


Утверждено приказом ГОУ ВПО ДонГУУ от 23.08.2017г. №675

ДОНЕЦКАЯ НАРОДНАЯ РЕСПУБЛИКА  
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«ДОНЕЦКАЯ АКАДЕМИЯ УПРАВЛЕНИЯ И ГОСУДАРСТВЕННОЙ СЛУЖБЫ  
ПРИ ГЛАВЕ ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ»

ФАКУЛЬТЕТ ГОСУДАРСТВЕННОЙ СЛУЖБЫ И УПРАВЛЕНИЯ  
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

УТВЕРЖДАЮ

Проректор по учебной работе

  
Л.Н.Костина

20.06.2017 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

**«Информатика и программирование»**

Направление подготовки

09.03.03 «Прикладная информатика»

Донецк  
2017

Рабочая программа учебной дисциплины «Информатика и программирование» для студентов 1-2 курса образовательного уровня «бакалавр» направления подготовки 09.03.03 «Прикладная информатика» очной и заочной форм обучения.

Автор,  
разработчик: старший преподаватель, Э.М. Лебезова

Программа рассмотрена на  
заседании ПМК кафедры

«Прикладная информатика»

Протокол заседания ПМК от

08.06.2017

№ 10

Председатель ПМК



А. Н. Верзилов

Программа рассмотрена на  
заседании кафедры

Информационных технологий

Протокол заседания кафедры от

09.06.2017

№ 13

Заведующая кафедрой



Н. В. Брадул

## 1. Цель освоения дисциплины и планируемые результаты обучения по дисциплине (соотнесенные с планируемыми результатами освоения образовательной программы)

Целями изучения дисциплины являются формирование у будущих специалистов общего представления о фундаментальных понятиях, используемых во всех курсах Computer Science, практических навыков по основам алгоритмизации вычислительных процессов и программированию решения экономических, вычислительных и других задач, развитие умения работы с персональным компьютером на высоком пользовательском уровне, обучение работе с научно-технической литературой и технической документацией по программному обеспечению ПЭВМ.

В результате освоения дисциплины студент должен:

### 1) Знать:

- основные понятия информатики;
- изобразительные средства описания алгоритмов;
- основные приемы алгоритмизации и программирования на языке высокого уровня;
- основные типы алгоритмов и их использование для решения вычислительных, инженерных, экономических и других типов прикладных задач;
- основные структуры данных, способы их представления и обработки;
- системы программирования на алгоритмических языках высокого уровня (C++, C#, Java);
- принципы разработки программ;
- основные методы автономной и комплексной отладки и тестирования простых программ;
- технологический процесс подготовки и решения задач на ПЭВМ.

### 2) Уметь:

- разрабатывать алгоритмы решения и программировать задачи обработки данных в предметной области;
- разрабатывать проект тестирования программы, выполнять тестирование и отладку программ;
- оформлять программную документацию.

### 3) Владеть

- культурой мышления, способностью к обобщению, анализу, восприятию информации, постановке цели и выбору путей ее достижения;
- способностью к саморазвитию, повышению своей квалификации и мастерства;
- способностью работать с информацией в глобальных компьютерных сетях;
- пониманием сущности и значение информации в развитии современного информационного общества, сознавать опасности и угрозы, возникающие в этом процессе, соблюдать основные требования информационной безопасности, в том числе защиты государственной тайны;
- способностью использовать основные законы естественнонаучных дисциплин в профессиональной деятельности и эксплуатировать современное электронное оборудование и информационно-коммуникационные технологии в соответствии с целями образовательной программы бакалавра.

Компетенции обучающегося, формируемые в результате освоения дисциплины:

Код соответствующей компетенции по ГОС	Наименование компетенций	Результат освоения (знать, уметь, владеть)
ОПК-1	способность использовать нормативно-правовые	<b>Знать:</b> - нормативные правовые документы

	документы, международные и отечественные стандарты в области информационных систем и технологий	в области информационных систем и технологий. <b>Уметь:</b> - использовать нормативные правовые документы в области информационных систем и технологий для организации защиты информации. <b>Владеть:</b> - международными и отечественными стандартами в области разработки информационных систем и технологий.
ОПК-3	способность использовать основные законы естественнонаучных дисциплин и современные информационно-коммуникационные технологии в профессиональной деятельности	<b>Знать:</b> - основные законы естественнонаучных дисциплин применительно к информатике и программированию - основные современные информационно-коммуникационные технологии в информатике и программировании; <b>Уметь:</b> - применять методы разработки алгоритмов и программ на основе законов естественнонаучных дисциплин <b>Владеть:</b> - навыками использования современных информационно-коммуникационных технологий
ОПК-4	способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	<b>Знать:</b> - методы разработки информационно-коммуникационных технологий; - виды и источники угроз безопасности информации для различных профессиональных областей; - основы законодательной базы в сфере информационной безопасности; - основные требования информационной безопасности. <b>Уметь:</b> - применять методы разработки программ и алгоритмов; - определять актуальные источники угроз безопасности для различных профессиональных областей <b>Владеть:</b> - навыками владения современных

		средств информационной безопасности.
ПК-7	способность эксплуатировать и сопровождать ИС и сервисы	<p><b>Знать:</b></p> <ul style="list-style-type: none"> <li>- принципы эксплуатации и сопровождения информационных систем и сервисов с точки зрения предметной области.</li> </ul> <p><b>Уметь:</b></p> <ul style="list-style-type: none"> <li>- эксплуатировать и сопровождать информационные системы и сервисы с точки зрения их эксплуатации.</li> </ul> <p><b>Владеть:</b></p> <ul style="list-style-type: none"> <li>- принципами эксплуатации и сопровождения информационных систем и сервисов с точки зрения их эффективного применения.</li> </ul>

## 2. Место дисциплины в структуре основной образовательной программы

Дисциплина «Информатика и программирование» относится к математическому и естественно-научному циклу дисциплин и входит в базовую часть этого цикла.

Содержательно она закладывает основы знаний для освоения многих дисциплин профессионального цикла.

### 2.1. Требования к предварительной подготовке обучающегося

- Математика (школьный курс);
- Информатика (школьный курс).

