

**Методическое пособие по дисциплине**  
**«Объектно-ориентированное программирование»**  
**ООП «Инженерия информационных и интеллектуальных систем»**  
**по направлению 09.03.04 – Программная инженерия**

**Фонд оценочных средств**

№ КМ	Вид КМ	Наименование КМ	Оценочные средства
1	Текущий контроль	Контрольная работа 1	<p>Какая предшествующая парадигма программирования оказала сильное влияние на становление объектно-ориентированной парадигмы?</p> <p>a. Структурная парадигма программирования  b. Парадигма обобщенного программирования  c. Парадигма логического программирования  d. Аспектно-ориентированная парадигма  e. Объектно-ориентированная парадигма была первой парадигмой программирования в истории</p> <p>Среди перечисленных конструкций C# укажите объявление свойства</p> <p>a. <code>string GetName() {return "Name";}</code>  b. <code>string Name;</code>  c. <code>string Name {get{return "Name";}}</code>  d. <code>string this[int i] {get{return "Name";}}</code></p> <p>Динамический метод можно вызвать только в контексте объекта (экземпляра класса)</p> <p>a. Верно  b. Неверно</p>
2	Текущий контроль	Проверка практического задания 1	<p>Практическое задание 1.</p> <p>Создать структуру с именем <code>time</code>. Три её поля, имеющие тип <code>int</code>, будут называться <code>hours</code>, <code>minutes</code> и <code>seconds</code>. Написать программу, которая просит пользователя ввести время в формате часы, минуты, секунды. Программа должна хранить время в структурной переменной типа <code>time</code> и выводить количество секунд в введенном времени.</p> <p>Создать перечисление с именем <code>pets</code> и значениями <code>dog</code>, <code>cat</code>, <code>rat</code>, <code>fish</code>, <code>bird</code>. Создать структуру с именем <code>animal</code>. Определить её поля: <code>name</code> как массив из 20 символов типа <code>char</code>, <code>type</code> как типа <code>pets</code> и <code>age</code> типа <code>float</code>, хранящие, соответственно, имя животного, его разновидность и возраст. Написать программу, которая создает 4 переменные типа <code>animal</code> и просит пользователя ввести значения разновидности, имени и возраста всех 4 животных. Причем разновидность животного вводится по первому символу, набранному на клавиатуре (т.е. если нажата <code>d</code>, то в соответствующее поле вводится значение <code>dog</code>). В качестве результата программа должна выдать список всех имеющихся животных со всеми значениями полей</p>

			<p>Примеры контрольных вопросов</p> <ol style="list-style-type: none"> <li>1. Какой синтаксис описания структуры?</li> <li>2. Как распределяется память в структуре?</li> <li>3. Как можно обеспечить доступ к элементу структуры?</li> <li>4. Как описать переменную структурированного типа?</li> <li>5. Как описать в структуре переменную структурированного типа?</li> <li>6. Что такое вложенные структуры?</li> <li>7. Как определить размер структуры?</li> <li>8. Что такое указатели на структуру?</li> </ol>
3	Текущий контроль	Проверка практического задания 2	<p>Практическое задание 2.</p> <p>Создать класс с именем <code>Time</code>, содержащий три поля типа <code>int</code>, предназначенных для хранения часов, минут и секунд. Один из конструкторов класса должен инициализировать поля нулевыми значениями, а другой – заданным набором значений. Создать метод класса, который будет выводить значения полей на экран, в формате <code>23:59:59</code> и метод, складывающий значения двух объектов типа <code>Time</code>, передаваемых ему в качестве аргументов. Продемонстрировать работу класса.</p> <p>Примеры контрольных вопросов</p> <ol style="list-style-type: none"> <li>1. Что такое класс?</li> <li>2. Что такое объект?</li> <li>3. Что такое метод класса?</li> <li>4. Что такое атрибут класса?</li> <li>5. Что такое модификаторы видимости? какие они бывают?</li> </ol>
4	Текущий контроль	Проверка практического задания 3	<p>Практическое задание 3.</p> <p>Создать класс с именем <code>fraction</code>, содержащий два поля типа <code>int</code> – числитель и знаменатель обыкновенной дроби. Конструктор класса должен инициализировать их заданным набором значений. Создать метод класса, который будет выводить дробь на экран в формате <code>x / y</code>, метод, складывающий две дроби, переданные ему в параметрах и метод, умножающий две дроби, переданные ему в параметрах.</p> <p>Примеры контрольных вопросов</p> <ol style="list-style-type: none"> <li>1. Что такое конструктор класса? когда он вызывается?</li> <li>2. Что такое деструктор класса? когда он вызывается?</li> <li>3. Передача и возврат параметров в функции по значению, по ссылке, по указателю.</li> <li>4. Статические методы и поля.</li> </ol>
5	Текущий контроль	Контрольная работа 2	<p>Для каких элементов класса справедливо утверждение: Чем больше в классе этих элементов, тем больше места в памяти занимает каждый экземпляр этого класса (выберите один или несколько правильных вариантов ответа)</p> <ol style="list-style-type: none"> <li>a. <code>string name;</code></li> <li>b. <code>string GetName() {return "Name";}</code></li> <li>c. <code>static string path;</code></li> <li>d. <code>static string GetPath() {return path;}</code></li> </ol>

