Триместр
		4
Контактная работа (всего)	40	40
в том числе:		
1) занятия лекционного типа (ЛК)	10	10
из них		
– лекции	10	10
2) занятия семинарского типа (ПЗ)	30	30

из них		
– семинары (С)	-	-
– практические занятия (ПР)	30	30
– лабораторные работы (ЛР)	-	-
3) групповые консультации	-	-
4) индивидуальная работа	-	-
5) промежуточная аттестация	-	-
Самостоятельная работа (всего) (СР)	104	104
в том числе:		
Курсовой проект (работа)	-	-
Расчетно-графические работы	-	-
Контрольная работа	-	-
Реферат	-	-
Самоподготовка (самостоятельное изучение разделов, проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам и т.д.)	104	104
Подготовка к аттестации	-	-
Общий объем, час	144	144
Форма промежуточной аттестации		Диф.зачет

Заочная форма обучения.

Вид учебной работы	Всего часов	Триместр
		4
Контактная работа (всего)	10,3	10,3
в том числе:		
1) занятия лекционного типа (ЛК)	4	4
из них		
– лекции	4	4
2) занятия семинарского типа (ПЗ)	6	6
из них		
– семинары (С)	-	-
– практические занятия (ПР)	6	6
– лабораторные работы (ЛР)	-	-
3) групповые консультации	-	-
4) индивидуальная работа	-	-
5) промежуточная аттестация	0,3	0,3
Самостоятельная работа (всего) (СР)	133,7	133,7
в том числе:		
Курсовой проект (работа)	-	-
Расчетно-графические работы	-	-
Контрольная работа	-	-
Реферат	-	-
Самоподготовка (самостоятельное изучение разделов, проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам и т.д.)	130	130

Подготовка к аттестации	3,7	3,7
Общий объем, час	144	144
Форма промежуточной аттестации		Диф.зачет

5. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

5.1. Содержание дисциплины

№ раздела (темы)	Наименование раздела (темы)	Содержание раздела (темы)
1.	Основные устройства ЭВМ и их назначение. История развития языков.	Основные устройства ЭВМ и их назначение. История развития языков.
2.	Общие принципы построения языков программирования. Препроцессор и макрообработка. Этапы решения задач на компьютере.	Общие принципы построения языков программирования. Препроцессор и макрообработка. Этапы решения задач на компьютере.
3.	Современные интегрированные среды. Встроенный отладчик. Библиотека программ и классов	Современные интегрированные среды. Встроенный отладчик. Библиотека программ и классов
4.	Простейшая программа. Вывод текста на экран. Директивы clrscr() и getch()	Простейшая программа. Вывод текста на экран. Препроцессор. Директивы clrscr() и getch()
5.	Память. Переменные. Вывод на экран. Запись в переменные типа int и float. Ввод с клавиатуры	Память. Переменные. Вывод на экран. Запись в переменные типа int и float. Ввод с клавиатуры.
6.	Арифметические операции. Математические выражения и функции	Арифметические операции. Математические выражения и функции.
7.	Операции сравнения и логические операции. Условные операторы	Операции сравнения и логические операции. Условные операторы
8.	Циклы	Цикл for. Цикл while. Цикл do – while.
9.	Массивы. Некоторые простейшие задачи. Матрицы	Массивы. Некоторые простейшие задачи. Матрицы
10.	Указатели. Связь указателя с массивами. Динамическая память	Указатели. Связь указателя с массивами. Динамическая память

5.2. Структура дисциплины

Очная форма обучения

№ раздела (темы)	Наименование раздела (темы)	Количество часов			
		Всего	Л	ПЗ (ПР)	СР

1.	Основные устройства ЭВМ и их назначение. История развития языков.	13	1	2	10
2.	Общие принципы построения языков программирования. Препроцессор и макрообработка. Этапы решения задач на компьютере.	15	1	4	10
3.	Современные интегрированные среды. Встроенный отладчик. Библиотека программ и классов	15	1	2	12
4.	Простейшая программа. Вывод текста на экран. Директивы clrscr() и getch()	15	1	4	10
5.	Память. Переменные. Вывод на экран. Запись в переменные типа int и float. Ввод с клавиатуры	13	1	2	10
6.	Арифметические операции. Математические выражения и функции	15	1	4	10
7.	Операции сравнения и логические операции. Условные операторы	14	1	2	10
8.	Циклы	15	1	4	10
9.	Массивы. Некоторые простейшие задачи. Матрицы	15	1	2	12
10.	Указатели. Связь указателя с массивами. Динамическая память	15	1	4	10
	Групповая консультация	-	-	-	-
	Промежуточная аттестация	-	-	-	-
	Общий объем:	144	10	30	104

Заочная форма обучения

№ раздела (темы)	Наименование раздела (темы)	Количество часов			
		Всего	Л	ПЗ (ПР)	СР
1.	Основные устройства ЭВМ и их назначение. История развития языков.	15	-	1	14
2.	Общие принципы построения языков программирования. Препроцессор и макрообработка. Этапы решения задач на компьютере.	13	1	-	12
3.	Современные интегрированные среды. Встроенный отладчик. Библиотека программ и классов	15	-	1	14
4.	Простейшая программа. Вывод текста на экран. Директивы clrscr() и getch()	13	-	1	12
5.	Память. Переменные. Вывод на экран. Запись в переменные типа int и float. Ввод с клавиатуры	13	1	-	12
6.	Арифметические операции. Математические выражения и функции	15	-	1	14
7.	Операции сравнения и логические операции. Условные операторы	13	1	-	12
8.	Циклы	15	-	1	14
9.	Массивы. Некоторые простейшие задачи. Матрицы	13	1	-	12

10.	Указатели. Связь указателя с массивами. Динамическая память	15	-	1	14
	Групповая консультация	-	-	-	-
	Промежуточная аттестация	4	-	-	-
	Общий объем:	144	4	6	130