### 2.2. Дисциплины и/или практики, для которых освоение данной дисциплины необходимо как предшествующее:

- Базы данных
- Программная инженерия
- Интернет-программирование
- Разработка информационных систем

## 3. Объем дисциплины в кредитах (зачетных единицах) с указанием количества академических часов, выделенных на аудиторную (по видам учебных занятий) и самостоятельную работу студента

	Зачетные единицы (кредиты ECTS)	Всего часов	Форма обучения			
			Очная			
			Семестр			
			№1	№2	№3	№4
<b>Общая трудоемкость</b>	<b>13</b>	<b>468</b>	<b>Количество часов на вид работы:</b>			
<b>Виды учебной работы, из них:</b>						
<b>Аудиторные занятия (всего)</b>		<b>210</b>	<b>54</b>	<b>54</b>	<b>54</b>	<b>48</b>
В том числе:						
Лекции		<b>70</b>	<b>18</b>	<b>18</b>	<b>18</b>	<b>16</b>
Практические занятия		<b>140</b>	<b>36</b>	<b>36</b>	<b>36</b>	<b>32</b>
<b>Самостоятельная работа (всего)</b>		<b>258</b>	<b>72</b>	<b>72</b>	<b>72</b>	<b>42</b>
<b>Промежуточная аттестация</b>						

В том числе:						
зачет /экзамен		дифзач	экзамен	дифзач	экзамен	
	Зачетные единицы (кредиты ECTS)	Всего часов	Форма обучения			
			Заочная Семестр			
			№1	№2	№3	№4
<b>Общая трудоемкость</b>	<b>13</b>	<b>468</b>	<b>Количество часов на вид работы:</b>			
<b>Виды учебной работы, из них:</b>						
<b>Аудиторные занятия (всего)</b>	<b>40</b>	<b>8</b>	<b>8</b>	<b>12</b>	<b>12</b>	
В том числе:						
Лекции	<b>12</b>	<b>2</b>	<b>2</b>	<b>4</b>	<b>4</b>	
Практические занятия	<b>28</b>	<b>6</b>	<b>6</b>	<b>8</b>	<b>8</b>	
<b>Самостоятельная работа (всего)</b>	<b>428</b>	<b>119</b>	<b>117</b>	<b>114</b>	<b>78</b>	
<b>Промежуточная аттестация</b>						
В том числе:						
зачет /экзамен		дифзач	экзамен	дифзач	экзамен	

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

##### 4.1. Разделы (темы) дисциплины с указанием отведенного на них количества академических часов и видов учебных занятий

Наименование раздела, темы дисциплины	Виды учебной работы (бюджет времени)									
	Очная форма обучения					Заочная форма обучения				
	Лекции	Практические занятия	Семинарские занятия	Самостоятельная работа	Всего	Лекции	Практические занятия	Семинарские занятия	Самостоятельная работа	Всего
1	2	3	4	5	6	7	8	9	10	11
<b>Раздел № 1. Информация и программирование.</b>										
Тема 1.1. Представление информации в ЭВМ. Системы исчисления.	3	6		10	19				18	18
Тема 1.2. Состав языка программирования С++	3	6		10	19	1	2		18	21
Тема 1.3. Базовые типы данных языка С++	3	6		10	19		1		18	19
<b>Итого по разделу:</b>	<b>9</b>	<b>18</b>		<b>30</b>	<b>57</b>	<b>1</b>	<b>3</b>		<b>54</b>	<b>58</b>
<b>Раздел № 2. Алгоритмизация и программирование на языках высокого уровня.</b>										
Тема 2.1. Алгоритмы и	3	6		12	21	1	2		18	21

Наименование раздела, темы дисциплины	Виды учебной работы (бюджет времени)									
	Очная форма обучения					Заочная форма обучения				
	Лекции	Практические занятия	Семинарские занятия	Самостоятель ная работа	Всего	Лекции	Практические занятия	Семинарские занятия	Самостоятель ная работа	Всего
программы										
Тема 2.2. Реализация базовых структур алгоритма в языках программирования.	3	6		12	21				19	18
Тема 2.3. Указатели и массивы.	3	6		12	21		1		20	19
<b>Итого по разделу:</b>	9	18		36	63	1	3		57	58
<b>Всего за семестр:</b>	18	36		66	120	2	6		119	127
<b>Раздел № 3. Структурное программирование на C++.</b>										
Тема 3.1. Парадигмы программирования	3	6		10	19				18	18
Тема 3.2. Реализация процедурного подхода с помощью функций	3	6		10	19	1	2		18	21
Тема 3.3. Создание больших приложений и специализированн ых библиотек функций.	3	4		10	17		1		18	19
<b>Итого по разделу:</b>	9	16		30	55	1	3		54	58
<b>Раздел № 4. Структуры данных и файлов</b>										
Тема 4.1. Методология проектирования программных продуктов. Способы конструирования программ на основе структур.	3	6		12	21	1	2		20	23
Тема 4.2.	3	6		12	21				22	18
Тема 4.3. Файлы последовательного и прямого доступа.	3	6		10	19		1		21	19
<b>Итого по разделу:</b>	9	18		34	61	1	3		63	60
<b>Всего за семестр:</b>	18	34		64	116	2	6		117	125
<b>Раздел № 5 Объектно-ориентированное программирование на C#</b>										
Тема 5.1. Программные системы,	3	6		10	19				18	18

Наименование раздела, темы дисциплины	Виды учебной работы (бюджет времени)									
	Очная форма обучения					Заочная форма обучения				
	Лекции	Практические занятия	Семинарские занятия	Самостоятельная работа	Всего	Лекции	Практические занятия	Семинарские занятия	Самостоятельная работа	Всего
предназначенные для написания объектно-ориентированных программ										
Тема 5.2. Введение в объектно-ориентированное программирование	3	6		10	19	2	2		18	22
Тема 5.3. Построение программ на основе фундаментальных принципов ООП.	3	6		10	19		2		18	20
<b>Итого по разделу:</b>	9	18		30	57	2	4		54	60
<b>Раздел № 6. Технологии создания приложений на платформе .NET</b>										
Тема 6.1. Обзор технологий платформы .NET.	3	6		12	21	2	2		20	24
Тема 6.2. Обработка событий и исключительных ситуаций в C#.	3	6		12	21				20	20
Тема 6.3. Создание многопоточных приложений	3	6		12	21		2		20	22
<b>Итого по разделу:</b>	9	18		36	63	2	4		60	66
<b>Всего за семестр:</b>	18	36		66	120	4	8		114	126
<b>Раздел № 7. Создание GUI с помощью MS VisualStudio</b>										
Тема 7.1. Базовые элементы построения десктопных приложений Windows Form	2	6		10	18	1			18	19
Тема 7.2. Создание собственных визуальных элементов.	2	6		10	18	1	2		18	21
Тема 7.3. Графические возможности в C#	3	4		10	16		1		18	19
<b>Итого по разделу:</b>	7	16		30	53	2	3		54	59
<b>Раздел № 8. Разработка приложений для БД</b>										
Тема 8.1.	3	6		12	21	2	2		8	12



Наименование раздела, темы дисциплины	Виды учебной работы (бюджет времени)									
	Очная форма обучения					Заочная форма обучения				
	Лекции	Практические занятия	Семинарские занятия	Самостоятельная работа	Всего	Лекции	Практические занятия	Семинарские занятия	Самостоятельная работа	Всего
Технологии построения приложений для БД в среде MS Visual Studio.										
Тема 8.2. Разработка программ для работы с базами данных MS SQL и My SQL	3	6		10	19		2		8	10
Тема 8.3. Разработка программ для работы с нереляционными базами данных	3	6		10	19		1		8	10
<b>Итого по разделу:</b>	9	18		32	59	2	5		24	31
<b>Всего за семестр:</b>	16	34		62	112	4	8		78	90

