

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Комсомольский-на-Амуре государственный университет»

УТВЕРЖДАЮ

Декан факультета

компьютерных технологий

Григорьев Я.Ю.

« 01 » 06 2020 г.

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

### Объектно-ориентированное программирование

Направление подготовки	<i>09.03.01 "Информатика и вычислительная техника"</i>
Направленность (профиль) образовательной программы	<i>Программное обеспечение средств вычислительной техники и автоматизированных систем</i>
Квалификация выпускника	<i>бакалавр</i>
Год начала подготовки (по учебному плану)	<i>2020</i>
Форма обучения	<i>очная</i>
Технология обучения	<i>традиционная</i>

Курс	Семестр	Трудоемкость, з.е.
2	4	3

Вид промежуточной аттестации	Обеспечивающее подразделение
<i>Зачет с оценкой</i>	<i>Кафедра «Проектирование, управление и разработка информационных систем»</i>

Комсомольск-на-Амуре 2020

Разработчик рабочей программы:

Доцент, Кандидат технических наук



Гордин С.А.

СОГЛАСОВАНО:

Заведующий кафедрой

Кафедра «Проектирование, управление и  
разработка информационных систем»



Тихомиров В.А.

## 1 Общие положения

Рабочая программа дисциплины «Объектно-ориентированное программирование» составлена в соответствии с требованиями федерального государственного образовательного стандарта, утвержденного приказом Министерства образования и науки Российской Федерации №929 19.09.2017, и основной профессиональной образовательной программы подготовки «Программное обеспечение средств вычислительной техники и автоматизированных систем» по направлению 09.03.01 "Информатика и вычислительная техника".

Задачи дисциплины	Дать представление студентам об объектно-ориентированном программировании в объеме достаточном для использования различных библиотек классов.
Основные разделы / темы дисциплины	Классы: основные понятия Иерархии классов Структуры данных, коллекции и классы прототипы

## 2 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Процесс изучения дисциплины «Объектно-ориентированное программирование» направлен на формирование следующих компетенций в соответствии с ФГОС ВО и основной образовательной программой (таблица 1):

Таблица 1 – Компетенции и индикаторы их достижения

Код по ФГОС	Индикаторы достижения	Планируемые результаты обучения по дисциплине
Универсальные		
Общепрофессиональные		
ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения;	ОПК-8.1 Знает алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения ОПК-8.2 Умеет составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули ОПК-8.3 Владеет навыками программирования, отладки и тестирования прототипов программно-	Знать типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения  Уметь применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов  Владеть навыками использования библиотек программных модулей, шаблонов, классов объектов при разработке ПО

Код по ФГОС	Индикаторы достижения	Планируемые результаты обучения по дисциплине
	технических комплексов задач	
Профессиональные		

### 3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Объектно-ориентированное программирование» изучается на 2 курсе в 4 семестре.

Дисциплина входит в состав блока 1 «Дисциплины (модули)» и относится к обязательной части.

Для освоения дисциплины необходимы знания, умения, навыки и (или) опыт практической деятельности, сформированные в процессе изучения дисциплин / практик: «Современные программные средства», «Языки программирования».

Знания, умения и навыки, сформированные при изучении дисциплины «Объектно-ориентированное программирование», будут востребованы при изучении последующих дисциплин «Структуры данных и алгоритмы», «Математическая логика и теория алгоритмов», «Программирование мобильных устройств».

Входной контроль проводится в виде тестирования. Задания тестов представлены в приложении 1 РПД.

### 4 Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость (объем) дисциплины составляет 3 з.е., 108 акад. час.

Распределение объема дисциплины (модуля) по видам учебных занятий представлено в таблице 2.

Таблица 2 – Объем дисциплины (модуля) по видам учебных занятий

Объем дисциплины	Всего академических часов
Общая трудоемкость дисциплины	108
<b>Контактная аудиторная работа обучающихся с преподавателем (по видам учебных занятий), всего</b>	64
В том числе:	
<b>занятия лекционного типа</b> (лекции и иные учебные занятия, предусматривающие преимущественную передачу учебной информации педагогическими работниками)	32
<b>занятия семинарского типа</b> (семинары, практические занятия, практикумы, лабораторные работы, коллоквиумы и иные аналогичные занятия)	32
<b>Самостоятельная работа обучающихся и контактная работа</b> , включающая групповые консультации, индивидуальную работу обучающихся с преподавателями (в том числе индивидуальные консультации);	44

Объем дисциплины	Всего академических часов
взаимодействие в электронной информационно-образовательной среде вуза	
Промежуточная аттестация обучающихся – Зачет с оценкой	-

**5 Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебной работы**

Таблица 3 – Структура и содержание дисциплины (модуля)

Наименование разделов, тем и содержание материала	Виды учебной работы, включая самостоятельную работу обучающихся и трудоемкость (в часах)			
	Контактная работа преподавателя с обучающимися			СРС
	Лекции	Семинарские (практические занятия)	Лабораторные занятия	
<b><i>Раздел 1 Объектно-ориентированное программирование</i></b>				
<b>Тема 1. Первый взгляд на объектно-ориентированное программирование.</b> Классы и объекты. Обзор языков и средств разработки.	2	-	2	3
<b>Тема 2. Основные понятия языка.</b> Состав языка. Типы данных.	2	-	2	3
<b>Тема 3. Переменные, операции и выражения.</b> Переменные. Именованные константы. Операции и выражения. Линейные программы.	2	-	2	3
<b>Тема 4. Операторы.</b> Выражения, блоки и пустые операторы. Операторы ветвления. Операторы цикла. Базовые конструкции структурного программирования. Обработка исключительных ситуаций. Операторы checked и unchecked.	2	-	2	3
<b>Тема 5. Классы: основные понятия.</b> Присваивание и сравнение объектов. Данные: поля и константы. Методы. Ключевое слово this. Конструкторы. Свойства.	2	-	2	3
<b>Тема 6. Массивы и строки.</b> Одномерные, двумерные и ступенчатые массивы. Оператор foreach. Массивы объектов. Символы и строки. Вспомогательные классы.	2	-	2	3
<b>Тема 7. Классы: подробности.</b> Перегрузка методов. Рекурсивные методы. Методы с переменным количеством аргументов. Метод Main. Индексаторы. Операции класса. Де-	2	-	2	3