5.3. Занятия семинарского типа

Очная форма обучения

№ п/п	№ раздела (темы)	Вид занятия	Наименование	Количество часов
1.	1.	ПР	Введение в языки программирования. Общие принципы построения и использования языков программирования.	2
2.	2.	ПР	Условные операторы: if и switch	4
3.	3.	ПР	Операторы цикла for и while	2
4.	4.	ПР	Программирование с использованием массивов	4
5.	5.	ПР	Функции	2
6.	6.	ПР	Указатели	4
7.	7.	ПР	Запись и чтение данных файла	2
8.	8.	ПР	Введение в объектно-ориентированное программирование (ООП)	4
9.	9.	ПР	Знакомство с принципами компонентно-ориентированного программирования (КОП)	2
10.	10.	ПР	Знакомство с принципами компонентно-ориентированного программирования (КОП)	4

Заочная форма обучения

№ п/п	№ раздела (темы)	Вид занятия	Наименование	Количество часов
1.	1.	ПР	Введение в языки программирования. Общие принципы построения и использования языков программирования.	1
2.	3.	ПР	Условные операторы: if и switch	1
3.	4.	ПР	Операторы цикла for и while	1
4.	6.	ПР	Программирование с использованием массивов	1
5.	8.	ПР	Функции	1
6.	10.	ПР	Указатели	1

5.4. Курсовой проект (курсовая работа, расчетно-графическая работа, реферат, контрольная работа)

Не предусмотрено

5.5. Самостоятельная работа

Очная форма обучения

№ раздела (темы)	Виды самостоятельной работы	Количество часов
1.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	10
2.	Подготовка к лекциям и практическим занятиям. Выполнение творческого задания.	10
3.	Подготовка к лекциям и практическим занятиям. Выполнение заданий, указанных в методических рекомендациях.	12
4.	Подготовка к лекциям и практическим занятиям. Выполнение заданий, указанных в методических рекомендациях.	10
5.	Подготовка к лекциям и практическим занятиям.	10
6.	Подготовка к лекциям и практическим занятиям. Выполнение заданий, указанных в методических рекомендациях.	10
7.	Подготовка к лекциям и практическим занятиям. Выполнение творческого задания.	10
8.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	10
9.	Подготовка к лекциям и практическим занятиям. Выполнение заданий, указанных в методических рекомендациях.	12
10.	Подготовка к лекциям и практическим занятиям. Выполнение заданий, указанных в методических рекомендациях.	10

Заочная форма обучения

№ раздела (темы)	Виды самостоятельной работы	Количество часов
1.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	14
2.	Подготовка к лекциям и практическим занятиям. Выполнение творческого задания.	12
3.	Подготовка к лекциям и практическим занятиям. Выполнение заданий, указанных в методических рекомендациях.	14
4.	Подготовка к лекциям и практическим занятиям. Выполнение заданий, указанных в методических рекомендациях.	12
5.	Подготовка к лекциям и практическим занятиям.	12
6.	Подготовка к лекциям и практическим занятиям. Выполнение заданий, указанных в методических рекомендациях.	14
7.	Подготовка к лекциям и практическим занятиям. Выполнение творческого задания.	12
8.	Подготовка к лекциям и практическим занятиям. Изучение источников информации по дисциплине.	14

9.	Подготовка к лекциям и практическим занятиям. Выполнение заданий, указанных в методических рекомендациях.	12
10.	Подготовка к лекциям и практическим занятиям. Выполнение заданий, указанных в методических рекомендациях.	14

6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Информационные технологии, используемые при осуществлении образовательного процесса по дисциплине:

- сбор, хранение, систематизация, обработка и представление учебной и научной информации;
- обработка различного рода информации с применением современных информационных технологий;
- самостоятельный поиск дополнительного учебного и научного материала, с использованием поисковых систем и сайтов сети Интернет, электронных энциклопедий и баз данных;
- использование электронной почты для рассылки и асинхронного общения, чата преподавателей и обучающихся, переписки и обсуждения возникших учебных проблем для синхронного взаимодействия
- дистанционные образовательные технологии (при необходимости).

Интерактивные и активные образовательные технологии

№ раздела (темы)	Вид занятия (ЛК, ПР, С, ЛР)	Используемые интерактивные и активные образовательные технологии	Количество часов ОФО/ЗФО
2.	ПР.	Проблемное обучение	2/1
3.	Л.	Интерактивная лекция	2/1
4.	ПР.	Опережающая самостоятельная работа	2/1
5.	ПР.	Проблемное обучение	2/1

7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Фонд оценочных средств (оценочные материалы) для текущего контроля успеваемости, промежуточной аттестации по дисциплине приводятся в приложении.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1. Основная литература

1. Огнева, М. В. Программирование на языке С++: практический курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2020. — 335 с. — (Высшее образование). — ISBN 978-5-534-05123-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/454165> (дата обращения: 18.08.2020).
2. Трофимов, В. В. Алгоритмизация и программирование : учебник для вузов / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2020. — 137 с. — (Высшее образование). — ISBN 978-5-534-

07834-3. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/452333> (дата обращения: 18.08.2020).

3. Тузовский, А. Ф. Объектно-ориентированное программирование : учебное пособие для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2020. — 206 с. — (Высшее образование). — ISBN 978-5-534-00849-4. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/451429> (дата обращения: 18.08.2020).

8.2. Дополнительная литература

1. Подбельский, В. В. Программирование. Базовый курс C# : учебник для вузов / В. В. Подбельский. — Москва : Издательство Юрайт, 2020. — 369 с. — (Высшее образование). — ISBN 978-5-534-10616-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/450868> (дата обращения: 18.08.2020).

2. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2020. — 320 с. — (Высшее образование). — ISBN 978-5-534-02444-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/450832> (дата обращения: 18.08.2020).

8.3 Программное обеспечение

1. MSOffice,
2. C++,
3. VisualStudio.

8.4 Профессиональные базы данных

1. Электронная библиотечная система «СКСИ» [Электронный ресурс] – Режим доступа: <https://www.sksi.ru/environment/ebs/1363/>

2. Интернет университет информационных технологий [Электронный ресурс] – Режим доступа: <http://www.intuit.ru/>

3. Электронная библиотечная система «Urait» [Электронный ресурс] – Режим доступа: <https://urait.ru/>

4. Электронная библиотека информационных технологий CITForum.ru [Электронный ресурс] – Режим доступа: (<http://citforum.ru>).

5. Виртуальная академия Microsoft [Электронный ресурс] – Режим доступа (<http://aka.ms/studentcourse>).