			<p>Отметьте все корректные обращения к полям объявленного класса SomeClass: class SomeClass</p> <pre>{     public static int s;     public int d; }</pre> <p>a. SomeClass.s = 42  b. SomeClass.d = 42  c. new SomeClass().s = 42  d. new SomeClass().d = 42</p>
6	Текущий контроль	Проверка практического задания 4	<p>Практическое задание 4.  Описать класс fraction, у которого поля x и y задают числитель и знаменатель обыкновенной дроби. Перегрузить для этого класса арифметические операции сложения, вычитания, умножения и деления так, чтобы они могли оперировать как с объектами класса, так и с числами (то есть выполнять, например, не только действие <math>3/4 + 2/5</math>, но и <math>1/2 + 4</math> или <math>2 * 5/6</math>). Также перегрузить операции сравнения <code>==</code> и <code>!=</code>. Продемонстрировать работу класса.</p> <p>Примеры контрольных вопросов</p> <ol style="list-style-type: none"> <li>1. Что такое перегрузка метода?</li> <li>2. Что такое полиморфизм?</li> <li>3. Как реализуется полиморфизм?</li> <li>4. Особенности перегрузки и возврата значений в операторах <code>&lt;=&gt;</code> и <code>&lt;+&gt;</code>. Перегрузка оператора <code>&lt;+==&gt;</code>.</li> <li>5. Перегрузка оператора приведения типа.</li> </ol>
7	Текущий контроль	Проверка практического задания 5	<p>Практическое задание 5.  Определить класс vector2 как вектор на плоскости с данными x и y. Определить для него операции сложения, вычитания и скалярного произведения, определить функцию присваивания значения координатам вектора и функцию вывода значений на консоль. Определить класс vector3 как вектор в пространстве, породив его от класса vector2. Переопределить для него функции и операции. Продемонстрировать работу класса.</p> <p>Примеры контрольных вопросов</p> <ol style="list-style-type: none"> <li>1. Что такое наследование?</li> <li>2. Что такое простое наследование?</li> <li>3. Что такое многоуровневое наследование?</li> <li>4. Что такое множественное наследование?</li> <li>5. Когда используется множественное наследование?</li> <li>6. Когда используется многоуровневое наследование?</li> <li>7. Какие члены класса может наследовать другой класс?</li> <li>8. Какова разница между базовым классом и производным классом?</li> </ol>
8	Текущий контроль	Проверка практического задания 6	<p>Практическое задание 6.  Создать абстрактный класс Figure с виртуальными методами вычисления площади и периметра. Создать</p>