Наименование разделов, тем и содержание материала	Виды учебной работы, включая самостоятельную работу обучающихся и трудоемкость (в часах)			
	Контактная работа преподавателя с обучающимися			СРС
	Лекции	Семинарские (практические занятия)	Лабораторные занятия	
структуры. Вложенные типы.				
<b>Тема 8. Иерархии классов.</b> Наследование. Виртуальные методы. Раннее и позднее связывание. Абстрактные классы. Бесплодные классы. Класс object.	2	-	2	3
<b>Тема 9. Интерфейсы и структурные типы.</b> Синтаксис интерфейса. Реализация интерфейса. Преобразования типов. Работа с объектами через интерфейсы. Операции is и as. Интерфейсы и наследование. Стандартные интерфейсы. Структуры.	2	-	2	3
<b>Тема 10. Делегаты, события и потоки выполнения.</b> Делегаты. События. Многопоточные приложения.	2	-	2	3
<b>Тема 11. Работа с файлами.</b> Потоки байтов. Асинхронный ввод-вывод. Потоки символов. Двоичные потоки. Консольный ввод-вывод. Работа с каталогами и файлами. Сохранение объектов (сериализация).	2	-	2	3
<b>Тема 12. Сборки, библиотеки, атрибуты, директивы препроцессора.</b> Сборки. Создание библиотеки. Использование библиотеки. Рефлексия. Атрибуты. Пространства имен. Директивы препроцессора.	2	-	2	3
<b>Тема 13. Структуры данных, коллекции и классы прототипы.</b> Абстрактные структуры данных. Коллекции. Классы прототипы. Частичные типы. Обнуляемые типы.	2	-	2	3
<b>Тема 14. Введение в программирование под Windows.</b> Событийно-управляемое программирование. Шаблон Windows-приложения. Элементы управления. Диалоговые окна. Краткое введение в графику.	4	-	4	3
<b>Тема 15. Технологии дополняющие ООП</b> Документирование в формате XML. Язык UML. Паттерны ООП.	2	-	2	2
<b>ИТОГО по дисциплине</b>	<b>32</b>		<b>32</b>	<b>44</b>

## **6 Внеаудиторная самостоятельная работа обучающихся по дисциплине (модулю)**

При планировании самостоятельной работы студенту рекомендуется руководствоваться следующим распределением часов на самостоятельную работу (таблица 4):

Таблица 4 – Рекомендуемое распределение часов на самостоятельную работу

<b>Компоненты самостоятельной работы</b>	<b>Количество часов</b>
Изучение теоретических разделов дисциплины	14
Подготовка к занятиям семинарского типа	14
Подготовка и оформление РГР	16
	44

## **7 Оценочные средства для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине (модулю)**

Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации представлен в Приложении 1.

Полный комплект контрольных заданий или иных материалов, необходимых для оценивания результатов обучения по дисциплине (модулю), практике хранится на кафедре-разработчике в бумажном и электронном виде.

## **8 Учебно-методическое и информационное обеспечение дисциплины (модуля)**

### **8.1 Основная литература**

1 Немцова, Т. И. Программирование на языке С++ [Электронный ресурс]: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2012. - 512 с. // ZNANIUM.COM : электронно-библиотечная система. – Режим доступа: <http://znanium.com/catalog.php#>, ограниченный. – Загл. с экрана.

### **8.2 Дополнительная литература**

1 Павловская, Т. А. С/С++. Объектно-ориентированное программирование: учебник для вузов / Т. А. Павловская. – СПб. : Питер, 2010; 2003; 2001. – 460 с.

2 Павловская, Т.А. С/С++. С/ С++. Объектно-ориентированное программирование. Структурное программирование: учеб. пособие для вузов: практикум / Т.А. Павловская, Ю.А. Щупак. — СПб.: Питер, 2003 — 238 с. : ил.– (Серия «Учебное пособие»)

3 Павловская, Т.А. С #: Объектно-ориентированное программирование: Учебник для вузов / Т. А. Павловская. - СПб.: Питер, 2010; 2007. - 432с.

### 8.3 Методические указания для студентов по освоению дисциплины

Методические указания к выполнению индивидуальных заданий приведены в учебнике [3] из списка основной литературы (раздел 8.2); теоретический материал в учебнике [3] из списка дополнительной литературы (раздел 8.2).

### 8.4 Современные профессиональные базы данных и информационные справочные системы, используемые при осуществлении образовательного процесса по дисциплине

1 Электронно-библиотечная система ZNANIUM.COM. Договор ЕП 44 № 003/10 эбс ИКЗ 191272700076927030100100120016311000 от 17 апреля 2019 г.

2 Электронно-библиотечная система IPRbooks. Лицензионный договор № ЕП44 № 001/9 на предоставление доступа к электронно-библиотечной системе IPRbooks ИКЗ 191272700076927030100100090016311000 от 27 марта 2019 г.

3 Электронно-библиотечная система eLIBRARY.RU. Договор № ЕП 44 № 004/13 на оказание услуг доступа к электронным изданиям ИКЗ 91272700076927030100100150016311000 от 15 апреля 2019 г.

4 Информационно-справочные системы «Кодекс»/ «Техэксперт». Соглашение о сотрудничестве № 25/19 от 31 мая 2019 г.

### 8.5 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)

C++ Standard Library Reference [Электронный ресурс] / Colin Robertson, Mike Blome, Gordon Hogenson, Saisang Cai. Дата обновления: 04.11.2016. – Режим доступа: <https://docs.microsoft.com/ru-ru/cpp/standard-library/cpp-standard-library-reference>, свободный. – Загл. с экрана. Яз. англ.