8.5. Информационные справочные системы

1. Информационно-справочная система для программистов [Электронный ресурс] – Режим доступа: <http://life-prog.ru>

8.6. Интернет-ресурсы

1. Академия ORACLE [Электронный ресурс] – Режим доступа: <https://academy.oracle.com/ru/>

2. Научная сеть Scipeople [Электронный ресурс] – Режим доступа: <http://scipeople.ru/>

3. Портал открытых данных [Электронный ресурс] – Режим доступа <https://data.gov.ru/>.

8.7. Методические указания по освоению дисциплины.

Методические указания при работе над конспектом во время проведения лекции

В ходе лекционных занятий необходимо вести конспектирование учебного материала. Общие и утвердившиеся в практике правила и приемы конспектирования лекций:

Конспектирование лекций ведется в специально отведенной для этого тетради, каждый лист которой должен иметь поля, на которых делаются пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

Необходимо записывать тему и план лекций, рекомендуемую литературу к теме. Записи разделов лекции должны иметь заголовки, подзаголовки, красные строки. Для выделения разделов, выводов, определений, основных идей можно использовать цветные карандаши и фломастеры.

Названные в лекции ссылки на первоисточники надо пометить на полях, чтобы при самостоятельной работе найти и вписать их.

В конспекте дословно записываются определения понятий, категорий и законов. Остальное должно быть записано своими словами.

Каждому обучающемуся необходимо выработать и использовать допустимые сокращения наиболее распространенных терминов и понятий.

В конспект следует заносить всё, что преподаватель пишет на доске, а также рекомендуемые схемы, таблицы, диаграммы и т.д.

Методические указания по подготовке к практическим работам

Целью практических и лабораторных работ является углубление и закрепление теоретических знаний, полученных обучающимися на лекциях и в процессе самостоятельного изучения учебного материала, а, следовательно, формирование у них определенных умений и навыков.

В ходе подготовки к практическим и лабораторным работам необходимо прочитать конспект лекции, изучить основную литературу, ознакомиться с дополнительной литературой, выполнить выданные преподавателем задания. При этом учесть рекомендации преподавателя и требования программы. Дорабатывать свой конспект лекции, делая в нем соответствующие записи из литературы. Желательно при подготовке к практическим и лабораторным работам по дисциплине одновременно использовать несколько источников, раскрывающих заданные вопросы.

Методические указания по организации самостоятельной работы

Самостоятельная работа приводит обучающегося к получению нового знания, упорядочению и углублению имеющихся знаний, формированию у него профессиональных навыков и умений.

Самостоятельная работа выполняет ряд функций:

- развивающую;
- информационно-обучающую;
- ориентирующую и стимулирующую;
- воспитывающую;
- исследовательскую.

Виды самостоятельной работы, выполняемые в рамках курса:

1. Проработка и повторение лекционного материала
2. Подготовка к практическим занятиям
3. Подготовка к лабораторным занятиям
4. Реферат
5. Подготовка к аттестации

Обучающимся рекомендуется с самого начала освоения курса работать с литературой и предлагаемыми заданиями в форме подготовки к очередному аудиторному занятию. При этом актуализируются имеющиеся знания, а также

создается база для усвоения нового материала, возникают вопросы, ответы на которые обучающийся получает в аудитории.

Можно отметить, что некоторые задания для самостоятельной работы по курсу имеют определенную специфику. При освоении курса обучающийся может пользоваться библиотекой вуза, которая в полной мере обеспечена соответствующей литературой. Значительную помощь в подготовке к очередному занятию может оказать имеющийся в учебно-методическом комплексе краткий конспект лекций. Он же может использоваться и для закрепления полученного в аудитории материала.

Методические указания по работе с литературой

Всю литературу можно разделить на учебники и учебные пособия, оригинальные научные монографические источники, научные публикации в периодической печати. Из них можно выделить литературу основную (рекомендуемую), дополнительную и литературу для углубленного изучения дисциплины.

Изучение дисциплины следует начинать с учебника, поскольку учебник – это книга, в которой изложены основы научных знаний по определенному предмету в соответствии с целями и задачами обучения, установленными программой.

При работе с литературой следует учитывать, что имеются различные виды чтения, и каждый из них используется на определенных этапах освоения материала.

Предварительное чтение направлено на выявление в тексте незнакомых терминов и поиск их значения в справочной литературе. В частности, при чтении указанной литературы необходимо подробнейшим образом анализировать понятия.

Сквозное чтение предполагает прочтение материала от начала до конца. Сквозное чтение литературы из приведенного списка дает возможность обучающемуся сформировать свод основных понятий из изучаемой области и свободно владеть ими.

Выборочное – наоборот, имеет целью поиск и отбор материала. В рамках данного курса выборочное чтение, как способ освоения содержания курса, должно использоваться при подготовке к лабораторным практикумам по соответствующим разделам.

Аналитическое чтение – это критический разбор текста с последующим его конспектированием. Освоение указанных понятий будет наиболее эффективным в том случае, если при чтении текстов обучающийся будет задавать к этим текстам вопросы. Часть из этих вопросов сформулирована в приведенном в ФОС перечне вопросов для собеседования. Перечень этих вопросов ограничен, поэтому важно не только содержание вопросов, но сам принцип освоения литературы с помощью вопросов к текстам.

Целью изучающего чтения является глубокое и всестороннее понимание учебной информации.

Есть несколько приемов изучающего чтения:

1. Чтение по алгоритму предполагает разбиение информации на блоки: название; автор; источник; основная идея текста; фактический материал; анализ текста путем сопоставления имеющихся точек зрения по рассматриваемым вопросам; новизна.

2. Прием постановки вопросов к тексту имеет следующий алгоритм:

- медленно прочитать текст, стараясь понять смысл изложенного;
- выделить ключевые слова в тексте;
- постараться понять основные идеи, подтекст и общий замысел автора.

3. Прием тезирования заключается в формулировании тезисов в виде положений, утверждений, выводов.

К этому можно добавить и иные приемы: прием реферирования, прием комментирования.