			<p>производные классы: Rectangle (прямоугольник), Circle (круг), Triangle (треугольник). Описать в производных классах функции вычисления периметра и площади, продемонстрировать работу механизма виртуальных функций.</p> <p>Примеры контрольных вопросов</p> <ol style="list-style-type: none"> <li>1. Что такое Абстрактный класс.</li> <li>2. Что такое виртуальная функция?</li> <li>3. Проиллюстрировать разницу в работе обычного и виртуального методов.</li> <li>4. Что такое интерфейс?</li> <li>5. Как полиморфизм позволяет реализовать интерфейсы?</li> <li>6. Что такое перегрузка метода?</li> </ol>
9	Текущий контроль	Проверка практического задания 7	<p>7. Создать шаблон функции, возвращающей среднее арифметическое всех элементов массива. Аргументами функции должны быть имя и размер массива (типа int). Создать шаблон функции, возвращающей значение максимального элемента массива. Аргументами функции должны быть имя и размер массива (типа int). Создать шаблон функции, обменивающей местами значения двух передаваемых ей по ссылке аргументов. Создать шаблон функции, осуществляющей сортировку данных массива. Аргументами функции должны быть имя и размер массива (типа int). Продемонстрировать работу шаблонов на данных различных типов, в том числе, на данных вновь созданного класса – вектор на плоскости, определив для него операции сравнения.</p> <p>Примеры контрольных вопросов</p> <ol style="list-style-type: none"> <li>1. Обобщенное программирование. Шаблоны функций.</li> <li>2. Обобщенное программирование. Шаблоны классов.</li> <li>3. Библиотека стандартных шаблонов (STL)</li> </ol>
11	Промежуточная аттестация	Тест	<ol style="list-style-type: none"> <li>1. Эволюция методологий программирования. Парадигмы программирования.</li> <li>2. Основные принципы объектно-ориентированного программирования: абстрагирование, инкапсуляция, модульность, иерархия, типизация, параллелизм, сохраняемость.</li> <li>3. Объект с точки зрения ООП. Состояние. Поведение. Идентичность и жизненный цикл объектов. Взаимоотношения между объектами.</li> <li>4. Понятие объектно-ориентированного проектирования.</li> <li>5. Язык UML (понятие, определение, мотивация применения, типы основных диаграмм).</li> <li>6. Язык UML. Диаграмма вариантов использования. Документ «Поток событий».</li> <li>7. Язык UML. Диаграмма последовательности.</li> <li>8. Язык UML. Кооперативная диаграмма.</li> <li>9. Язык UML. Отношения между классами (обобщения, ассоциации, зависимости и реализации).</li> <li>10. Язык UML. Диаграмма классов. Диаграмма пакетов.</li> <li>11. Язык UML. Диаграмма состояний. Диаграмма деятельности.</li> </ol>

		<p>12.Язык UML. Диаграмма компонентов. Диаграмма размещения.</p> <p>13.Объявление класса. Методы inline.</p> <p>14.Конструктор (значение, свойства, перегрузка, вызов).</p> <p>15.Конструктор копирования.</p> <p>16.Статические поля и методы класса.</p> <p>17.Деструктор.</p> <p>18.Дружественные функции и классы.</p> <p>19.Перегруженные операторы и методы класса.</p> <p>20.Простое наследование. Механизм наследования.</p> <p>21.Переопределение методов и виртуальные функции.</p> <p>22.Многократное наследование.</p> <p>23.Конструкторы и деструкторы при наследовании.</p> <p>24.Чисто виртуальные методы и абстрактные классы.</p> <p>25.Шаблоны классов. Типы по умолчанию и явная специализация обобщённого класса.</p> <p>26.Разделение объектно-ориентированной программы на потоки. Изучение способов синхронизации потоков в C++.</p> <p>27.Обработка исключений в C++(механизм, синтаксис, перехват).</p> <p>28.Исключения в конструкторах и деструкторах.</p> <p>29.Классы стандартной библиотеки C++. Поточные классы. Стандартные потоки. Форматирование данных.</p> <p>30.Поточные классы: ошибки потоков.</p> <p>31.Поточные классы : файловые потоки (конструкторы, режимы открытия, закрытие).</p> <p>32.Класс string стандартной библиотеки (конструкторы, операции, функции).</p> <p>33.Контейнеры. Виды контейнеров. Итераторы</p>
--	--	---

### *Паспорт фонда оценочных средств*

Компетенции	Результаты обучения	№ КМ													
		1	2	3	4	5	6	7	8	9	10	11	12	13	
ОПК-2	Знает: основные возможности современных интегрированных сред разработки программного обеспечения на объектно-ориентированных языках программирования; возможности компиляторов программных проектов под различные операционные системы; наборы инструкций для системных утилит автоматической сборки программного обеспечения и установки программных пакетов объектно-ориентированных библиотек и фреймворков	+		+		++		+				+		+	+