### 8.6 Лицензионное и свободно распространяемое программное обеспечение, используемое при осуществлении образовательного процесса по дисциплине

Таблица 7 – Перечень используемого программного обеспечения

Наименование ПО	Реквизиты / условия использования
Microsoft Imagine Premium	Лицензионный договор АЭ223 №008/65 от 11.01.2019
OpenOffice	Свободная лицензия, условия использования по ссылке: <a href="https://www.openoffice.org/license.html">https://www.openoffice.org/license.html</a>
Visual Studio Community 2019 и более поздние версии	Visual Studio Community может использовать неограниченное число пользователей в организации в следующих случаях: в учебных аудиториях, для научных исследований или участия в проектах с открытым кодом.

## 9 Организационно-педагогические условия

Организация образовательного процесса регламентируется учебным планом иписанием учебных занятий. Язык обучения (преподавания) — русский. Для всех видов аудиторных занятий академический час устанавливается продолжительностью 45 минут.

При формировании своей индивидуальной образовательной траектории обучающийся имеет право на перезачет соответствующих дисциплин и профессиональных моду-



лей, освоенных в процессе предшествующего обучения, который освобождает обучающегося от необходимости их повторного освоения.

### **9.1 Образовательные технологии**

Учебный процесс при преподавании курса основывается на использовании традиционных, инновационных и информационных образовательных технологий. Традиционные образовательные технологии представлены лекциями и семинарскими (практическими) занятиями. Инновационные образовательные технологии используются в виде широкого применения активных и интерактивных форм проведения занятий. Информационные образовательные технологии реализуются путем активизации самостоятельной работы студентов в информационной образовательной среде.

### **9.2 Занятия лекционного типа**

Лекционный курс предполагает систематизированное изложение основных вопросов учебного плана.

На первой лекции лектор обязан предупредить студентов, применительно к какому базовому учебнику (учебникам, учебным пособиям) будет прочитан курс.

Лекционный курс должен давать наибольший объем информации и обеспечивать более глубокое понимание учебных вопросов при значительно меньшей затрате времени, чем это требуется большинству студентов на самостоятельное изучение материала.

### **9.3 Занятия семинарского типа**

Семинарские занятия представляют собой детализацию лекционного теоретического материала, проводятся в целях закрепления курса и охватывают все основные разделы.

Основной формой проведения семинаров является обсуждение наиболее проблемных и сложных вопросов по отдельным темам, а также разбор примеров и ситуаций в аудиторных условиях. В обязанности преподавателя входят: оказание методической помощи и консультирование студентов по соответствующим темам курса.

Активность на семинарских занятиях оценивается по следующим критериям:

- ответы на вопросы, предлагаемые преподавателем;
- участие в дискуссиях;
- выполнение проектных и иных заданий;
- ассистирование преподавателю в проведении занятий.

Ответ должен быть аргументированным, развернутым, не односложным, содержать ссылки на источники.

Доклады и оппонирование докладов проверяют степень владения теоретическим материалом, а также корректность и строгость рассуждений.

Оценивание заданий, выполненных на семинарском занятии, входит в накопленную оценку.

### **9.4 Самостоятельная работа обучающихся по дисциплине (модулю)**

Самостоятельная работа студентов – это процесс активного, целенаправленного приобретения студентом новых знаний, умений без непосредственного участия преподавателя, характеризующийся предметной направленностью, эффективным контролем и оценкой результатов деятельности обучающегося.

Цели самостоятельной работы:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную и справочную документацию, специальную литературу;

- развитие познавательных способностей, активности студентов, ответственности и организованности;
- формирование самостоятельности мышления, творческой инициативы, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений и академических навыков.

Самостоятельная работа может осуществляться индивидуально или группами студентов в зависимости от цели, объема, уровня сложности, конкретной тематики.

Технология организации самостоятельной работы студентов включает использование информационных и материально-технических ресурсов университета.

Перед выполнением обучающимися внеаудиторной самостоятельной работы преподаватель может проводить инструктаж по выполнению задания. В инструктаж включается:

- цель и содержание задания;
- сроки выполнения;
- ориентировочный объем работы;
- основные требования к результатам работы и критерии оценки;
- возможные типичные ошибки при выполнении.

Инструктаж проводится преподавателем за счет объема времени, отведенного на изучение дисциплины.

Контроль результатов внеаудиторной самостоятельной работы студентов может проходить в письменной, устной или смешанной форме.

Студенты должны подходить к самостоятельной работе как к наиважнейшему средству закрепления и развития теоретических знаний, выработке единства взглядов на отдельные вопросы курса, приобретения определенных навыков и использования профессиональной литературы.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

### **9.5 Методические указания для обучающихся по освоению дисциплины**

При изучении дисциплины обучающимся целесообразно выполнять следующие рекомендации:

1. Изучение учебной дисциплины должно вестись систематически.
2. После изучения какого-либо раздела по учебнику или конспектным материалам рекомендуется по памяти воспроизвести основные термины, определения, понятия раздела.
3. Особое внимание следует уделить выполнению отчетов по практическим занятиям и индивидуальным комплексным заданиям на самостоятельную работу.
4. Вся тематика вопросов, изучаемых самостоятельно, задается на лекциях преподавателем. Им же даются источники (в первую очередь вновь изданные в периодической научной литературе) для более детального понимания вопросов, озвученных на лекции.

При самостоятельной проработке курса обучающиеся должны:

- просматривать основные определения и факты;
- повторить законспектированный на лекционном занятии материал и дополнить его с учетом рекомендованной по данной теме литературы;
- изучить рекомендованную литературу, составлять тезисы, аннотации и конспекты наиболее важных моментов;
- самостоятельно выполнять задания, аналогичные предлагаемым на занятиях;
- использовать для самопроверки материалы фонда оценочных средств.