Важной составляющей любого солидного научного издания является список литературы, на которую ссылается автор. При возникновении интереса к какой-то обсуждаемой в тексте проблеме всегда есть возможность обратиться к списку относящейся к ней литературы. В этом случае вся проблема как бы разбивается на составляющие части, каждая из которых может изучаться отдельно от других. При этом важно не терять из вида общий контекст и не погружаться чрезмерно в детали, потому что таким образом можно не увидеть главного.

Методические указания к разработке и проведению проблемного обучения.

Под проблемным обучением понимается такая организация учебного процесса, которая предполагает создание под руководством учителя проблемных ситуаций и активную самостоятельную деятельность учащихся по их разрешению, в результате чего и происходит творческое овладение предметными знаниями, умениями, навыками (ЗУН) и развитие творческих способностей.

Данный вид обучения:

1. направлен на самостоятельный поиск учащимися новых понятий и способов действий;
2. предполагает последовательное и целенаправленное выдвижение перед учащимися познавательных проблем, разрешение которых (под руководством учителя) приводит к активному усвоению новых знаний;
3. обеспечивает особый способ мышления, прочность знаний и творческое их применение в практической деятельности.

При проблемном обучении преподаватель не сообщает готовых знаний, а организует учащихся на их поиск: понятия, закономерности, теории познаются в ходе поиска, наблюдений, анализа фактов, мыслительной деятельности.

Необходимыми составляющими проблемного обучения являются следующие понятия: «проблема», «проблемная ситуация», «гипотеза», «эксперимент».

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для реализации дисциплины требуется следующее материально-техническое обеспечение (специальные помещения):

- для проведения занятий лекционного типа
учебная аудитория, оснащенная учебной мебелью, оборудованная проектором, ПК, экраном, доской.
- для проведения занятий семинарского типа, практических занятий
учебная аудитория, оснащенная учебной мебелью, оборудованная проектором, ПК, экраном, доской.
- для проведения, текущего контроля и промежуточной аттестации
учебная аудитория, оснащенная учебной мебелью, оборудованная проектором, ПК, экраном, доской.
- для групповых и индивидуальных консультаций
учебная аудитория, оснащенная учебной мебелью, оборудованная проектором, ПК, экраном, доской.
- для самостоятельной работы:
помещение, оснащенное компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду Института

10. ОСОБЕННОСТИ ОСВОЕНИЯ ДИСЦИПЛИНЫ ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники, учебные пособия и дидактические материалы, специальные технические средства обучения коллективного и индивидуального пользования, услуги ассистента (тьютора), оказывающего обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.

Освоение дисциплины обучающимися с ограниченными возможностями здоровья может быть организовано совместно с другими обучающимися, а также в отдельных группах.

Освоение дисциплины обучающимися с ограниченными возможностями здоровья осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья.

В целях доступности получения высшего образования по образовательной программе лицами с ограниченными возможностями здоровья при освоении дисциплины обеспечивается:

1) для лиц с ограниченными возможностями здоровья по зрению:

– присутствие тьютора, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе, записывая под диктовку),

– письменные задания, а также инструкции о порядке их выполнения оформляются увеличенным шрифтом,

– специальные учебники, учебные пособия и дидактические материалы (имеющие крупный шрифт или аудиофайлы),

– индивидуальное равномерное освещение не менее 300 люкс,

– при необходимости студенту для выполнения задания предоставляется увеличивающее устройство;

2) для лиц с ограниченными возможностями здоровья по слуху:

– присутствие ассистента, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе, записывая под диктовку),

– обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающемуся предоставляется звукоусиливающая аппаратура индивидуального пользования;

– обеспечивается надлежащими звуковыми средствами воспроизведения информации;

3) для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата:

– письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются тьютору;

– по желанию студента задания могут выполняться в устной форме.

Приложение 1

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ (ОЦЕНОЧНЫЕ МАТЕРИАЛЫ) ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

По дисциплине «Языки программирования»

1. Показатели, критерии оценки освоения дисциплины

Результаты обучения	Показатели оценивания	Критерии оценивания	Процедуры оценивания
ПК-3 Способен анализировать требования к программному обеспечению, разрабатывать технические спецификации на программные компоненты и их взаимодействие			
Знает принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения, типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения, методы и средства проектирования программного обеспечения, баз данных и программных интерфейсов.	Демонстрация знаний принципов построения архитектуры программного обеспечения и видов архитектуры программного обеспечения, типовых решений, библиотеки программных модулей, шаблонов, классов объектов, используемых при разработке программного обеспечения, методов и средства проектирования программного обеспечения, баз данных и программных интерфейсов.	Полнота знаний принципов построения архитектуры программного обеспечения и видов архитектуры программного обеспечения, типовых решений, библиотеки программных модулей, шаблонов, классов объектов, используемых при разработке программного обеспечения, методов и средства проектирования программного обеспечения, баз данных и программных интерфейсов.	устный опрос
Умеет использовать существующие типовые решения и шаблоны проектирования программного обеспечения, применять	Демонстрация умений в области использования существующих типовых решений и шаблонов	Методическая грамотность использования умений в области использования существующих	Практические задания

методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.	проектирования программного обеспечения, применения методов и средств проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.	типовых решений и шаблонов проектирования программного обеспечения, применения методов и средств проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.	
Владеет навыками: проведения анализа исполнения требований к ПО, выработки вариантов реализации требований к ПО, оценки и обоснования рекомендуемых решений по ПО, выбора средств и вариантов реализации требований к программному обеспечению.	Демонстрирует владение навыками проведения анализа исполнения требований к ПО, выработки вариантов реализации требований к ПО, оценки и обоснования рекомендуемых решений по ПО, выбора средств и вариантов реализации требований к программному обеспечению.	Адекватность выбора методов проведения анализа исполнения требований к ПО, выработки вариантов реализации требований к ПО, оценки и обоснования рекомендуемых решений по ПО, выбора средств и вариантов реализации требований к программному обеспечению.	Практические задания
ПК-4 Способен проектировать программное обеспечение			
Знает правила настройки и эксплуатации устанавливаемого системного программного обеспечения, включая лицензионные требования; стандарты информационного	Демонстрация знаний правил настройки и эксплуатации устанавливаемого системного программного обеспечения, включая лицензионные	Полнота знаний по правилам настройки и эксплуатации устанавливаемого системного программного обеспечения, включая лицензионные требования; стандарты	устный опрос