Компетенции	Результаты обучения	№ КМ											
ОПК-2	Умеет: использовать функциональные возможности современных интегрированных сред разработки программного обеспечения на объектно-ориентированных языках программирования для разработки прикладных программ; использовать утилиты автоматической сборки и развертывания программ в операционных системах	+	+	+	+	+	+	+	+	+	+		+
ОПК-2	Имеет практический опыт: работы с основными современными интегрированными средами разработки программного обеспечения на объектно-ориентированных языках, разработки, отладки и развёртывания программного обеспечения в операционных системах семейства Windows и Linux			+	+		+	+	+				+
ОПК-6	Знает: методы разработки алгоритмов и программ в рамках объектно-ориентированной парадигмы программирования на современном языке высокого уровня; принципы объектно-ориентированной парадигмы: абстрагирование, инкапсуляция, наследование, полиморфизм; основные синтаксические конструкции объектно-ориентированного языка программирования: классы, поля, свойства, методы, выражения, события; методы обобщенного программирования; методы оценки сложности алгоритмов; функциональные возможности стандартной библиотеки языка и фреймворка	+	+		+	+		+	+	+	+	+	+
ОПК-6	Умеет: разрабатывать алгоритмы и программы в рамках объектно-ориентированной парадигмы на современном языке программирования высокого уровня с применением основных синтаксических конструкций и функциональных возможностей стандартной библиотеки языка и фреймворка	+	+	+	+	+	+	+	+	+	+		+
ОПК-6	Имеет практический опыт: разработки алгоритмов и программ, отладки, поиска и устранения ошибок программного кода, оценки сложности алгоритмов, использования возможностей стандартной библиотеки, сторонних библиотек программного кода и фреймворков	+	+		+	+	+	+	+				+

Компетенции	Результаты обучения	№ КМ														
ПК-4	Знает: современные языки программирования, библиотеки и программные платформы для объектно-ориентированного программирования приложений (C++, C#)	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
ПК-4	Умеет: разрабатывать программные приложения с использованием современных языков программирования, библиотек и программных платформ объектно-ориентированного программирования (C++, C#)	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
ПК-4	Имеет практический опыт: создания сложных программных систем с применением принципов ООП	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

### ***Контрольная работа 1***

Контрольная работа проводится в форме тестирования. В тесте 3 вопроса. На прохождение теста дается одна попытка. Ограничение по времени - 9 минут. Тест доступен только во время лекции. Система тестирования автоматически рассчитывает долю правильных ответов и выставляет соответствующий балл.

### **Вопросы**

1. Какая предшествующая парадигма программирования оказала сильное влияние на становление объектно-ориентированной парадигмы?
  - a. Структурная парадигма программирования
  - b. Парадигма обобщенного программирования
  - c. Парадигма логического программирования
  - d. Аспектно-ориентированная парадигма
  - e. Объектно-ориентированная парадигма была первой парадигмой программирования в истории
  
2. Среди перечисленных конструкций C# укажите объявление свойства
  - a. `string GetName() {return "Name";}`
  - b. `string Name;`
  - c. `string Name {get{return "Name";}}`

d. `string this[int i] {get{return "Name";}}`;

3. Динамический метод можно вызвать только в контексте объекта (экземпляра класса)

- a. Верно
- b. Неверно

### ***Контрольная работа 2***

Контрольная работа проводится в форме тестирования. В тесте 3 вопроса. На прохождение теста дается одна попытка. Ограничение по времени - 9 минут. Тест доступен только во время лекции. Система тестирования автоматически рассчитывает долю правильных ответов и выставляет соответствующий балл.

### Вопросы

1. Для каких элементов класса справедливо утверждение: Чем больше в классе этих элементов, тем больше места в памяти занимает каждый экземпляр этого класса (выберите один или несколько правильных вариантов ответа)

- a. `string name;`
- b. `string GetName() {return "Name";}`
- c. `static string path;`
- d. `static string GetPath() {return path;}`

2. Отметьте все корректные обращения к полям объявленного класса

```
SomeClass: class SomeClass
{
    public static int s;
    public int d;
}
```

- a. `SomeClass.s = 42`
- b. `SomeClass.d = 42`
- c. `new SomeClass().s = 42`
- d. `new SomeClass().d = 42`

### ***Практическая работа 1***

Для прохождения собеседования по практической работе обучающийся предоставляет отчет. В процессе собеседования обучающийся должен свободно ориентироваться в отчете, сопоставлять теорию и полученные практические результаты, отвечать на 3 дополнительных вопроса. При прохождении собеседования обучающийся получает 1 балл за каждый правильный ответ.