#### 4.2. Содержание разделов дисциплины:

Наименование раздела, темы дисциплины	Содержание разделов дисциплины	Содержание семинарских/практических занятий		
		Часы	Кол-во часов	
			0	3
1	2	3	4	5
<b>СЕМЕСТР 1</b>				
<b>Раздел 1. Информация и программирование.</b>				
Т 1.1	Основные положения теории информации. Понятие информации. Единицы измерения, хранения и передачи информации: бит, байт, машинное слово. Формы представления числовой, символьной, графической, звуковой информации в ЭВМ. Представление чисел в разных системах счисления (двоичной, десятичной, восьмеричной, шестнадцатеричной). Связь двоичной, восьмеричной и шестнадцатеричной систем счисления.	<b>Практическое занятие №1-2</b>	6	
		1. Системы классификации и кодирования информации. Кодирование и декодирование.	2	
		2. Представление чисел в различных системах счисления. Перевод чисел из одной системы в другую.	4	

Т 1.2	Программа как последовательность действий компьютера. Исходная и объектная программа. Трансляция как процесс преобразования исходного кода в объектный. Революция персональных компьютеров. Программа как совокупность операторов описания и исполняемых операторов. Алфавит языка. Понятия: лексема, комментарий, идентификатор, служебное слово и их реализация в языке C++.	<b>Практическое занятие №3-4</b>	6	2
		3. Структура программы C++. Функция main, пространство имён, включения.	2	1
		4. Операторы ввода-вывода C++. Решение задач в C++.	4	1
Т 1.3	Представление отрицательных чисел. Дополнительный код. Знаковая и беззнаковая формы представления в C++. Представление символьной и текстовой информации. Представление чисел с плавающей запятой. Базовые типы данных языка C++. Явное и неявное преобразование в языке C++.	<b>Практическое занятие №5-6</b>	6	1
		5. Работа с типами. Явное и неявное преобразование типов.	4	1
		6. Арифметические и логические операции.	2	
<b>Раздел 2. Алгоритмизация и программирование на языках высокого уровня.</b>				
Т 2.1	Определение алгоритма. Свойства алгоритма по Д. Кнуту. Отличие понятия алгоритм от понятий рецепт, метод, процедура. Предметная область алгоритма, набор операций, интерпретатор. Формы представления алгоритма - блок схема, псевдокод. Базовые структуры (следование, ветвление, цикл).	<b>Практическое занятие №7-8</b>	6	2
		7. Управляющие структуры и операторы ветвления в C++ (if, switch)	2	1
		8. Решение задач в C++ с использованием базовых структур.	4	1
Т 2.2	Программа как реализация понятия алгоритма в среде обработки данных. Этапы подготовки и решения задач на компьютере. Компоненты языка программирования - типы данных и переменных, операции, логика алгоритма, модульная организация программы. Реализация базовых структур алгоритма в языках программирования. Циклы с постусловием и предусловием. Оператор ветвления.	<b>Практическое занятие №9-10</b>	6	
		9. Циклы с предусловием и постусловием	2	
		10. Счётные циклы и циклы обхода коллекций.	4	
Т 2.3	Переменная как объект языка программирования. Адресация объектов в программе. Указатель, обобщенный указатель, адресная арифметика, тип указателя. Классы ошибок, возникающих при работе с указателями. Одномерные и многомерные массивы. Взаимосвязь между массивами и указателями. Разность указателей. Представление текстовой информации в ЭВМ.	<b>Практическое занятие №11-12</b>	6	1
		11. Использование массивов при решении задач на C++. Операции с массивами. Ввод и вывод массивов.	2	1
		12. Алгоритмы поиска и упорядочения массива.	4	
<b>СЕМЕСТР 2</b>				
<b>Раздел № 3. Структурное программирование на C++.</b>				
Т 3.1	Основы традиционной технологии программирования. Модульное программирование, нисходящее и пошаговое проектирование. Структурное программирование. Объектно-ориентированное программирование. Способы конструирования программ с помощью	<b>Практическое занятие №13-14</b>	6	
		13. Знакомство с функциями. Выполнение алгоритма с помощью стандартных и собственных функций.	2	

	принципа «Разделяй и властвуй».	14.Функции для работы со строкой.	4	
Т 3.2	Функции и их назначение. Прототип функции, объявление и вызов. Инлайн функции. Способы передачи параметров. Передача параметров по ссылке и значению.	<b>Практическое занятие №15-16</b>	6	2
		15.Модульная структура программы.	2	2
		16.Функции поиска и упорядочения массива в C++.	4	
Т 3.3	Программирование рекурсивных алгоритмов. Перегрузка функций. Шаблоны функций. Создание больших приложений и специализированных библиотек функций.	<b>Практическое занятие №17-18</b>	4	1
		17.Создание рекурсивных алгоритмов с помощью функций.	2	1
		18.Создание собственной библиотеки и подключение её к проекту.	2	
<b>Раздел 4. Структуры данных и файлов на C++.</b>				
Т 4.1	Жизненный цикл программного обеспечения. Анализ проблемной области. Системный анализ и подготовка технического задания (спецификаций) на разработку комплекса программ. Постановка задачи и спецификация программных продуктов. Методология проектирования программных продуктов. Способы конструирования программ.	<b>Практическое занятие №19</b> 19.Создание стека для вычислений математических выражений.	6	2
Т 4.2	Стандартные типы данных и типы данных, определяемые пользователем. Списки: основные виды и способы реализации. Понятие кучи, списка, очереди и стека. Статические и динамические контейнеры. Стандартные контейнеры данных List, Map, Set.	<b>Практическое занятие №20-21</b>	6	
		20.Динамические структуры данных и списки: основные виды и способы реализации.	4	
		21.Моделирование очереди с помощью однонаправленных списков.	2	
Т 4.3	Файлы последовательного и прямого доступа. Основы доказательства правильности. Технологии обработки и отладки программ. Критерии качества программы. О-нотации.	<b>Практическое занятие №22-23</b>	6	1
		22.Типы данных, определяемые пользователем; записи; файлы.	4	1
		23.Оценка эффективности алгоритмов сортировки данных.	2	
<b>СЕМЕСТР 3</b>				
<b>Раздел № 5. Объектно-ориентированное программирование на C#.</b>				
Т 5.1	Программные системы, предназначенные для написания объектно-ориентированных программ, основные особенности и их	<b>Практическое занятие №24-25</b>	6	2