взаимодействия систем; регламенты обеспечения информационной безопасности.	требования; стандарты информационного взаимодействия систем; регламенты обеспечения информационной безопасности.	информационного взаимодействия систем; регламенты обеспечения информационной безопасности.	
Умеет идентифицировать инциденты при работе системного программного обеспечения; применять специализированные программно-аппаратные средства для локализации инцидентов при работе системного программного обеспечения; выполнять настройку системного программного обеспечения в соответствии с регламентами обеспечения информационной безопасности; проводить авторизацию пользователей, имеющих доступ к настройке системного программного обеспечения инфокоммуникационной системы организации; применять программные и программно-аппаратные средства защиты информации.	Демонстрация умений в области идентификации инцидентов при работе системного программного обеспечения; применения специализированных программно-аппаратных средств для локализации инцидентов при работе системного программного обеспечения.	Методическая грамотность использования умений в области идентификации инцидентов при работе системного программного обеспечения; применения специализированных программно-аппаратных средств для локализации инцидентов при работе системного программного обеспечения.	Практические задания
Владеет навыками разработки, изменения и	Демонстрирует владение навыками	Адекватность выбора методов	Практические задания

<p>согласования архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения, проектирования структуры данных, базы данных и программного интерфейса, оценивания и согласования сроков выполнения задач проектирования.</p>	<p>разработки, изменения и согласования архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения, проектирования структуры данных, базы данных и программного интерфейса, оценивания и согласования сроков выполнения задач проектирования.</p>	<p>разработки, изменения и согласования архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения, проектирования структуры данных, базы данных и программного интерфейса, оценивания и согласования сроков выполнения задач проектирования.</p>	
<p>ПК-7 Способен организовывать и выполнять проектирование и дизайн ИС, разрабатывать базы данных ИС</p>			
<p>Знает языки программирования и работы с базами данных, инструменты и методы проектирования и дизайна ИС, верификации структуры программного кода, инструменты и методы проектирования и верификации структуры баз данных.</p>	<p>Демонстрация знаний языков программирования и работы с базами данных, инструменты и методы проектирования и дизайна ИС, верификации структуры программного кода, инструменты и методы проектирования и верификации структуры баз данных.</p>	<p>Полнота знаний языков программирования и работы с базами данных, инструменты и методы проектирования и дизайна ИС, верификации структуры программного кода, инструменты и методы проектирования и верификации структуры баз данных.</p>	<p>устный опрос,</p>
<p>Умеет кодировать на языках программирования, верифицировать</p>	<p>Демонстрация умений кодировать на языках программирования,</p>	<p>Методическая грамотность использования умений кодировать</p>	<p>Практические задания</p>

структуру программного кода, разрабатывать и верифицировать структуру баз данных.	верифицировать структуру программного кода, разрабатывать и верифицировать структуру баз данных.	на языках программирования, верифицировать структуру программного кода, разрабатывать и верифицировать структуру баз данных.	
Владеет навыками: разработки и верификации структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС, разработки структуры баз данных ИС в соответствии с архитектурной спецификацией, верификации структуры баз данных ИС относительно архитектуры ИС и требований заказчика к ИС, устранения обнаруженных несоответствий ИС требованиям.	Демонстрирует владение навыками разработки и верификации структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС, разработки структуры баз данных ИС в соответствии с архитектурной спецификацией, верификации структуры баз данных ИС относительно архитектуры ИС и требований заказчика к ИС, устранения обнаруженных несоответствий ИС требованиям.	Адекватность выбора методов разработки и верификации структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС, разработки структуры баз данных ИС в соответствии с архитектурной спецификацией, верификации структуры баз данных ИС относительно архитектуры ИС и требований заказчика к ИС, устранения обнаруженных несоответствий ИС требованиям.	Практические задания
ПК-3 ПК-4 ПК-7			Промежуточная аттестация: дифференцированный зачет

2. Методические материалы, определяющие процедуры оценивания

2.1. Методические материалы, определяющие процедуры оценивания в рамках текущего контроля успеваемости

Устные опросы проводятся во время лекций, практических занятий и возможны при проведении промежуточной аттестации в качестве дополнительного испытания при недостаточности результатов тестирования. Основные вопросы для устного опроса доводятся до сведения студентов на предыдущем занятии.

Количество вопросов определяется преподавателем.

Время проведения опроса от 10 минут до 1 академического часа.

Устные опросы строятся так, чтобы вовлечь в тему обсуждения максимальное количество обучающихся в группе, проводить параллели с уже пройденным учебным материалом данной дисциплины и смежными курсами, находить удачные примеры из современной действительности, что увеличивает эффективность усвоения материала на ассоциациях.

Критерии и шкала оценки устного опроса

Развернутый ответ студента должен представлять собой связное, логически последовательное сообщение на заданную тему, показывать его умение применять определения, правила в конкретных случаях.

«отлично» ставится, если:

- 1) студент полно излагает материал, дает правильное определение основных понятий;
- 2) обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только из учебника, но и самостоятельно составленные;
- 3) излагает материал последовательно и правильно с точки зрения норм литературного языка.

«хорошо» - студент дает ответ, удовлетворяющий тем же требованиям, что и для «отлично», но допускает 1–2 ошибки, которые сам же исправляет, и 1–2 недочета в последовательности и языковом оформлении излагаемого.

«удовлетворительно» – студент обнаруживает знание и понимание основных положений данной темы, но:

- 1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил;
- 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры;
- 3) излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.

«неудовлетворительно» ставится, если студент обнаруживает незнание большей части соответствующего вопроса, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал. Оценка «2» отмечает такие недостатки в подготовке, которые являются серьезным препятствием к успешному овладению последующим материалом.

Практические задания выполняются студентами на практических занятиях. Студентам необходимо выполнить практические задания, указанные преподавателем. Результаты работы сохранить в файлах. После выполнения заданий необходимо преподавателю продемонстрировать результаты работы и быть готовым ответить на вопросы и продемонстрировать выполнение отдельных пунктов заданий. Защита выполненных практических заданий осуществляется на практическом занятии.