## **10 Описание материально-технического обеспечения, необходимого для осуществления образовательного процесса по дисциплине (модулю)**

### **10.1 Учебно-лабораторное оборудование**

Таблица 8 – Перечень оборудования лаборатории

Аудитория	Наименование аудитории (лаборатории)	Используемое оборудование
компьютерные классы ФКТ	Учебные лаборатории «Полигон вычислительной техники» 321(3)	10 персональных ЭВМ, каждая из которых оснащена процессором Intel(R) Core (TM) i3-2100 CPU @3.10 GHz и оперативной памятью 2ГБ. Операционная система - Windows 7. В классе имеется сетевой коммутатор Cisco catalyst 2960 с ПО IOS ver 12.2(55)SE5.

### **10.2 Технические и электронные средства обучения**

При проведении занятий используется аудитория, оборудованная проектором (стационарным или переносным) для отображения презентаций. Кроме того, при проведении лекций и практических занятий необходим компьютер с установленным на нем браузером и программным обеспечением для демонстрации презентаций.

## **11 Иные сведения**

### **Методические рекомендации по обучению лиц с ограниченными возможностями здоровья и инвалидов**

Освоение дисциплины обучающимися с ограниченными возможностями здоровья может быть организовано как совместно с другими обучающимися, так и в отдельных группах. Предполагаются специальные условия для получения образования обучающимися с ограниченными возможностями здоровья.

Профессорско-педагогический состав знакомится с психолого-физиологическими особенностями обучающихся инвалидов и лиц с ограниченными возможностями здоровья, индивидуальными программами реабилитации инвалидов (при наличии). При необходимости осуществляется дополнительная поддержка преподавания тьюторами, психологами, социальными работниками, прошедшими подготовку ассистентами.

В соответствии с методическими рекомендациями Минобрнауки РФ (утв. 8 апреля 2014 г. N АК-44/05вн) в курсе предполагается использовать социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе. Подбор и разработка учебных материалов производится с учетом предоставления материала в различных формах: аудиальной, визуальной, с использованием специальных технических средств и информационных систем.

Освоение дисциплины лицами с ОВЗ осуществляется с использованием средств обучения общего и специального назначения (персонального и коллективного использования). Материально-техническое обеспечение предусматривает приспособление аудиторий к нуждам лиц с ОВЗ.

Форма проведения аттестации для студентов-инвалидов устанавливается с учетом индивидуальных психофизических особенностей. Для студентов с ОВЗ предусматривается доступная форма предоставления заданий оценочных средств, а именно:

- в печатной или электронной форме (для лиц с нарушениями опорно-двигательного аппарата);
- в печатной форме или электронной форме с увеличенным шрифтом и контрастностью (для лиц с нарушениями слуха, речи, зрения);
- методом чтения ассистентом задания вслух (для лиц с нарушениями зрения).

Студентам с инвалидностью увеличивается время на подготовку ответов на контрольные вопросы. Для таких студентов предусматривается доступная форма предоставления ответов на задания, а именно:

- письменно на бумаге или набором ответов на компьютере (для лиц с нарушениями слуха, речи);
- выбором ответа из возможных вариантов с использованием услуг ассистента (для лиц с нарушениями опорно-двигательного аппарата);
- устно (для лиц с нарушениями зрения, опорно-двигательного аппарата).

При необходимости для обучающихся с инвалидностью процедура оценивания результатов обучения может проводиться в несколько этапов.

## ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине

### Объектно-ориентированное программирование

Направление подготовки	<i>09.03.01 "Информатика и вычислительная техника"</i>
Направленность (профиль) образовательной программы	<i>Программное обеспечение средств вычислительной техники и автоматизированных систем</i>
Квалификация выпускника	<i>бакалавр</i>
Год начала подготовки (по учебному плану)	<i>2020</i>
Форма обучения	<i>очная</i>
Технология обучения	<i>традиционная</i>

Курс	Семестр	Трудоемкость, з.е.
<i>2</i>	<i>4</i>	<i>3</i>

Вид промежуточной аттестации	Обеспечивающее подразделение
<i>Зачет с оценкой</i>	<i>Кафедра «Проектирование, управление и разработка информационных систем»</i>

**1 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций**

Таблица 1 – Компетенции и индикаторы их достижения

Код по ФГОС	Индикаторы достижения	Планируемые результаты обучения по дисциплине
<b>Универсальные</b>		
<b>Общепрофессиональные</b>		
ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения;	<p>ОПК-8.1 Знает алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения</p> <p>ОПК-8.2 Умеет составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули</p> <p>ОПК-8.3 Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач</p>	<p>Знать типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения</p> <p>Уметь применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов</p> <p>Владеть навыками использования библиотек программных модулей, шаблонов, классов объектов при разработке ПО</p>
<b>Профессиональные</b>		

Таблица 2 – Паспорт фонда оценочных средств

Контролируемые разделы (темы) дисциплины	Формируемая компетенция	Наименование оценочного средства	Показатели оценки
<b>Тема 1. Первый взгляд на объектно-ориентированное программирование.</b> Классы и объекты. Обзор языков и средств разработки.	ОПК-8	Задание 1. Линейные программы Расчетно-графическая работа	Знает возможности существующей программно-технической архитектуры Умеет проводить оценку и обоснование рекомендуемых решений при разработке ПО
<b>Тема 2. Основные понятия языка.</b> Состав языка. Типы данных.	ОПК-8	Задание 1. Линейные программы Задание 2. Разветв-	Владеет приемами разработки компонентов программных комплексов на