	предназначение. Среда разработки VisualStudio, история развития, основные особенности. Средства объектно-ориентированного программирования. Ввод и отладка программ в среде программирования языка высокого уровня. Преимущества объектно-ориентированного подхода.	24. Создание объектов существующих классов. Использование полей, методов, свойств, конструкторов.	2	
		25. Классы и объекты. Создание собственных классов.	4	
Т 5.2	Введение в объектно-ориентированное программирование. Жизненный цикл программного обеспечения, место ООП в нем. Построение и использование классов. Инкапсуляция. Разделение полей и методов класса. Методы доступа private, protected и public при создании класса в C#.	<b>Практическое занятие №26-27</b>	6	
		26.Динамическое создание и удаление объектов. Использование стандартных классов.	2	
		27.Создание иерархии классов путём наследования	4	2
Т 5.3	Принципы ООП. Наследование, правила наследования, использование наследования при написании программ на C#. Полиморфизм. Преобразование типов. Использование полиморфизма при построении иерархии классов на языке программирования C#.	<b>Практическое занятие №28</b> 28.Создание программы «Расчёт заработной платы» с использованием полиморфизма	6	
<b>Раздел № 6. Технологии создания приложений на платформе .NET</b>				
Т 6.1	Современные технологии объектно-ориентированного программирования. Обзор технологий для создания десктопных и веб-приложений в среде MS Visual Studio.	<b>Практическое занятие №29</b> 29.Создание обучающей программы для изучения математического метода.	6	2
Т 6.2	Программирование обработчиков событий для классов и объектов. Обработка исключительных ситуаций в C#.	<b>Практическое занятие №30-31</b>	6	
		30.Программирование обработчиков событий	4	
		31. Обработка исключительных ситуаций	2	
Т 6.3	Создание многопоточных приложений. Написание многопоточных приложений. Типы многозадачности, используемые в операционных системах. Написание многопоточных приложений с использованием ASP.NET.	<b>Практическое занятие № 32</b> 32.Многопоточность и игры. Проект «Шахматы»	6	1
СЕМЕСТР 4.				
<b>Раздел № 7. Создание GUI с помощью MS VisualStudio</b>				
Т 7.1	Базовые элементы построения десктопных приложений Windows Form. Размещение элементов управления в формах Windows Forms. Создание элементов управления Windows Forms во время разработки. Использование технологии клиент-сервер при	<b>Практическое занятие №33-34</b>	6	
		33. Создание программы «Калькулятор» с использованием Windows Form	3	

	построении объектно-ориентированных программ.	34. Создание программы «Расчёт заработной платы» с использованием Windows Form	3	
T 7.2	Разработка пользовательских элементов управления Windows Forms в .NET Framework. Критерии оценки правильности построения классов.	<b>Практическое занятие №35</b> Разработка программы моделирования экосистемы.	6	2
T 7.3	Графические возможности в C#. Работа с растровыми и векторными изображениями с использованием классов Image, Bitmap и Metafile. Объект Graphics и графический контекст GDI	<b>Практическое занятие №36-37</b>	4	1
		36. Разработка системы классов для отрисовки графических объектов.	2	1
		37. Разработка классов для написания интерактивной графической игры.	2	
<b>Раздел № 8. Разработка приложений для БД</b>				
T 8.1	Обзор технологий построения приложений для БД в среде MS Visual Studio. Избежание повреждения существующего кода базы данных. Рефакторинг базы данных. Анализ кода базы данных.	<b>Практическое занятие №39-40</b>	6	2
		39. Разработка программ для работы с базами данных local MS SQL	3	2
		40. Разработка программ для работы с базами данных MS SQL	3	
T 8.2	Привязка элементов управления Windows Forms к данным в Visual Studio. Подготовка приложения к получению данных. Выборка данных в приложение. Привязка элементов управления к данным в Visual Studio. Редактирование данных в приложении. Проверка данных. Сохранение данных. Общие сведения о компоненте TableAdapterManager .	<b>Практическое занятие №41</b> Разработка программ для работы с базами данных MY SQL	6	
T 8.3	Язык интегрированных запросов LINQ, запросы к источнику данных. Согласованная модель для работы с источниками данными различных типов и форматов. Работа с XML-элементами при использовании объектной модели документа (DOM) W3C.	<b>Практическое занятие №42</b> Разработка программ для работы с нереляционными базами данных	6	1

## 5. Перечень учебной литературы, необходимой для освоения дисциплины

### 5.1. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Элементы учебно-методического комплекса дисциплины утверждены на заседании кафедры информационных технологий (протокол №1 от 29.08.2017).

Все материалы представлены в онлайн-курсе по ссылкам:

<http://elearn.dsum.org/course/view.php?id=10> .

<http://elearn.dsum.org/course/view.php?id=18>

### Контрольные вопросы для самоподготовки

1. Этапы решения задач на ЭВМ.
2. Что такое алгоритм? Привести примеры алгоритмов (из жизни, из алгебры, геометрии, русского языка и т.д.).
3. Сформулировать свойства алгоритма. Привести примеры.
4. Понятие исполнителя. Система команд исполнителя. Примеры.
5. Рассказать о формальном исполнении алгоритма. Трассировка. Примеры.
6. Основные алгоритмические структуры. Примеры алгоритмов.
7. Полная и неполная развилка. Примеры.
8. Циклы. Примеры.
9. Величины (постоянные, переменные), их характеристики. Как изменить значение переменной величины? Примеры.
10. Аргументы, результаты, промежуточные величины. Присваивание значений. Примеры.
11. Отношение между величинами в качестве условий. Примеры.
12. Принципы структурного программирования.
13. Типы величин (целые, вещественные, логические, символьные). Диапазон допустимых значений, представление в памяти ЭВМ, операции, определенные над величинами каждого типа.
14. Вспомогательный алгоритм. Формальные и фактические параметры, соответствие между ними.
15. Что такое каталог и подкаталог?
16. Что такое шаблоны файлов? В чем отличие шаблонов «?» и «\*»?
17. Какие вы знаете стандартные расширения файлов?
18. Как осуществляется выделение файлов по маске и произвольным образом и для чего это нужно?
19. Как найти файл, содержащий определенную строку символов?
20. Как осуществляются операции копирования, перемещения и удаления файлов, создания каталога?
21. Каковы назначение и возможности системы codeblobs?
22. Как запустить программу на трансляцию и выполнение?
23. Как записываются операторы начала и конца программы?
24. Какая функция является точкой входа в C++?
25. С помощью какого оператора функция возвращает вычисленное значение
26. Какой тип у функции, которая ничего не возвращает
27. Из каких разделов состоит программа на языке C++?
28. В какой последовательности должны быть записаны разделы программы на языке C++?
29. Как записываются операторы вывода на экран в C++?
30. Какие команды текстового редактора вы знаете?
31. Что такое блок текста и как его выделить?
32. Какие операторы используются для программирования разветвлений?
33. Как выполняются операторы перехода?
34. Какую из функций: Sin(x), Abs(x), Trunc(x) можно заменить условным оператором  $\text{if } (x < 0) \ x = -x;$ ?
35. Как записывается и как работает оператор FOR?
36. Для организации каких циклов применим оператор FOR?
37. В чем отличие оператора WHILE от оператора DO..WHILE?
38. Как программируются циклические алгоритмы с явно заданным числом повторений цикла?
39. Как программируются циклические алгоритмы с заранее неизвестным числом повторений цикла?
40. Напишите оператор цикла, который не выполняется ни разу.