Критерии и шкала оценки практических заданий

«отлично» ставится, если: студент самостоятельно и правильно решил учебно-профессиональную задачу, уверенно, логично, последовательно и аргументировано излагал свое решение, используя изученные понятия.

«хорошо» ставится, если: студент самостоятельно и в основном правильно решил учебно-профессиональную задачу, уверенно, логично, последовательно и аргументировано излагал свое решение, используя изученные понятия.

«удовлетворительно» ставится, если: студент в основном решил учебно-профессиональную задачу, допустил несущественные ошибки, слабо аргументировал свое решение, используя в основном изученные понятия.

«неудовлетворительно» ставится, если: студент не решил учебно-профессиональную задачу.

2.2. Методические материалы, определяющие процедуры оценивания в рамках промежуточной аттестации

Промежуточная аттестация по дисциплине проводится в форме дифференцированного зачета.

Дифференцированный зачет служит формой проверки успешного выполнения студентами практических заданий и усвоения знаний на занятиях лекционного и семинарского типа.

Результаты дифференцированного зачета определяются на основании результатов текущего контроля успеваемости студента в течение периода обучения.

Критерии и шкала оценки дифференцированного зачета

Дифференцированный зачет выставляется по совокупному результату текущего контроля успеваемости студента по дисциплине:

«отлично» - выставляется при условии, если студент выполнил все практические занятия с оценкой не ниже «хорошо и отлично», результаты тестирования и устного опроса в совокупности также оценены с оценкой не ниже «хорошо и отлично» при этом совокупный средний балл оценок не ниже 4,6.

«хорошо» - выставляется при условии, если студент выполнил все практические занятия с оценкой не ниже «хорошо, отлично, удовлетворительно», результаты тестирования и устного опроса в совокупности также оценены с оценкой не ниже «хорошо, отлично, удовлетворительно», при этом совокупный средний балл оценок не ниже 3,8.

«удовлетворительно» - выставляется при условии, если студент выполнил все практические занятия с оценкой не ниже «удовлетворительно», результаты тестирования и устного опроса в совокупности также оценены с оценкой не ниже «удовлетворительно», при этом совокупный средний балл оценок не ниже 3,8.

«неудовлетворительно» - не выполнены условия для получения оценки «удовлетворительно».

3. Типовые контрольные задания

Типовые задания для текущего контроля успеваемости

3.1 Перечень типовых вопросов для устного опроса

1. Основные устройства ЭВМ и их назначение.
2. История развития языков.

3. Общие принципы построения языков программирования.
4. Препроцессор и макрообработка.
5. Этапы решения задач на компьютере.
6. Современные интегрированные среды.
7. Встроенный отладчик.
8. Библиотека программ и классов
9. Простейшая программа.
10. Вывод текста на экран.
11. Препроцессор.
12. Директивы clrscr() и getch()
13. Память.
14. Переменные.
15. Вывод на экран.
16. Запись в переменные типа int и float.
17. Ввод с клавиатуры.
18. Арифметические операции.
19. Математические выражения и функции.
20. Операции сравнения и логические операции.
21. Условные операторы
22. Цикл for.
23. Цикл while.
24. Цикл do – while.
25. Массивы.
26. Некоторые простейшие задачи.
27. Матрицы
28. Указатели.
29. Связь указателя с массивами.
30. Динамическая память

3.2. Типовые практические задания

Цель работы: ознакомиться со средой разработки Visual Studio.

Теоретическая часть

Программные задачи в Visual Studio .Net оформляются в виде проектов; обычно для каждой отдельной программы создается свой проект. Проект (project) представляет собой набор файлов, которые совместно используются для создания одной программы.

В этой и последующих лабораторных работах используем шаблон консольного приложения, в котором уже имеет необходимые ссылки проекта и файлы. В частности, имеются ссылки на следующие пространства имен .NET Framework:

- o Система, в которой содержатся основные и базовые классы для часто используемых значений и ссылочных типов данных, события и обработчики событий, интерфейсы, атрибуты и обработке исключений.

- o DLL сборки – Mscorlib, поддерживающее разработку платформы .NET Framework.

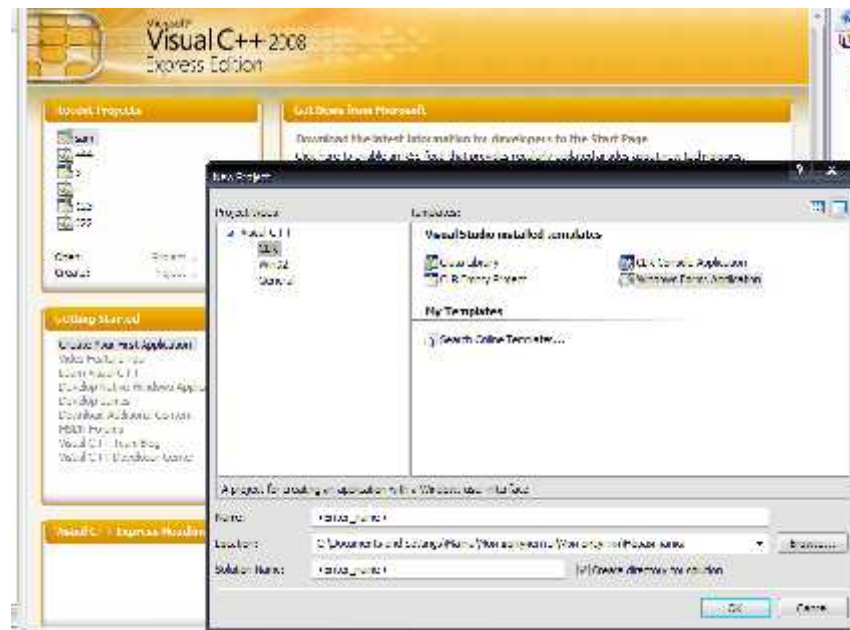


Рис.1. Создание нового проекта

А также здесь помещаются исходные файлы:

- o Консоль (сpp-файл) – основной файл источника. Он указывает на DLL-файл проекта и пространство имен проекта.

- o Stdafx.h – заголовочный файл для построения и предварительно компилируются с именем файла StdAfx.obj типов.