Задание 1. Создать структуру с именем `time`. Три её поля, имеющие тип `int`, будут называться `hours`, `minutes` и `seconds`. Написать программу, которая просит пользователя ввести время в формате часы, минуты, секунды. Программа должна хранить время в структурной переменной типа `time` и выводить количество секунд в введенном времени.

Задание 2. Создать перечисление с именем `pets` и значениями `dog`, `cat`, `rat`, `fish`, `bird`. Создать структуру с именем `animal`. Определить её поля: `name` как массив из 20 символов типа `char`, `type` как типа `pets` и `age` типа `float`, хранящие, соответственно, имя животного, его разновидность и возраст. Написать программу, которая создает 4 переменные типа `animal` и просит пользователя ввести значения разновидности, имени и возраста всех 4 животных. Причем разновидность животного вводится по первому символу, набранному на клавиатуре (т.е. если нажата `d`, то в соответствующее поле вводится значение `dog`). В качестве результата программа должна выдать список всех имеющихся животных со всеми значениями полей

## Примеры контрольных вопросов

1. Какой синтаксис описания структуры?
2. Как распределяется память в структуре?
3. Как можно обеспечить доступ к элементу структуры?
4. Как описать переменную структурированного типа?
5. Как описать в структуре переменную структурированного типа?
6. Что такое вложенные структуры?
7. Как определить размер структуры?
8. Что такое указатели на структуру?

## ***Практическая работа 2***

Для прохождения собеседования по практической работе обучающийся предоставляет отчет. В процессе собеседования обучающийся должен свободно ориентироваться в отчете, сопоставлять теорию и полученные практические результаты, отвечать на 3 дополнительных вопроса. При прохождении собеседования обучающийся получает 1 балл за каждый правильный ответ.

Задание. Создать класс с именем Time, содержащий три поля типа int, предназначенных для хранения часов, минут и секунд. Один из конструкторов класса должен инициализировать поля нулевыми значениями, а другой – заданным набором значений. Создать метод класса, который будет выводить значения полей на экран, в формате 23:59:59 и метод, складывающий значения двух объектов типа Time, передаваемых ему в качестве аргументов. Продемонстрировать работу класса.

## Примеры контрольных вопросов

1. Что такое класс?
2. Что такое объект?
3. Что такое метод класса?
4. Что такое атрибут класса?

## 5. Что такое модификаторы видимости?

### ***Практическая работа 3***

Для прохождения собеседования по практической работе обучающийся предоставляет отчет. В процессе собеседования обучающийся должен свободно ориентироваться в отчете, сопоставлять теорию и полученные практические результаты, отвечать на 3 дополнительных вопроса. При прохождении собеседования обучающийся получает 1 балл за каждый правильный ответ.

Задание. Создать класс с именем `fraction`, содержащий два поля типа `int` – числитель и знаменатель обыкновенной дроби. Конструктор класса должен инициализировать их заданным набором значений. Создать метод класса, который будет выводить дробь на экран в формате `x / y`, метод, складывающий две дроби, переданные ему в параметрах и метод, умножающий две дроби, переданные ему в параметрах.

### Примеры контрольных вопросов

1. Что такое конструктор класса? когда он вызывается?
2. Что такое деструктор класса? когда он вызывается?
3. Передача и возврат параметров в функции по значению, по ссылке, по указателю.
4. Статические методы и поля.

### ***Практическая работа 4***

Для прохождения собеседования по практической работе обучающийся предоставляет отчет. В процессе собеседования обучающийся должен свободно ориентироваться в отчете, сопоставлять теорию и полученные практические результаты, отвечать на 3 дополнительных вопроса. При прохождении собеседования обучающийся получает 1 балл за каждый правильный ответ.