41. Напишите оператор цикла, который выполняется неограниченное число раз.
42. Каким образом определяются переменные типа массив (одномерный и двумерный)?
43. Как осуществляется доступ к отдельному элементу одномерного и двумерного массива?
44. Каким образом выводятся элементы массива на экран?
45. Приведите пример фрагмента программы, который выводит на экран двумерный массив в виде матрицы.
46. Как описываются строковые переменные?
47. Какая максимальная длина строки допустима в C++?
48. Какие операции допустимы над строковыми данными?
49. В чем отличие строковой переменной от массива символов?
50. Какие стандартные процедуры и функции для работы со строками вы знаете?
51. Понятие указателя. Статические и динамические переменные.
52. Карта памяти. Динамическое распределение памяти. Создание и уничтожение динамических переменных.
53. Фрагментация динамической памяти. Освобождение динамической памяти.
54. Примеры использования динамической памяти.
55. Принципы структурного программирования. Блочный (модульный) подход к построению алгоритмов решения задач.
56. Процедуры. Вызов процедуры в программах. Формальные и фактические параметры. Соответствие между параметрами. Примеры.
57. Параметры-переменные, параметры-значения, параметры-константы. Примеры.
58. Глобальные и локальные идентификаторы. Видимость объектов (идентификаторов). Примеры.
59. Область действия переменных и других идентификаторов.
60. Функции. Их отличие от процедур. Способ обращения к функции. Примеры.
61. Символьный тип данных. Строковый тип данных. Строковые переменные и константы. Двойственный характер строк.
62. Функции, ориентированные на работу со строками. Примеры.
63. Процедуры, ориентированные на работу со строками. Примеры.
64. "Пузырьковая" сортировка (обменами).
65. Сортировка выбором.
66. Сортировка вставками.
67. Примеры рекурсивных функций: количество цифр в записи данного натурального числа; поиск минимального элемента в массиве; проверка, является ли число простым.
68. Рекурсия. Механизм рекурсии: действия на рекурсивном спуске и рекурсивном возврате. Понятие о глубине рекурсии. Примеры.
69. Однонаправленные списки и действия с ними.
70. Двухнаправленные списки.
71. Кольцевые списки. Списки с ключом.
72. Стек.
73. Очередь.
74. Дек.
75. Двоичные деревья. Формирование дерева. Добавление элемента в дерево. Печать дерева. Способы обхода дерева.
76. Двоичные деревья. Исключение компонента из дерева.
77. Понятие об ООП. Отличие ООП подхода к программированию от структурного.
78. Основные понятия ООП. Примеры.
79. Инкапсуляция. Примеры.
80. Наследование. Примеры.
81. Полиморфизм. Примеры.
82. Что такое комбинированный тип данных (запись)? Способы описания (примеры).

83. Способы обращения к компонентам записи.
84. Оператор присоединения и его использование.
85. Записи с вариантами. Примеры.
86. Понятие множества. Обозначение множества в C++.
87. Задание множественного типа и множественная переменная.
88. Операции над множествами. Примеры.
89. Понятие файла. Логические и физические файлы.
90. Файловые типы C++ (общие понятия).
91. Понятие текущей длины файла, указателя или окна файла, буфера файла.
92. Общие процедуры и функции для работы с файлами.
93. Компонентные (типизированные) файлы: описание, отличие от других видов файлов.
94. Текстовые файлы, принципы работы. Процедуры и функции, ориентированные на работу с текстовыми файлами, ввод-вывод информации.
95. Нетипизированные файлы и процедуры ввода-вывода. Примеры использования.
96. Прямой и последовательный доступ к компонентам файлов.
97. Работа с файлами записей. Примеры.
98. Понятие модуля. Разделы модуля. Подключение модуля. Компиляция модулей.
99. Построение диаграмм.
100. Построение графиков функций.

### **Примерный перечень тем курсовых работ:**

Тема. Исследование структур хранения данных и выполнения запросов в оперативной памяти и файлах.

Примерный перечень предметных областей, на основании которых может быть сформирован индивидуальный вариант задания:

- учёт абонентов телефонной сети (Ф. И. О., адрес, номер телефона, тариф, стоимость);
- учёт участников студенческой конференции (Ф. И. О., факультет, группа, курс, название научной секции, руководитель, название доклада);
- «Календарь погоды» (дата, температура воздуха, направление ветра, осадки);
- учёт участников предвыборной компании депутатов горсовета (Ф. И. О., возраст, политическая партия, должность, семейное положение);
- учёт книг в домашней библиотеке (название, автор, литературный жанр, издательство, цена);
- учёт автомобилей на платной стоянке (номер автомобиля, фамилия владельца, время пользования стоянкой, стоимость);
- учёт результатов баскетбольного турнира, данные о командах- участницах турнира (название, город, фамилия тренера, количество проведённых игр, количество побед);
- учёт компьютерного оборудования в учебном классе, данные о компьютере (учётный номер, код модели, тип процессора, размер диска, объём памяти);
- учёт результатов сдачи экзаменов студентами одной группы за один семестр (Ф. И. О. студента, номер зачётки, оценка по информатике, оценка по ин. яз., оценка по математике, оценка по физике);
- формирования графиков отпусков сотрудников кафедры, данные о сотрудниках (Ф. И. О., должность, табельный номер, продолжительность отпуска);
- «Электронный журнал преподавателя», учёт посещаемости и успеваемости студентов (Ф. И. О. студента, дисциплина, дата, посещение, оценка);