		ляющиеся вычислительные процессы Расчетно-графическая работа	языках разного уровня
<b>Тема 3. Переменные, операции и выражения.</b> Переменные. Именованные константы. Операции и выражения. Линейные программы.	ОПК-8	Задание 1. Линейные программы Задание 2. Разветвляющиеся вычислительные процессы Расчетно-графическая работа	Умеет применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов
<b>Тема 4. Операторы.</b> Выражения, блоки и пустые операторы. Операторы ветвления. Операторы цикла. Базовые конструкции структурного программирования. Обработка исключительных ситуаций. Операторы checked и unchecked.	ОПК-8	Задание 1. Линейные программы Задание 2. Разветвляющиеся вычислительные процессы Задание 3. Организация циклов Расчетно-графическая работа	Владеет навыками разработки ПО на базе классов Владеет навыками разработки ПО на базе объектно-ориентированного программирования
<b>Тема 5. Классы: основные понятия.</b> Присваивание и сравнение объектов. Данные: поля и константы. Методы. Ключевое слово this. Конструкторы. Свойства.	ОПК-8	Задание 4. Простейшие классы Расчетно-графическая работа	Владеет навыками разработки ПО на базе классов Владеет навыками разработки ПО на базе объектно-ориентированного программирования
<b>Тема 6. Массивы и строки.</b> Одномерные, двумерные и ступенчатые массивы. Оператор foreach. Массивы объектов. Символы и строки. Вспомогательные классы.	ОПК-8	Задание 5. Одномерные массивы Задание 6. Двумерные массивы Задание 7. Строки Расчетно-графическая работа	Владеет навыками разработки ПО на базе классов Владеет навыками разработки ПО на базе объектно-ориентированного программирования
<b>Тема 7. Классы: подробности.</b> Перегрузка методов. Рекурсивные методы. Методы с переменным количеством аргументов. Метод Main. Индексаторы. Операции класса. Деструкторы. Вложенные типы.	ОПК-8	Задание 8. Классы и операции Расчетно-графическая работа	Умеет применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов
<b>Тема 8. Иерархии классов.</b> Наследова-	ОПК-8	Задание 9. Наследование	Умеет применять методы и средства проектирова-

ние. Виртуальные методы. Раннее и позднее связывание. Абстрактные классы. Бесплодные классы. Класс object.		Расчетно-графическая работа	ния программного обеспечения, структур данных, баз данных, программных интерфейсов
<b>Тема 9. Интерфейсы и структурные типы.</b> Синтаксис интерфейса. Реализация интерфейса. Преобразования типов. Работа с объектами через интерфейсы. Операции is и as. Интерфейсы и наследование. Стандартные интерфейсы. Структуры.	ОПК-8	Задание 10. Структуры Задание 11. Интерфейсы и параметризованные коллекции Расчетно-графическая работа	Знает типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения
<b>Тема 10. Делегаты, события и потоки выполнения.</b> Делегаты. События. Многопоточные приложения.	ОПК-8	Задание 12. Создание Windows-приложений Расчетно-графическая работа	Знает типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения
<b>Тема 11. Работа с файлами.</b> Потоки байтов. Асинхронный ввод-вывод. Потоки символов. Двоичные потоки. Консольный ввод-вывод. Работа с каталогами и файлами. Сохранение объектов (сериализация).	ОПК-8	Задание 7. Строки Расчетно-графическая работа	Знает типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения
<b>Тема 12. Сборки, библиотеки, атрибуты, директивы препроцессора.</b> Сборки. Создание библиотеки. Использование библиотеки. Рефлексия. Атрибуты. Пространства имен. Директивы препроцессора.	ОПК-8	Расчетно-графическая работа	Умеет применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов
<b>Тема 13. Структуры данных, коллекции и классы прототипы.</b> Абстрактные структуры данных. Коллекции. Классы прототипы. Частичные типы. Обнуля-	ОПК-8	Задание 10. Структуры Задание 11. Интерфейсы и параметризованные коллекции Расчетно-	Владеет навыками использования библиотек программных модулей, шаблонов, классов объектов при разработке



емые типы.		графическая работа	
<b>Тема 14. Введение в программирование под Windows.</b> Событийно-управляемое программирование. Шаблон Windows-приложения. Элементы управления. Диалоговые окна. Краткое введение в графику.	ОПК-8	Задание 12. Создание Windows-приложений Расчетно-графическая работа	Владеет навыками использования библиотек программных модулей, шаблонов, классов объектов при разработке
<b>Тема 15. Технологии дополняющие ООП</b> Документирование в формате XML. Язык UML. Паттерны ООП.	ОПК-8	Расчетно-графическая работа	Знает типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения

## 2 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующие процесс формирования компетенций

Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, представлены в виде технологической карты дисциплины (таблица 3).

Таблица 3 – Технологическая карта

	Наименование оценочного средства	Сроки выполнения	Шкала оценивания	Критерии оценивания
4 семестр <i>Промежуточная аттестация в форме зачета с оценкой</i>				
1	Задание 1. Линейные программы	1 неделя	5	0 баллов – задание не выполнено или выполнено не верно 3 балла – задание выполнено с недочетами и не в срок 4 балла – задание выполнено без недочетов и не в срок 5 баллов – задание выполнено без недочетов и в срок
2	Задание 2. Разветвляющиеся вычислительные процессы	2 неделя	5	0 баллов – задание не выполнено или выполнено не верно 3 балла – задание выполнено с недочетами и не в срок 4 балла – задание выполнено без недочетов и не в срок 5 баллов – задание выполнено без недочетов и в срок
3	Задание 3. Организация циклов	3 неделя	5	0 баллов – задание не выполнено или выполнено не верно

	<b>Наименование оценочного средства</b>	<b>Сроки выполнения</b>	<b>Шкала оценивания</b>	<b>Критерии оценивания</b>
				3 балла – задание выполнено с недочетами и не в срок 4 балла – задание выполнено без недочетов и не в срок 5 баллов – задание выполнено без недочетов и в срок
4	Задание 4. Простейшие классы	4 неделя	5	0 баллов – задание не выполнено или выполнено не верно 3 балла – задание выполнено с недочетами и не в срок 4 балла – задание выполнено без недочетов и не в срок 5 баллов – задание выполнено без недочетов и в срок
5	Задание 5. Одномерные массивы	5 неделя	5	0 баллов – задание не выполнено или выполнено не верно 3 балла – задание выполнено с недочетами и не в срок 4 балла – задание выполнено без недочетов и не в срок 5 баллов – задание выполнено без недочетов и в срок
6	Задание 6. Двумерные массивы	6 неделя	5	0 баллов – задание не выполнено или выполнено не верно 3 балла – задание выполнено с недочетами и не в срок 4 балла – задание выполнено без недочетов и не в срок 5 баллов – задание выполнено без недочетов и в срок
7	Задание 7. Строки	7 неделя	8	0 баллов – задание не выполнено или выполнено не верно 3 балла – задание выполнено с недочетами и не в срок 4 балла – задание выполнено без недочетов и не в срок 5 баллов – задание выполнено без недочетов и в срок
8	Задание 8. Классы и операции	8 неделя	5	0 баллов – задание не выполнено или выполнено не верно 3 балла – задание выполнено с недочетами и не в срок 4 балла – задание выполнено без недочетов и не в срок 5 баллов – задание выполнено без недочетов и в срок
9	Задание 9. Наследование	10 неделя	5	0 баллов – задание не выполнено или выполнено не верно 3 балла – задание выполнено с недочетами и не в срок