### Задание.

Описать класс `fraction`, у которого поля `x` и `y` задают числитель и знаменатель обыкновенной дроби. Перегрузить для этого класса арифметические операции сложения, вычитания, умножения и деления так, чтобы они могли оперировать как с объектами класса, так и с числами (то есть выполнять, например, не только действие  $3/4 + 2/5$ , но и  $1/2 + 4$  или  $2 * 5/6$ ). Также перегрузить операции сравнения `==` и `!=`. Продемонстрировать работу класса.

### Примеры контрольных вопросов

1. Что такое перегрузка метода?
2. Что такое полиморфизм?
3. Как реализуется полиморфизм?
4. Особенности перегрузки и возврата значений в операторах «`==`» и «`+`».  
Перегрузка оператора «`+=`».
5. Перегрузка оператора приведения типа.

### ***Практическая работа 5***

Для прохождения собеседования по практической работе обучающийся предоставляет отчет. В процессе собеседования обучающийся должен свободно ориентироваться в отчете, сопоставлять теорию и полученные практические результаты, отвечать на 3 дополнительных вопроса. При прохождении собеседования обучающийся получает 1 балл за каждый правильный ответ.

### Задание

Определить класс `vector2` как вектор на плоскости с данными `x` и `y`.  
Определить для него операции сложения, вычитания и скалярного произведения, определить функцию присваивания значения координатам вектора и функцию вывода значений на консоль. Определить класс `vector3`

как вектор в пространстве, породив его от класса vector2. Переопределить для него функции и операции. Продемонстрировать работу класса.

### Примеры контрольных вопросов

1. Что такое наследование?
2. Что такое простое наследование?
3. Что такое многоуровневое наследование?
4. Что такое множественное наследование?
5. Когда используется множественное наследование?
6. Когда используется многоуровневое наследование?
7. Какие члены класса может наследовать другой класс?
8. Какова разница между базовым классом и производным классом?

### ***Практическая работа 6***

Для прохождения собеседования по практической работе обучающийся предоставляет отчет. В процессе собеседования обучающийся должен свободно ориентироваться в отчете, сопоставлять теорию и полученные практические результаты, отвечать на 3 дополнительных вопроса. При прохождении собеседования обучающийся получает 1 балл за каждый правильный ответ.

### Задание

Создать абстрактный класс Figure с виртуальными методами вычисления площади и периметра. Создать производные классы: Rectangle (прямоугольник), Circle (круг), Triangle (треугольник). Описать в производных классах функции вычисления периметра и площади, продемонстрировать работу механизма виртуальных функций.

### Примеры контрольных вопросов

1. Что такое Абстрактный класс.
2. Что такое виртуальная функция?

3. Проиллюстрировать разницу в работе обычного и виртуального методов.
4. Что такое интерфейс?
5. Как полиморфизм позволяет реализовать интерфейсы?
6. Что такое перегрузка метода?

### ***Практическая работа 7***

Для прохождения собеседования по практической работе обучающийся предоставляет отчет. В процессе собеседования обучающийся должен свободно ориентироваться в отчете, сопоставлять теорию и полученные практические результаты, отвечать на 3 дополнительных вопроса. При прохождении собеседования обучающийся получает 1 балл за каждый правильный ответ.

#### Задание

Создать шаблон функции, возвращающей среднее арифметическое всех элементов массива. Аргументами функции должны быть имя и размер массива (типа `int`). Создать шаблон функции, возвращающей значение максимального элемента массива. Аргументами функции должны быть имя и размер массива (типа `int`). Создать шаблон функции, обменивающей местами значения двух передаваемых ей по ссылке аргументов. Создать шаблон функции, осуществляющей сортировку данных массива. Аргументами функции должны быть имя и размер массива (типа `int`). Продемонстрировать работу шаблонов на данных различных типов, в том числе, на данных вновь созданного класса – вектор на плоскости, определив для него операции сравнения.

#### Примеры контрольных вопросов

1. Обобщенное программирование. Шаблоны функций.
2. Обобщенное программирование. Шаблоны классов.
3. Библиотека стандартных шаблонов (STL)

## ***Практическая работа 8***

Для прохождения собеседования по практической работе обучающийся предоставляет отчет. В процессе собеседования обучающийся должен свободно ориентироваться в отчете, сопоставлять теорию и полученные практические результаты, отвечать на 3 дополнительных вопроса. При прохождении собеседования обучающийся получает 1 балл за каждый правильный ответ.