- «Расписание движения поездов по станции Ухта», информацию о поездах, данные о поездах (номер, пункт отправления, станция назначения, время в пути, время прибытия);
- учёт клиентов библиотеки, данные о клиентах (Ф. И. О., индивидуальный номер, название книги, автор, инвентарный номер книги, срок сдачи);
- формирования и учёт больничных листов, данные о больничных листах (Ф. И. О. больного, диагноз, номер больничного листа, фамилия врача, дата выдачи, дата закрытия, количество дней);
- учёт работы копировального аппарата (дата, фамилия (кто копировал), отдел (из какого отдела), количество страниц, количество копий, цена одной копии, процент надбавки на каждые сто листов, итого);
- учёт ассортимента обуви на складе (артикул (тип обуви – мужская, женская, детская), наименование, стоимость 1 пары, покупатель, отпущено (в парах), процент скидки за объём партии, стоимость партии);
- учёт оплаты за жилую площадь (желец, адрес, площадь жилого помещения, оплата (за один кв. метр), доплата за лишнюю жилплощадь (рассчитывается в зависимости от превышения жилой площади на одного проживающего), количество жильцов, итого);
- учёт клубных взносов в спортивном клубе (Ф. И. О. спортсмена, вид спорта, соревнование, место, клубный взнос, скидка на клубный взнос (рассчитывается в зависимости от занятого места), сумма взноса);
- учёт клиентов парикмахерского салона (клиент, мастер, квалификация мастера, услуга, дата, скидка постоянному клиенту, надбавка за квалификацию мастера, стоимость услуги);
- учёт продажи билетов в театре (название, жанр, режиссёр, стоимость билета, продано билетов, скидка на групповые заявки, дата, остаток билетов, выручено за спектакль);
- учёт отработанных часов и начисленной заработной платы сотрудников в отделе кадров (сотрудник, отдел, количество отработанных лет, отработанные часы, тариф, надбавка за выслугу лет, начислено);
- учёт доставки продуктов клиентам (клиент, продукт, количество, дальность доставки, надбавка за дальность доставки, скидка постоянному клиенту, сумма);
- учёт проката видео/аудио дисков (заказчик, название, тип носителя, формат записи, дата выдачи, дата возврата, количество дней, тариф в сутки проката, надбавка за задержку, итого);
- учёт работ строительно-отделочной организации (заказчик, вид услуги, стоимость, срок выполнения фирмой, срок выполнения заказчика, надбавка за срочность, стоимость материала заказчика, итоговая стоимость);
- учёт междугородных телефонных переговоров (абонент, город соединения, дата, коэффициент зоны, количество минут, льгота в ночное время, стоимость);
- учёт работы Интернет-клуба (пользователь, №компьютера, начало сеанса, время пользования, тариф (от быстрогодействия компьютера), стоимость услуги консультанта, скидка за ночное время, стоимость);
- учёт клиентов Интернет-магазина (фамилия, телефон, код товара, дата покупки, стоимость единицы товара, количество, итоговая сумма);
- учёт клиентов салона по продаже автомобилей (фамилия, № паспорта, марка

машины, год выпуска, сумма, скидка, итоговая сумма);

- учёт оптовой продажи стройматериалов (№ заказа, дата продажи, вид товара, стоимость единицы, кол-во товара, сумма, скидка, итоговая сумма);
- абитуриентов ВУЗа, данные об абитуриентах (Ф. И. О., адрес, № паспорта, специальность, средний балл, проходной балл, зачисление).

## 5.2. Перечень основной учебной литературы

1. Белоцерковская И.Е. Алгоритмизация. Введение в язык программирования С++. Учебное пособие: Учебное пособие [Электронный ресурс] / И.Е. Белоцерковская, Н.В. Галина, Л.Ю. Катаева. – Москва: НОУ Интуит, 2016. – 197 с. – Режим доступа: <https://www.book.ru/book/917549>
2. Белоцерковская И.Е. Объектное программирование в классах на С# 3.0. Учебное пособие: Курс лекций [Электронный ресурс] / И.Е. Белоцерковская, Н.В. Галина, Л.Ю. Катаева – Москва: НОУ Интуит, 2016. – 286 с. – Режим доступа: <https://www.book.ru/book/917807>
3. Вязовик Н.А. Программирование на Java. Учебное пособие: Курс лекций [Электронный ресурс] / Н.А. Вязовик - Москва: НОУ Интуит, 2016. – 604 с. – Режим доступа: <https://www.book.ru/book/918118>
4. Язык программирования С++ [Электронный ресурс]. – Режим доступа: <http://cppstudio.com/cat/274/>.
5. Руководство по С# [Электронный ресурс]. – Режим доступа: [https://professorweb.ru/my/csharp/charp\\_theory/level1/index.php](https://professorweb.ru/my/csharp/charp_theory/level1/index.php).
6. Г.Буч .Объектно-ориентированный анализ и проектирование с примерами приложений на С++ 4-е изд. – М. Бином, 2015. – 940 с.

## 5.3. Перечень дополнительной литературы

1. Кнут Д.Э. Искусство программирования: В 3 т. Т.1. Основные алгоритмы / Под общ. ред. Ю.В. Козаченко. – 3-е изд. - М.: Вильямс, 2004. – 713 с.
2. Кнут Д.Э. Искусство программирования: В 3 т. Т.2: Получисленные алгоритмы / Под общ. ред. Ю.В. Козаченко. – 3-е изд. – М.: Вильямс, 2004. – 829 с.
3. Кнут Д.Э. Искусство программирования: В 3 т. Т.3: Сортировка и поиск : учеб. пособие / Под общ. ред. Ю. В. Козаченко. - 2-е изд. – М.: Вильямс, 2004. – 823 с.
4. Степанов А.Н. Информатика. - СПб.: Питер, 2008. - 765 с. 5. Информатика и программирование: компьютерный практикум. - М.: Дашков и К, 2010. - 238 с.
5. Кудинов Ю. И., Пашенко Ф. Ф. Основы современной информатики: Учебное пособие. 2 е изд., испр. – СПб.: Издательство «Лань», 2011. – 256 с. Режим доступа: [http://e.lanbook.com/books/element.php?pl1\\_cid=25&pl1\\_id=2024](http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=2024)
6. Кауфман В.Ш. Языки программирования. Концепции и принципы. – М.: ДМК Пресс, 2010. – 464 с. Режим доступа: [http://e.lanbook.com/books/element.php?pl1\\_cid=25&pl1\\_id=1270](http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=1270)
7. Информатика и программирование: компьютерный практикум: учеб. пособие / А.Н. Гуда, М.А. Бутакова, Н.М. Нечитайло [и др.]; под общ. ред. В.И. Колесникова. – М.: Дашков и К, 2010. – 238 с.
8. Информатика: общий курс: учеб. / А.Н. Гуда, М.А. Бутакова, Н.М. Нечитайло [и др.]; под общ. ред. В.И. Колесникова. – 4-е изд. – М.: Дашков и К, 2011. – 400 с.

## 6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»,

**необходимых для освоения дисциплины**

1. <http://www.inuit.ru> – национальный открытый университет
2. <http://www.sql.ru/> - портал ИТ-разработчиков
3. <http://habrahabr.ru/> - портал ИТ-специалистов
4. <http://www.osp.ru/> - Издательство Открытые системы

**7. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем****7.1. Перечень информационных технологий**

Технология командной разработки проектов на GitHub. Дистанционное обучение с помощью СДО Moodle.

**7.2. Перечень программного обеспечения**

Для проведения лабораторных занятий и выполнения самостоятельной работы необходим компьютерный класс; программное обеспечение: операционная система Windows XP и выше, пакет Microsoft Visual Studio 2015 и выше, Git Desktop, Codebloks.