	Наименование оценочного средства	Сроки выполнения	Шкала оценивания	Критерии оценивания
				тами и не в срок 4 балла – задание выполнено без недочетов и не в срок 5 баллов – задание выполнено без недочетов и в срок
10	Задание 10. Структуры	12 неделя	5	0 баллов – задание не выполнено или выполнено не верно 3 балла – задание выполнено с недочетами и не в срок 4 балла – задание выполнено без недочетов и не в срок 5 баллов – задание выполнено без недочетов и в срок
11	Задание 11. Интерфейсы и параметризованные коллекции	14 неделя	5	0 баллов – задание не выполнено или выполнено не верно 3 балла – задание выполнено с недочетами и не в срок 4 балла – задание выполнено без недочетов и не в срок 5 баллов – задание выполнено без недочетов и в срок
12	Задание 12. Создание Windows-приложений	16 неделя	5	0 баллов – задание не выполнено или выполнено не верно 3 балла – задание выполнено с недочетами и не в срок 4 балла – задание выполнено без недочетов и не в срок 5 баллов – задание выполнено без недочетов и в срок
13	Расчетно-графическая работа	16 неделя	40	5 баллов – задание выполнено с недочетами и не в срок 15 баллов – задание выполнено без недочетов и не в срок 25 баллов – задание выполнено без недочетов и в срок
ИТОГО:		-	100 баллов	-
<p><b>Критерии оценки результатов обучения по дисциплине:</b>  0 – 64 % от максимально возможной суммы баллов – «неудовлетворительно» (недостаточный уровень для промежуточной аттестации по дисциплине);  65 – 74 % от максимально возможной суммы баллов – «удовлетворительно» (пороговый (минимальный) уровень);  75 – 84 % от максимально возможной суммы баллов – «хорошо» (средний уровень);  85 – 100 % от максимально возможной суммы баллов – «отлично» (высокий (максимальный) уровень)</p>				

**3 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие процесс формирования компетенций в ходе освоения образовательной программы**

**3.1 Задания для текущего контроля успеваемости**

**Комплект заданий для лабораторных работ (семестр 4)  
Выполнить задания с использованием актуальной версии языка C#.**

**Задание 1.** Написать программу расчета по двум формулам. Предварительно подготовить тестовые примеры.

1. Вариант 1

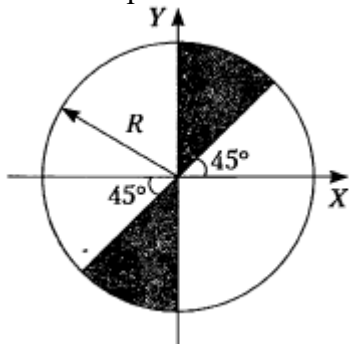
$$z1 = 2 \sin^2(3\pi - 2\alpha) \cos^2(5\pi + 2\alpha), \quad z2 = \frac{1}{4} - \frac{1}{4} \sin\left(\frac{5}{2}\pi - 8\alpha\right).$$

2. Вариант 2

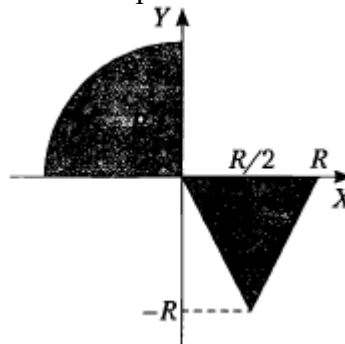
$$z1 = \cos \alpha + \sin \alpha + \cos 3\alpha + \sin 3\alpha, \quad z2 = 2\sqrt{2} \cos \alpha \cdot \sin\left(\frac{\pi}{4} - 2\alpha\right).$$

**Задание 2.** Написать программу, определяющую попадание точки с определенными координатами в заданную плоскую область. Значения координат вводить с клавиатуры. Результат работы программы вывести в виде текстового сообщения

1. Вариант 1

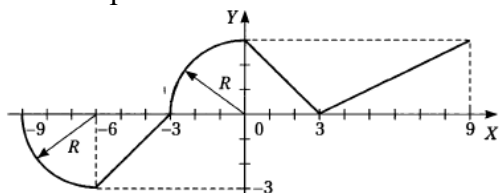


2. Вариант 2

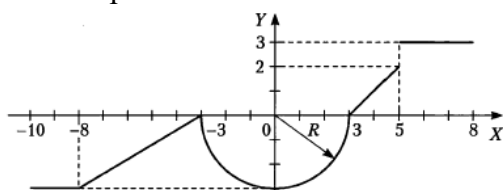


**Задание 3.** Разработать программу, вычисления значения функции, заданной графически. Результаты вывести на экран в виде таблицы на интервале от хнач до хкон с шагом dx. Значения хнач до хкон с шагом dx получить от пользователя.

1. Вариант 1



2. Вариант 2



**Задание 4.** Каждый разрабатываемый класс должен содержать следующие элементы: скрытые поля, конструкторы с параметрами и без параметров, методы, свойства. Методы и свойства должны обеспечивать непротиворечивый, полный, минимальный и удобный интерфейс класса. При возникновении ошибок должны выбрасываться исключения. В программе должна выполняться проверка всех разработанных элементов класса.