### Задание

Создать шаблон функции, возвращающей среднее арифметическое всех элементов массива. Аргументами функции должны быть имя и размер массива (типа `int`). Создать шаблон функции, возвращающей значение максимального элемента массива. Аргументами функции должны быть имя и размер массива (типа `int`). Создать шаблон функции, обменивающей местами значения двух передаваемых ей по ссылке аргументов. Создать шаблон функции, осуществляющей сортировку данных массива. Аргументами функции должны быть имя и размер массива (типа `int`). Продемонстрировать работу шаблонов на данных различных типов, в том числе, на данных вновь созданного класса – вектор на плоскости, определив для него операции сравнения.

### Примеры контрольных вопросов

1. Работа с массивами
2. Шаблоны классов.
3. Библиотека стандартных шаблонов (STL)

### ***Итоговый тест***

№	Вопрос	Варианты ответа
1.	Какой вид отношения между классами содержит семантику "является" ("is-a")	a. Обобщение b. Ассоциация c. Композиция d. Агрегация e. Зависимость

№	Вопрос	Варианты ответа
2.	Выберите виды отношений, которые имеют семантику "Часть - Целое"	<ul style="list-style-type: none"> <li>a. Агрегация</li> <li>b. Композиция</li> <li>c. Обобщение</li> <li>d. Реализация</li> <li>e. Ассоциация</li> <li>f. Зависимость</li> </ul>
3.	Какие диаграммы UML предназначены для описания взаимодействия объектов системы во времени?	<ul style="list-style-type: none"> <li>a. Диаграммы кооперации</li> <li>b. Диаграммы последовательностей</li> <li>c. Диаграмма деятельности</li> <li>d. Диаграмма классов</li> <li>e. Диаграмма объектов</li> <li>f. Диаграмма прецедентов</li> </ul>
4.	Какой вид диаграммы на UML аналогичен блок-схеме алгоритма?	<ul style="list-style-type: none"> <li>a. диаграмма деятельности</li> <li>b. диаграмма взаимодействия</li> <li>c. диаграмма последовательности</li> <li>d. диаграмма классов</li> <li>e. диаграмма прецедентов</li> </ul>
5.	Выберите диаграммы, которые используются в UML для описания поведения системы (динамических аспектов)	<ul style="list-style-type: none"> <li>a. диаграммы взаимодействия</li> <li>b. диаграммы состояний</li> <li>c. диаграммы прецедентов (вариантов использования)</li> <li>d. диаграммы деятельности</li> <li>e. диаграммы классов</li> <li>f. диаграммы развертывания</li> <li>g. диаграммы компонентов</li> </ul>
6.	Какой вид диаграммы акцентирует внимание на временной упорядоченности сообщений, использует элементы "фокус управления" и "линия жизни объекта" и дает наглядную картину, позволяющую понять развитие потока управления во времени?	<ul style="list-style-type: none"> <li>a. Диаграмма прецедентов</li> <li>b. Диаграмма деятельности</li> <li>c. Диаграмма состояний</li> <li>d. Диаграмма кооперации</li> <li>e. Диаграмма последовательностей</li> </ul>
7.	Какие отношения могут быть между акторами на диаграмме прецедентов?	<ul style="list-style-type: none"> <li>a. Обобщения</li> <li>b. Ассоциации</li> <li>c. Включения (include)</li> <li>d. Расширения (extends)</li> <li>e. Агрегации</li> <li>f. Реализации</li> </ul>
8.	Какие отношения могут быть между прецедентами на диаграмме прецедентов?	<ul style="list-style-type: none"> <li>a. Обобщения</li> <li>b. Ассоциации</li> <li>c. Включения (include)</li> <li>d. Расширения (extends)</li> <li>e. Агрегации</li> <li>f. Зависимости</li> </ul>
9.	Какие элементы отображаются на диаграмме сценариев использования (Use cases)	<ul style="list-style-type: none"> <li>a. Актор</li> <li>b. Прецедент</li> <li>c. Класс</li> <li>d. Интерфейс</li> <li>e. Линии жизни объекта (object life line)</li> </ul>