**8. Фонд оценочных средств для контроля уровня сформированности компетенций****8.1. Виды промежуточной аттестации.**

Текущий контроль успеваемости позволяет оценить уровень сформированности элементов компетенций (знаний и умений), компетенций с последующим объединением оценок и проводится в форме устного опроса (фронтальный, индивидуальный, комплексный), письменной проверки (ответы на вопросы, тестовые задания), включая задания для самостоятельной работы.

Промежуточная аттестация по итогам освоения дисциплины в форме диф.зачета, позволяет оценить уровень сформированности компетенций и может осуществляться по результатам текущего контроля и итоговой контрольной работы, тестовых заданий и т.п.

Промежуточная аттестация в форме экзамена позволяет оценить уровень сформированности компетенций в целом по дисциплине осуществляется в устной форме.

**8.2. Показатели и критерии оценки результатов освоения дисциплины.**

Средним баллом за дисциплину является средний балл за текущую учебную деятельность.

Механизм конвертации результатов изучения студентом дисциплины в оценки по традиционной (государственной) шкале и шкале ECTS представлен в таблице.

<b>Средний балл по дисциплине (текущая успеваемость)</b>	<b>Отношение полученного студентом среднего балла по дисциплине к максимально возможной величине этого показателя</b>	<b>Оценка по государственной шкале</b>	<b>Оценка по шкале ECTS</b>	<b>Определение</b>
4,5 –5,0	90% – 100%	5	A	отлично – отличное выполнение с незначительным количеством неточностей (до 10%)

4,0 – 4,49	80% – 89%	4	В	хорошо – в целом правильно выполненная работа с незначительным количеством ошибок (до 20%)
3,75 – 3,99	75% – 79%	4	С	хорошо – в целом правильно выполненная работа с незначительным количеством ошибок (до 25%)
3,25 – 3,74	65% – 74%	3	Д	удовлетворительно – неплохо, но со значительным количеством недостатков (до 35%)
3,0 – 3,24	60% – 64%	3	Е	достаточно – выполнение удовлетворяет минимальные критерии, но со значительным количеством недостатков (до 40%)
до 3,0	35% – 59%	2	FX	неудовлетворительно с возможностью повторной сдачи (ошибок свыше 40%)
	0 – 34%	2	F	неудовлетворительно – надо поработать над тем, как получить положительную оценку (ошибок свыше 65%)

### 8.3. Критерии оценки работы студента.

При усвоении каждой темы за текущую учебную деятельность студента выставляются оценки по 5-балльной (государственной) шкале. Оценка за каждое задание в процессе текущей учебной деятельности определяется на основе процентного отношения операций, правильно выполненных студентом во время выполнения задания:

- 90-100% – «5»,
- 75-89% – «4»,
- 60-74% – «3»,
- менее 60% – «2».

Если на занятии студент выполняет несколько заданий, оценка за каждое задание выставляется отдельно.

#### 8.3.1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или опыта деятельности, характеризующие этапы формирования компетенций в процессе освоения образовательной программы)

## Контрольные задания

### Раздел № 1. Информация и программирование.

#### Вариант 1:

1. Написать программу, которая выводит первую букву вашего имени в виде звёздочек, а затем все остальные буквы.

\* \* \* \* \*

```
*
*
*
*   аня
```

2. На вход программе через стандартный ввод передаются два целых числа, по модулю не превышающие 100000. Выведите в стандартный вывод их сумму.

stdin	stdout
2 3	5
-4 -9	-13
-1 2	1

3. Написать программу, которая, реализует диалог с пользователем: запрашивает с клавиатуры два целых числа, и выводит на экран сумму данных чисел.

4. Написать программу, которая подсчитывает площадь равностороннего треугольника, периметр которого равен  $p$ .

5. Написать программу, которая определяет наибольшую цифру в натуральном двухзначном числе.

### Раздел № 2. Алгоритмизация и программирование на языках высокого уровня.

#### Вариант 1:

1. Дана строка. Найдите в этой строке второе вхождение буквы **f** и выведите индекс этого вхождения. Если буква **f** в данной строке встречается только один раз, выведите число -1, а если не встречается ни разу, выведите число -2. Индексы нумеруются с нуля.

#### Пример

stdin	stdout
comfort	-1
coffee	3
car	-2

2. В stdin даны два натуральных числа. Найти их наибольший общий делитель.

#### Пример

stdin	stdout
25 27	1
12 16	4

13 13 13

3. На вход дано целое положительное число  $N$ . Выведите его в двоичной системе счисления без ведущих нулей.

### Пример

stdin	stdout
5	101
32	100000
1	1

### Подсказка

- число  $N$  можно считывать в переменную типа `int`
- в этой задаче вам может пригодиться контейнер `vector`; напомним, что добавление значения  $x$  в конец вектора  $v$  делается так: `v.push_back(x)`;

## Раздел № 3. Структурное программирование на C++.

### Вариант 1:

1. Смоделировать бросание игральной кости 20 раз. Выводить по 5 значений в строке.
2. Проверьте вероятность выпадания каждой из граней. Используйте 6000000 бросков.
3. Напишите код для простой игры. Правила игры: Компьютер загадывает число от 5 до 20 и просит игрока угадать число. Если угадано, выводится сообщение "Вы угадали за  $n$  попыток". Иначе выводится сообщение "Загаданное число меньше/больше" и предлагается ещё раз угадать число. Реализуйте код в виде функций (бросок и проверка).
4. Сколько необходимо сделать попыток, чтобы отгадать любое число в этом диапазоне?

## Раздел № 4. Структуры данных и файлов

### Вариант 1.

1. На языке C# разработать контейнер «коллекция» — `TCollection`, который:
  - содержит элементы типа `TCollectionItem`, содержащие ссылку на своего владельца,
  - увеличивает свой размер при добавлении в него элементов,
  - предоставляет типовые свойства (текущее количество элементов, количество зарезервированных элементов),
  - предоставляет типовые процедуры, применяемые при работе с коллекцией (вставка, удаление, очистка, копирование в другую коллекцию, поиск индекса элемента по ссылке, сортировка),
  - предоставляет типовые события, происходящие при работе с коллекцией (вставка, удаление, очистка),
  - при обращении к элементу коллекции проверяет, что индекс находится в допустимых пределах.

Продемонстрировать использование класса.

2. Представьте заданную таблицу в виде файла БД, выберите тип поля и его разрядность.

Список студентов

1. ФИО

1. Фамилия

- 2. Имя
- 3. Отчество
- 2. Пол
- 3. Возраст
- 4. Курс
- 5. Специальность

- Напишите функции создания файла, добавления в него записей, удаления записей, исправления записей, вывода содержимого файла на экран в виде таблицы и просмотра этой таблицы страницами, двигаясь вверх и вниз по записям, по страницам, сразу перемещаясь к началу или концу файла.
- Напишите функцию, которая сортировала бы файл по любому указанному полю.
- Напишите функцию, которая отыскивала бы и выводила для просмотра записи файла найденные по заданному логическому выражению, содержащему поля файла, числа, операции.
- Напишите программу, которая бы объединяла все выше описанные функции с помощью комбинированного меню в единую систему.