1. Вариант 1

Описать класс, реализующий десятичный счетчик, который может увеличивать или уменьшать свое значение на единицу в заданном диапазоне. Предусмотреть инициализацию счетчика значениями по умолчанию и произвольными значениями. Счетчик имеет два метода: увеличения и уменьшения, и свойство, позволяющее получить его текущее состояние. При выходе за границы диапазона выбрасываются исключения. Продемонстрировать применение всех разработанных элементов класса.

2. Вариант 2

Описать класс, реализующий шестнадцатеричный счетчик, который может увеличивать или уменьшать свое значение на единицу в заданном диапазоне. Предусмотреть инициализацию счетчика значениями по умолчанию и произвольными значениями. Счетчик имеет два метода: увеличения и уменьшения, и свойство, позволяющее получить его текущее состояние. При выходе за границы диапазона выбрасываются исключения. Продемонстрировать применение всех разработанных элементов класса.

**Задание 5.** Разработать программу обработки одномерного массива вещественных величин.

1. Вариант 1

1. Найти сумму отрицательных элементов массива.

2. Найти произведение элементов массива, расположенных между максимальным и минимальным элементами.

3. Упорядочить элементы массива по возрастанию.

2. Вариант 2

1. Найти сумму положительных элементов массива.

2. Найти произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.

3. Упорядочить элементы массива по убыванию.

**Задание 6.** Разработать программу обработки двумерного массива.

1. Вариант 1

Дана целочисленная прямоугольная матрица. Определить:

1) количество строк, не содержащих ни одного нулевого элемента;

2) максимальное из чисел, встречающихся в заданной матрице более одного раза.

2. Вариант 2

Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного нулевого элемента.

Характеристикой строки целочисленной матрицы назовем сумму ее положительных четных элементов. Переставляя строки заданной матрицы, располагать их в соответствии с ростом характеристик.

**Задание 7.** Разработать программу обработки текстового файла.

1. Вариант 1

Написать программу, которая считывает из текстового файла три предложения и выводит их в обратном порядке.

2. Вариант 2

Написать программу, которая считывает текст из файла и выводит на экран только

предложения, содержащие введенное с клавиатуры слово.

**Задание 8.** Каждый разрабатываемый класс должен, как правило, содержать следующие элементы: скрытые поля, конструкторы с параметрами и без параметров, методы; свойства, индексаторы; перегруженные операции. Функциональные элементы класса должны обеспечивать непротиворечивый, полный, минимальный и удобный интерфейс класса. При возникновении ошибок должны выбрасываться исключения. В программе должна выполняться проверка всех разработанных элементов класса.

1. Вариант 1

Описать класс для работы с одномерным массивом целых чисел (вектором). Обеспечить следующие возможности:

- задание произвольных целых границ индексов при создании объекта;
- обращение к отдельному элементу массива с контролем выхода за пределы массива;
- выполнение операций поэлементного сложения и вычитания массивов с одинаковыми границами индексов;
- выполнение операций умножения и деления всех элементов массива на скаляр;
- вывод на экран элемента массива по заданному индексу и всего массива.

Написать программу, демонстрирующую все разработанные элементы класса.

2. Вариант 2

Описать класс для работы с одномерным массивом строк фиксированной длины. Обеспечить следующие возможности:

- задание произвольных целых границ индексов при создании объекта;
- обращение к отдельной строке массива по индексу с контролем выхода за пределы массива;
- выполнение операций поэлементного сцепления двух массивов с образованием нового массива;
- выполнение операций слияния двух массивов с исключением повторяющихся элементов;
- вывод на экран элемента массива по заданному индексу и всего массива.

Написать программу, демонстрирующую все разработанные элементы класса.

**Задание 9.** В программах требуется описать базовый класс (возможно, абстрактный), в котором с помощью виртуальных или абстрактных методов и свойств задается интерфейс для производных классов. Целью лабораторной работы является максимальное использование наследования, даже если для конкретной задачи оно не дает выигрыша в объеме программы. Во всех классах следует переопределить метод Equals, чтобы обеспечить сравнение значений, а не ссылок.

Функция Main должна содержать массив из элементов базового класса, заполненный ссылками на производные классы. В этой функции должно демонстрироваться использование всех разработанных элементов классов.

1. Вариант 1

Создать класс Point (точка). На его основе создать классы ColoredPoint и Line (линия). На основе класса Line создать классы ColoredLine и PolyLine (многоугольник). В классах описать следующие элементы:

- конструкторы с параметрами и конструкторы по умолчанию;
- свойства для установки и получения значений всех координат, а также для изменения цвета и получения текущего цвета;
- для линий—методы изменения угла поворота линий относительно первой точки;
- для многоугольника—метод масштабирования.

2. Вариант 2

Создать абстрактный класс Vehicle (транспортное средство). На его основе реализовать классы Plane (самолет), Car (автомобиль) и Ship (корабль). Классы должны иметь возможность задавать и получать координаты и параметры средств передвижения (цена, скорость, год выпуска и т. п.) с помощью свойств. Для самолета должна быть определена высота, для самолета и корабля – количество пассажиров, для корабля – порт приписки. Динамические характеристики задать с помощью методов.

**Задание 10.** Описать структуру с заданным именем и полями. Разработать программу, обрабатывающую массив таких структур указанным способом.

1. Вариант 1

Описать структуру с именем STUDENT, содержащую следующие поля: фамилия и инициалы; номер группы; успеваемость (массив из пяти элементов). Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из пяти структур типа STUDENT; записи должны быть упорядочены по возрастанию номера группы;
- вывод на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4.0;
- если таких студентов нет, вывести соответствующее сообщение.

2. Вариант 2

Описать структуру с именем STUDENT, содержащую следующие поля: фамилия и инициалы; номер группы; успеваемость (массив из пяти элементов). Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из пяти структур типа STUDENT; записи должны быть упорядочены по возрастанию среднего балла;
- вывод на дисплей фамилий и номеров групп для всех студентов, имеющих оценки и 5;
- если таких студентов нет, вывести соответствующее сообщение.