Создание проекта приложения консоли (CLR) среды CLR

1. В строке меню «Файл» выберите пункты «Создать», «Проект».
2. В диалоговом окне «Создать проект» в поле Visual C++ выберите узел CLR, а затем выберите шаблон консольного приложения.
3. В поле «Имя» введите имя приложения.
4. Нажмите «ОК».

Таблица 1. Типы данных C++

Имя типа	Байты	Диапазон значений
int (целочисленное значение)	4	от -2 147 483 648 до 2 147 483 647
unsigned int	4	от 0 до 4 294 967 295
bool	1	false или true
char	1	от -128 до 127
signed char	1	от -128 до 127
unsigned char	1	от 0 до 255
short	2	от -32 768 до 32 767
unsigned short	2	от 0 до 65 535
long	4	от -2 147 483 648 до 2 147 483 647
unsigned long	4	от 0 до 4 294 967 295
long long	8	от -9 223 372 036 854 775 808 до
unsigned long long	8	от 0 до 18 446 744 073 709 551
float	4	3,4E +/- 38 (7 знаков)

double	8	1,7E +/- 308 (15 знаков)
long double	8	1,7E +/- 308 (15 знаков)

Для реализации ввод/вывода данных в консоль в этой и последующих лабораторных работах используются потоки ввод / вывода cin и cout. Для их использования необходимо добавить в коде используемого шаблона библиотеку iostream и прописать пространство имен std.

Пространство имен — это декларативная область, в рамках которой определяются различные идентификаторы (имена типов, функций, переменных, и т. д.). Все типы и функции стандартной библиотеки C++ объявлены в пространстве имен std.

Компиляторы Visual C++ поддерживают типы, указанные таблице 1.

Для объявления переменных можно использовать следующую схему:

[тип данных] [имя переменной];

или вариант с инициализацией

[тип данных] [имя переменной] = [инициализирующее выражение];

Оператор присваивания «=» используется

- 1) для инициализации переменных;
- 2) для изменения значения переменной путем копирования;
- 3) для получения адреса переменной.

Примеры

a= 0;

a -= 10; // вычитание из переменной значения 10

a /= 15; // деление переменной на 15

a = b+c; // сложение переменных

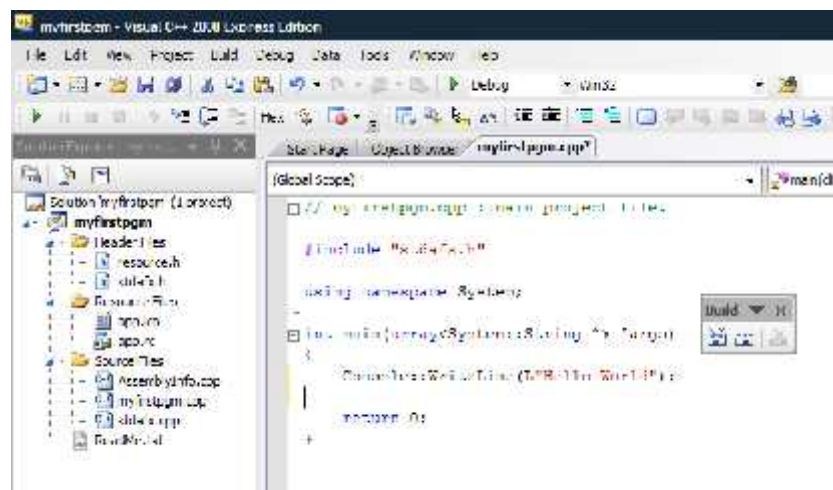


Рис. 2. Структуры программы-шаблона CLR

Оборудование и материалы

Персональные ПК

Microsoft Visual Studio 2008 / 2010 / 2012

Указания по технике безопасности

Перед началом работы необходимо:

- убедиться в отсутствии видимых повреждений на рабочем месте;
- разместить на столе тетради, учебные пособия так, что бы они не мешали работе на компьютере;

- принять правильную рабочую позу;
- посмотреть на индикатор монитора и системного блока и определить, включён или выключен компьютер. Переместите мышь, если компьютер находится в энергосберегающем состоянии

или включить монитор, если он был выключен.

При работе в компьютерном классе категорически запрещается:

- находиться в классе в верхней одежде;
- класть одежду и сумки на столы;
- находиться в классе с напитками и едой;
- располагаться сбоку или сзади от включенного монитора;
- присоединять или отсоединять кабели, трогать разъемы, провода и розетки;
- передвигать компьютеры и мониторы;
- открывать системный блок;
- включать и выключать компьютеры самостоятельно.
- пытаться самостоятельно устранять неисправности в работе аппаратуры;
- перекрывать вентиляционные отверстия на системном блоке и мониторе;
- ударять по клавиатуре, нажимать бесцельно на клавиши;
- класть книги, тетради и другие вещи на клавиатуру, монитор и системный блок;
- удалять и перемещать чужие файлы;
- приносить и запускать компьютерные игры.

Задания

Задания базового уровня

Задание 1. Запустите Visual Studio .Net путем выбора кнопок

Пуск ->Все программы -> Microsoft Visual Studio 2008 ->MS Visual

Studio 2008. На экране появится окно стартовой страницы с именем MS Visual Studio .Net.

Меню стартовой страницы содержит следующие пункты:

File, Edit, View, Tools, Window, Help

Задание 2. Раскройте каждый пункт меню, ознакомьтесь с содержанием и запишите содержание основных из них.

Окно стартовой страницы также содержит две кнопки New Project и Open Project, позволяющие создать новый или открыть существующий проект.

Задание 3. Создание проекта приложения

На стартовой странице выберите гиперссылку Create: Project

или в главном меню Visual Studio .NET выбираем File -> New -> Project.

При этом раскрывается окно (рисунок 1), в котором можно

выбрать тип проекта (например Visual C++), шаблон CLR проекта (выберите, например, CLR Console Application). Далее необходимо ввести имя (Name) проекта – my_first_pgm – и указать через кнопку Browse путь для сохранения проекта в вашей папке. В окне New Projects щелкаем кнопку ОК.