## Раздел № 5 Объектно-ориентированное программирование на C#

### Вариант 1:

Описать класс Комплексное число. Объекты данного класса должны состоять из двух полей разделённых символом  $i$ . Первое поле задает значение действительной части числа, а второе - значение мнимой. Каждое из полей может содержать только символы десятичных цифр и символы  $-$  и  $+$ , определяющие знак числа. Символы  $-$  или  $+$  могут находиться только в первой позиции числа, причем символ  $+$  может отсутствовать, в этом случае число считается положительным. Если в составе инициализирующей строки будет встречен любой символ, отличный от допустимых, объект класса Комплексное число должен принимать нулевое значение. Примеры строк:  $33i12$ ,  $-7U00$ ,  $+5i-21$ .

Для класса Комплексное\_число определить следующие методы:

- проверка на равенство;
- сложение чисел;
- умножение чисел.

Написать тестовую программу, которая демонстрирует работу класса.

## Раздел № 6. Технологии создания приложений на платформе .NET

### Вариант 1:

В исследовательской лаборатории фирмы Robots&Co разработали новую модель робота. Главной особенностью данной модели робота является то, что он работает по заранее заданной программе, в которой могут присутствовать команды: сделать шаг на Юг, на Север, на Восток или на Запад. Робот исполняет программу строго последовательно и, дойдя до конца программы, останавливается. Специалисты из Robots&Co заинтересовались вопросом, сколько существует различных программ, состоящих из  $K$  инструкций, таких, что робот, выйдя из начала координат, придет в точку с координатами  $(X, Y)$ . Оси координат располагаются параллельно сторонам света, и единица измерения, соответствует одному шагу робота. Напишите программу, которая дает ответ на этот вопрос и показывает движение робота в реальном времени.

Формат входных данных

Во входном файле находятся три числа  $K$ ,  $X$  и  $Y$  ( $0 \leq K \leq 16$ ,  $|X|, |Y| \leq 16$ ), разделенные пробелами.

Формат выходных данных

В выходной файл ваша программа должна поместить одно число — количество программ для робота.

## Раздел № 7. Создание GUI с помощью MS VisualStudio

### Вариант 1:

1. Используя, пример построения графика, разберитесь с основными методами класса Graphics
2. График какой функции строится в примере?
3. Создайте функционал для построения графика любой линейной функции  $y=a*x+b$ . Отрисовку графика на холсте панели перенесите в отдельный класс.
4. Расширьте проект, создав класс для отрисовки графика функции  $y=x^2$

## Раздел № 8. Разработка приложений для БД

### Вариант 1:

1. Реализуйте приложение для БД, используя пример из раздела.
2. Добавьте к БД из примера новую таблицу "категории товара" с двумя полями (id, name). К таблице товары добавьте поле "код товара".
3. Отобразите в Grid выборку данных из двух таблиц
4. Реализуйте функционал фильтрации данных, путём выбора из ComboBox необходимой для отображения категории товара.
5. Дополнительно. Реализуйте поиск данных по любому полю.

*Критерии оценивания компетенций (результатов) по уровням освоения учебного материала:*

- 1 – репродуктивный (освоение знаний, выполнение деятельности по образцу, инструкции или под руководством), если самостоятельно (или с помощью преподавателя) выполнены все пункты работы;
- 2 – продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач; применение умений в новых условиях), если выполнены все пункты работы самостоятельно и улучшена точность результата;
- 3 – творческий (самостоятельное проектирование экспериментальной деятельности; оценка и самооценка инновационной деятельности), если предложен более рациональный алгоритм решения задачи.

### 8.3.2. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и/или опыта деятельности:

Оценивание проводится преподавателем в течении всего учебного процесса на основе выполнения текущих контрольных и индивидуальных заданий, самостоятельной работы за компьютером, тестирования в системе elearn.. Результаты выполнения практических работ предъявляются в виде отчетов в электронном виде(в системе elearn). Оценивание практических работ осуществляет преподаватель, который проводит практические занятия.

В конце 2 семестра студенты защищают курсовой проект, который выполняется ими в течение семестра.



Зачеты (1 и 3 семестр) принимает преподаватель, который проводит практические занятия. 2 и 4 семестры курса заканчиваются экзаменом, который принимает преподаватель читающий лекции.

#### **9. Методические указания для обучающихся по освоению дисциплины**

Рекомендации, позволяющие обучающимся оптимальным образом организовать процесс изучения как теоретического учебного материала дисциплины, так и подготовки к практическим занятиям: изучение лекций, коллективное обсуждение тем на практических занятиях, индивидуальная работа за компьютером, самостоятельная работа над текущими темами, самостоятельная работа над индивидуальными заданиями представлены в виде элементов электронного курса в системе elearn

<http://elearn.dsum.org/course/view.php?id=10> .

<http://elearn.dsum.org/course/view.php?id=18>

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

При изучении дисциплины используются компьютерные классы первого корпуса (704 или 705). Компьютерные классы оборудованы ПК IBM PC (Core I5-750 2.66 GHz, MB S-1156 ASUS P7P55 Lx, DDR3 2200 4 Gb, PCI Express, HDD 1,5 Tb), лекционные аудитории – мультимедийной проекционной техникой в стационарном или переносном варианте. Чтение лекций сопровождается демонстраций электронных презентаций по каждой теме.

#### **11.Иные сведения и (или) материалы: (включаются на основании решения кафедры)**

### Оформление сведений о дополнении и изменении рабочей программы учебной дисциплины

Рабочие программы учебных дисциплин ежегодно обсуждаются, актуализируются на заседаниях ПМК, рассматриваются на заседаниях кафедр и утверждаются проректором по учебной работе, информация об изменениях отражается в листе сведений о дополнении и изменении рабочей программы учебной дисциплины. В случае существенных изменений программа полностью переформируется. Обновленный электронный вариант программы размещается на сервере ГОУ ВПО «ДонАУиГС».

Изменения в РПУД могут вноситься в следующих случаях:

- изменение государственных образовательных стандартов или других нормативных документов, в том числе локальных нормативных актов;
- изменение требований работодателей к выпускникам;
  - разработка новых методик преподавания и контроля знаний студентов.

Ответственность за актуализацию РПУД несут преподаватели, реализующие дисциплину.

### СВЕДЕНИЯ О ДОПОЛНЕНИИ И ИЗМЕНЕНИИ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ НА 20\_\_/20\_\_ УЧЕБНЫЙ ГОД

«Название дисциплины»

Направление подготовки

(профиль/магистерская программа)

**ДОПОЛНЕНО** (с указанием раздела РПУД)

**ИЗМЕНЕНО** (с указанием раздела РПУД)

**УДАЛЕНО** (с указанием раздела РПУД)

Реквизиты протокола заседания кафедры  
от \_\_\_\_\_ № \_\_\_\_\_  
дата