**Задание 11.** Выполнить задания 9, используя для хранения экземпляров разработанных классов стандартные параметризованные коллекции. Во всех классах реализовать интерфейс Comparable и перегрузить операции отношения для реализации значимой семантики сравнения объектов по какому-либо полю на усмотрение студента.

**Задание 12.** Написать Windows-приложение, заголовок главного окна которого содержит Ф. И. О., группу студента и номер варианта. В программе должна быть предусмотрена обработка исключений, возникающих из-за ошибочного ввода.

1. Вариант 1

Создать меню с командами Input, Calc и Exit.

При выборе команды Input открывается диалоговое окно, содержащее:

- три поля типа Text Box для ввода длин трех сторон треугольника;
- группу из двух флажков (Периметр и Площадь) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность:

- ввода длин трех сторон треугольника;
- выбора режима с помощью флажков: подсчет периметра и/или площади треугольника

При выборе команды Calc открывается диалоговое окно с результатами. При выборе команды Exit приложение завершается.

2. Вариант 2

Создать меню с командами Size, Color, Paint, Quit.

Команда Paint недоступна. При выборе команды Quit приложение завершается.

При выборе команды Size открывается диалоговое окно, содержащее:

- два поля типа Text Box для ввода длин сторон прямоугольника;
- группу из трех флажков (Red, Green, Blue) типа CheckBox;
- кнопку типа Button.

Обеспечить возможность:

- ввода длин сторон прямоугольника в пикселах в поля ввода;
- выбора его цвета с помощью флажков.

После задания параметров команда Paint становится доступной.

При выборе команды Paint в главном окне приложения выводится прямоугольник заданного размера и сочетания цветов или выдается сообщение, если введенные размеры превышают размер окна.

## Задания для расчетно-графической работы (семестр 4)

1. Выполнить верификацию и тестирование кода полученного в ходе текущей аттестации при выполнении индивидуальных заданий путем проведения тестовых запусков программ на заранее подготовленных наборах данных.

2. Описать решения индивидуальных заданий полученные в ходе прохождения текущей аттестации в соответствии с требованиями единой системы программной документации (ЕСПД) и РД 013-2016 «Текстовые студенческие работы. Правила оформления».

Структурными элементами данной контрольной работы должны быть:

- титульный лист;
- текст всех заданий (в соответствии с вариантом);
- содержание;
- введение
- основная часть;
- оценка результатов проектирования;
- заключение;
- список использованных источников;
- приложения.

Во введении дается краткое описание изучаемой дисциплины, которой посвящена данная работа, а также приводится обзор выполненной работы.

Каждое выполненное индивидуальное задание оформляется отдельным разделом основной части отчета. Каждый раздел включает теоретические сведения по теме раздела с обязательным указанием ссылок на источники информации, в случае её заимствования. Обязательными подразделами каждого отчета являются «Текст программы», «Описание программы», «Программа и методика испытаний». Требования к содержанию подразделов приведены в таблице:

<b>Вид программного документа</b>	<b>Содержание программного документа</b>
Описание программы	Сведения о логической структуре и функционировании программы
Текст программы	Запись программы с необходимыми комментариями
Программа и методика испытаний	Требования, подлежащие проверке при испытании программы, а также порядок и методы их контроля

В заключении приводятся общие выводы и предложения, а также краткое описание проделанной работы; даются практические рекомендации. (1,5 - 2 страницы).



Список литературы состоит из нормативно-правовых актов, учебников и учебных пособий, научных статей, использованных в ходе выполнения индивидуального задания.

Приложения помещают после списка литературы в порядке их отсылки или обращения к ним в тексте. В качестве приложений рекомендуется предоставлять копии документов, бланков договоров, организационно-распорядительных документов, аналитических таблиц, иных документов, иллюстрирующих содержание основной части.

Отчет предварительно оценивается и допускается к защите после проверки его соответствия требованиям, предъявляемым данными методическими указаниями. Защита отчетов организуется в форме собеседования.

## Приложение А

Задания для организации «входного» контроля знаний учащихся.

1. Что из указанного ниже является элементами программ:
  - а) операторы;
  - б) компиляторы;
  - в) переменные;
  - г) производные.
  
2. Зачем нужен тип данных
  - а) для определения допустимых операций;
  - б) для обеспечения целостности данных;
  - в) для создания точек останова программ;
  - г) для указания переменной типа ее содержимого.
  
3. Укажите, что является результатом выполнения оператора присваивания:
  - а) изменение значения переменной на заданное значение;
  - б) изменение значения константы на вычисленное значение;
  - в) передача управления подпрограмме;
  - г) проверка на равенство двух величин.
  
4. Для чего предназначен оператор цикла:
  - а) для передачи управления подпрограмме;
  - б) для продолжения работы программы после останова;
  - в) для организации многократного выполнения группы операторов;
  - г) для выбора одного из нескольких вариантов дальнейших действий.
  
5. Что такое массив
  - а) именованный набор переменных, имеющих различные типы данных, и располагающихся в одной области памяти;
  - б) переменный набор имен, имеющий один тип и разное место в памяти;
  - в) набор переменных имен и функций, которые располагаются в одной области памяти;
  - г) именованный набор переменных, имеющий один тип данных, и располагающихся в одной области памяти.
  
6. Что такое функция
  - а) некоторая часть программы имеющая собственное имя и которое может быть вызвано необходимое количество раз;
  - б) некоторая часть программы содержащая вредоносный код, и блокирующая определенные действия системы;
  - в) часть программы, в которой хранятся глобальные переменные;
  - г) некоторое действие, не имеющее возвращаемого результата.
  
7. В чем состоит основное назначение модульного программирования:
  - а) сокращение размеров программ;
  - б) упрощение решения задачи за счет разбиения её на подзадачи;
  - в) ускорение выполнения вычислений;
  - г) создание библиотек полезных подпрограмм.

### Лист регистрации изменений к РПД

	Номер протокола заседания кафедры, дата утверждения изменения	Количество страниц изменения	Подпись разработчика РПД