Задание 4. Дополнение кода приложения

Для того, чтобы приложение выполняло какие-либо действия необходимо дополнить его код. Текст кода в нашем случае будем писать до строки Console::WriteLine(L"Hello World"); в пределах между фигурными скобками, но перед функцией return 0.

Прежде, чем осуществить запуск программы следует ее отладить и откомпилировать. Для этого выбираем пункт меню

Build/Build Project. В нижнем окне появляется результат об ошибках компиляции. Далее, если ошибок нет, запускаем программу на исполнение.

Задание 5. Изучение принципа использования идентификатора cout для работы со стандартным потоком ввода/вывода. Поток – это некоторая абстракция, отражающая перемещение данных от источника к приемнику. Операция << называется операцией вставки. Она копирует содержимое переменной, стоящей в правой части, в объект, содержащийся в левой ее части. Операцию « cout << » используют для вывода текстовых и числовых данных в консоль при запуске программы.

Пример: cout << “В нашей группе 25 человек\n”

// \n – символ новой строки

Для работы с потоками ввода вывода следует подключить библиотеку iostream, для чего в начале программы добавьте строку: #include <iostream>, добавьте в заголовках using namespace std.

Задание 6. Напишите программу, используя горизонтальную табуляцию (знак \t) и переходы на новую строку (знак \n), выво-

дящую следующую таблицу (при выводе цифр кавычки в коде программы не ставятся):

```
1990  135
1991  7290
```

Выполните это задание, используя манипулятор setw.

Задание 7. Изучение принципа работы с переменными. Переменные в программе предназначены для возможности выделения памяти для временного хранения данных, с которыми производятся операции. Переменные целого типа объявляются как int (например, int var1; // var1 – это наша первая переменная) в пределах фигурных скобок в первую очередь, так как программа читает код сверху вниз, и эти переменные будут использоваться далее в программе.

Задание 8. Постройте таблицу, в которой для типов переменных: int, char, string, float, double, long double укажите объем выделяемой памяти при объявлении переменной такого типа, возможные максимальные и минимальные значения переменной, а также ее назначение.

Задание 9. Объявить две переменные, инициализировать первую, используя оператор присваивания, вывести в консоль результат умножения первой переменной на 3.

Задание 10. Изучение принципа использования идентификатора `cin` для работы со стандартным потоком ввода/вывода. Операцию « `cin >>` » используют для запроса от пользователя данных, вводимых с клавиатуры.

Пример:

```
int n; // объявление переменной целочисленного типа
cout << "Сколько человек у вас в группе ?\n";
cin << n;
cout << "У нас " << n << " человек в группе \n";
```

Задание 11. Задайте три переменные `var1`, `var2`, `var3`. Первые две – целого типа (`int`), третью – действительного типа (`float`). Запросите у пользователя значения `var1` и `var2`, разделите одно значение на другое, выведите результат – `var3 =` , используя операцию преобразования типов.

Задание 12. Используя манипуляторы `dec`, `oct`, `hex`, представьте число 15 в десятичной, восьмеричной, шестнадцатеричной системах исчисления.

Задания повышенного уровня

Задание 1. Решить задачу. Две дроби можно сложить следующим образом: $a/b+c/d=(a*d+b*c)/b*d$. Пользователь вводит значения

двух дробей, эти две дроби складываются, а программа выдает результат в форме дроби. Пример взаимодействия программы с пользователем:

```
«Введите первую дробь:» 1/2,
«введите вторую дробь:» 2/5,
«сумма равна 9/10».
```

Замечание. Операция извлечения `>>` может считывать более одного значения за раз: `cin >> a >> dummychar >> b. dummychar` – некоторая символьная переменная.

Задание 2. Решить задачу. Подключить библиотеку с математическими функциями. Используя конструкцию по типу `Math::??`

Из открывающего списка выпишите перечень математических функций, доступных для использования. Напишите программу, в которой у пользователя запрашивается число с плавающей точкой, далее это число возводим в степень 3 и вычитаем экспоненту этого числа. Результат выводим в консоль.

Задание 3*. Решить задачу. Вычислить значение функции $y(x)=x0.2 - 11x0.5\sin(0.008x)$ при условии, что x – случайное число.

Задание 4*. Используя функцию `atoi(ch)`, осуществить перевод десятичного числа, заданного пользователем в двоичную.

Здесь `ch` – символьная переменная.

Содержание отчета

Для получения зачета по лабораторной работе необходимо:

- 1) Предоставить в электронном виде все исходные коды программ по заданиям базового уровня
- 2) Конспект ответов на контрольные вопросы
- 3) Условия и решения задач профильного уровня при необходимости, дополненные блок-схемами.

Контрольные вопросы

1. Основные элементы языков программирования.
2. Типы данных: целочисленные, числа с плавающей точкой, строковые и символьные типы данных.
3. Оператор присваивания.
4. Ввод – вывод данных для консольных приложений. Поток: cin, cout.
5. Как создать приложение?
6. Как отладить программу?
7. Как откомпилировать программу?
8. Как запустить программу на исполнение?
9. Как вывести текст в консоль на русском?
10. Какие типы данных вам известны?
11. Как осуществить ввод данных с клавиатуры?
12. Какие последовательные действия необходимо сделать, чтобы запросить у пользователя ввести последовательно с клавиатуры переменные следующего типа: строковую – Фамилия, символьные – Инициалы, целочисленные – возраст, число с плавающей точкой – вес в кг?

Типовые задания для промежуточной аттестации

3.3 Типовые контрольные вопросы для устного опроса на дифференцированном зачете

1. Этапы решения задач на компьютере.
2. Современные интегрированные среды.
3. Встроенный отладчик.
4. Запись в переменные типа int и float.
5. Ввод с клавиатуры.
6. Арифметические операции.
7. Математические выражения и функции.
8. Операции сравнения и логические операции.
9. Условные операторы
10. Цикл for.
11. Цикл while.
12. Вывод текста на экран.
13. Препроцессор.
14. Директивы clrscr() и getch()
15. Память.
16. Переменные.
17. Основные устройства ЭВМ и их назначение.
18. История развития языков.
19. Общие принципы построения языков программирования.
20. Препроцессор и макрообработка.
21. Матрицы
22. Указатели.
23. Связь указателя с массивами.

24. Динамическая память