№	Вопрос	Варианты ответа
10.	<p>Каким будет результат выполнения следующего кода: using System; public class Program{ public static void Main() { Child obj; obj.Show(2); }} public class Parent{ public void Show(int iToShow) { Console.WriteLine(iToShow); } } public class Child : Parent{ public void Show() { Console.WriteLine(1); }}</p>	<p>a. 1 b. 2 c. ошибка компиляции d. ошибка времени выполнения</p>
11.	<p>Чем можно заменить строку "// 1" чтобы программа скомпилировалась. Выберите все правильные варианты ответа: using System; public class Program{ public static void Main() { B b = new B(); b.SomeMethod(); }} public class A{ public void SomeMethod() { /* ... */ }} public class B : A{ public void SomeMethod(int someArgument) { /* ... */ // 1 }</p>	<p>a. код скомпилируется и так - изменений не требуется b. public void SomeMethod() { base.SomeMethod(); } c. public virtual void SomeMethod(); d. public void A.SomeMethod(); e. public void B.SomeMethod();</p>
12.	<p>Допустимо ли определение абстрактного метода следующим образом: public class Abstract{ public abstract void Method() { /* функционал */ }}</p>	<p>a. Да, метод доступен для вызова у объектов по имени Method() b. Да, метод доступен для вызова у класса по имени Abstract.Method() c. Да, но метод не может быть вызван d. Нет, недопустимо</p>
13.	<p>Что будет выведено после компиляции и выполнении следующего кода: using System; public class Program{ public static void Main() { B b = new B(); }} public class A{ public A(){ Console.WriteLine("A "); } ~A(){ Console.WriteLine("~A" ); }} public class B : A{ public B(){ Console.WriteLine("B "); } ~B(){ Console.WriteLine("~B "); }}</p>	<p>a. A() B() ~A() ~B() b. A() B() ~B() ~A() c. B() A() ~A() ~B() d. B() A() ~B() ~A() e. Ошибка компиляции f. Ошибка времени выполнения</p>
14.	<p>Что будет выведено на экран в результате выполнения данного кода? using System; public class Program{ public static void Main() { B b = new B(); Method(b); } public static void Method(A a) { a.Method(); }} public class A{ public virtual void Method() { Console.WriteLine("A "); }} public class B : A{ public override void Method() { Console.WriteLine("B "); }}</p>	<p>a. A b. B c. ошибка компиляции d. ошибка времени выполнения</p>

№	Вопрос	Варианты ответа
15.	У класса определены два перегруженных конструктора: <code>MyClass::MyClass();</code> <code>MyClass::MyClass(int a, int b);</code> Выберите все строки кода, при выполнении которых происходит вызов конструктора класса	a. <code>MyClass obj;</code> b. <code>MyClass obj(50,100);</code> c. <code>MyClass obj[10];</code> d. <code>MyClass* obj = new MyClass;</code> e. <code>MyClass* obj = new MyClass(50);</code> f. <code>obj-&gt;MyClass();</code> g. <code>obj-&gt;MyClass(50,100);</code>
16.	Выберите все варианты с корректно компилируемой реализацией метода <code>Count</code> класса <code>Counterclass</code> <code>Counter{int num;public: void Count();};</code> //выберите варианты реализации метода вне объявления класса	a. <code>void Counter::Count();</code> b. <code>void Counter::Count() {}</code> c. <code>void Counter::Count() {num++;}</code> d. <code>void Count() {count++;}</code> e. <code>void Counter::Count() = 0;</code> f. <code>void Counter::Count(int num) {num++;}</code> g. <code>Counter::Count() {num++;}</code>
17.	Выберите модификаторы доступа, которые используются в языке C#	a. <code>private</code> b. <code>protected</code> c. <code>public</code> d. <code>abstract</code> e. <code>virtual</code> f. <code>static</code> g. <code>internal</code>
18.	Какое из определений описывает понятие "интерфейс" ?	a. Абстракция поведения класса b. Механизм, реализующий поведение класса c. Инкапсуляция поведения класса d. Иерархия поведения класса e. Методы и поля класса
19.	Что можно назвать классом?	a. Медведь b. Кошка по кличке "Пушок" c. Стол с царапиной на столешнице d. Стол e. Средство передвижения
20.	Что можно назвать экземпляром класса?	a. Медведь b. Кошка по кличке "Пушок" c. Стол с царапиной на столешнице d. Стол e. Средство передвижения