КАЗАНСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Р.С. ЯРУЛЛИН, И.Я. ЗАБОТИН

СБОРНИК ЗАДАЧ ПО КУРСУ «ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ АНАЛИЗ И ПРОГРАММИРОВАНИЕ»

Задачник

КАЗАНЬ

2024

УДК 519.682
ББК 22.18

Рекомендовано учебно-методической комиссией Института вычислительной математики и информационных технологий Казанского (Приволжского) федерального университета

Рецензенты:
д.ф.-м.н., профессор К(П)ФУ В.М. Конюхов,
к.т.н, доцент К(П)ФУ Б.Г. Мубараков

Яруллин Р.С.
Сборник задач по курсу «Объектно-ориентированный анализ и программирование»: задачник / Р.С. Яруллин, И.Я. Заботин. – Казань: 2024. – 43 с.

В задачнике представлены типовые задачи по курсу «Объектно-ориентированный анализ и программирование» и проверочные тесты. Разработанное учебное издание предназначено для студентов, обучающихся по специальностям и направлениям подготовки «Прикладная информатика», «Бизнес-информатика», и может быть использовано при изучении курсов «Объектно-ориентированный анализ и программирование», «Программирование на языке C#».

УДК 519.682
ББК 22.18

Оглавление

Предисловие

В данном задачнике представлены типовые задачи и тесты по курсу «Объектно-ориентированный анализ и программирование». Представленные задачи и тесты позволяют освоить базовые принципы объектно-ориентированного программирования, основные синтаксические правила языка программирования C#, а также выработать навыки разработки программного кода в стиле объектно-ориентированного программирования на языке C#.

Предлагаемый задачник предназначен для самостоятельной работы обучающихся на основе лекционных учебных материалов, а также для использования на практических занятиях по таким курсам, как «Программирование на языке C#», «Объектно-ориентированное программирование» в институте вычислительной математики и информационных технологий К(П)ФУ.

1 Основные понятия объектно-ориентированного программирования. Инкапсуляция

1. *Ответьте на вопросы и выберите все правильные варианты ответов.*

1.1. Инкапсуляция – это:

a. принцип ООП, при котором определяется способность языка программирования скрывать излишние детали реализации от пользователей объекта.

b. способность языка программирования строить новые определения классов на основе определений существующих классов.

c. способность языка программирования к изменению (переопределению) функционала, унаследованного от базового класса, в дочерних классах.

d. способность языка программирования объявлять пользовательские типы данных.

1.2. Что такое класс?

a. Это пользовательский тип данных, который состоит из данных и методов для выполнения

с этими данными различных действий.

b. Это фрагмент кода, в рамках которого доступна переменная.

c. Это действия, которые позволяют изменять состояние объекта.

d. Это представитель моделируемой предметной области.

1.3. Каким символом определяется начало контекста переменной?

a. «<».

b. «{».

c. «[».

d. «(».

1.4. Каким символом определяется конец контекста переменной?

a. «>».

b. «]».

c. «)».

d. «}».

1.5. Что такое контекст переменной?

a. Это экземпляр (представитель) класса.

b. Это атрибут, определяющий состояние объекта.

c. Это фрагмент кода, в рамках которого доступна переменная.

d. Это пользовательский тип данных.

1.6. С помощью какой команды выделяется память для экземпляра класса?

a. class.

b. =.

c. new.

d. public.

1.7. Каким оператором происходит доступ к открытым членам экземпляра класса?

a. new.

b. class.

c. «{».

d. «.».

1.8. Какие функции выполняет инкапсуляция?

a. Защита данных.

b. Сокрытие внутренней реализации.

c. Переопределение существующих функционалов.

d. Создание нового функционала на базе существующего.

e. Сохранение целостности данных.

1.9. Что такое модификатор доступа?

a. Это ключевое слово, позволяющий определять уровень доступа к членам класса.

b. Это атрибуты объекта класса, определяющие его состояние.

c. Это область видимости переменной в теле класса.

d. Это метод, изменяющий состояние объекта.

1.10. Что такое private?

a. Это модификатор доступа, определяющий доступ к члену класса на уровне контекста класса.

b. Это модификатор доступа, определяющий доступ к члену класса на уровне контекста класса и дочерних классов.

c. Это модификатор доступа, определяющий доступ к члену класса на уровне контекста сборки.

d. Это модификатор доступа, определяющий общедоступный член класса.

1.11. Что такое protected?

a. Это модификатор доступа, определяющий доступ к члену класса на уровне контекста класса.

b. Это модификатор доступа, определяющий доступ к члену класса на уровне контекста класса и дочерних классов.

c. Это модификатор доступа, определяющий доступ к члену класса на уровне контекста сборки.

d. Это модификатор доступа, определяющий общедоступный член класса.

1.12. Что такое internal?

a. Это модификатор доступа, определяющий доступ к члену класса на уровне контекста класса.

b. Это модификатор доступа, определяющий доступ к члену класса на уровне контекста класса и дочерних классов.

c. Это модификатор доступа, определяющий доступ к члену класса на уровне контекста сборки.

d. Это модификатор доступа, определяющий общедоступный член класса.

1.13. Что такое public?

a. Это модификатор доступа, определяющий доступ к члену класса на уровне контекста класса.

b. Это модификатор доступа, определяющий доступ к члену класса на уровне контекста класса и дочерних классов.

c. Это модификатор доступа, определяющий доступ к члену класса на уровне контекста сборки.

d. Это модификатор доступа, определяющий общедоступный член класса.

1.14. С помощью какой команды объявляется класс?

a. new.

b. private.

c. class.

d. constructor.

e. sample.

f. object.

1.15. Выберите правильные формулировки.

a. Данные класса определяют состояние экземпляра класса. Методы класса позволяют изменять состояние объекта класса.

b. Данные класса позволяют изменять состояние экземпляра класса. Методы класса задают состояние объекта класса.

c. К данным класса относятся свойства и атрибуты.

d. В классах нельзя объявлять методы.

2. *Используя [7], напишите программный код следующих заданий.*

2.1. Создайте класс Student. Определите в этом классе строковое поле name (имя).

2.2. Создайте класс Car. Определите в этом классе целочисленное поле year (год выпуска).

2.3. Реализуйте класс Point. Определите в этом классе целочисленное поле x.

2.4. Создайте класс Person с целочисленным полем age (возраст). Реализуйте в этом классе метод Print для печати поля age в консольном окне.

2.5. Разработайте класс Table с целочисленными полями rows (строки), cols (столбцы). Реализуйте в этом классе метод Display для отображения полей rows, cols в консольном окне.

2.6. Объявите класс Manager с полями age (целочисленный тип, возраст) и name (строковый тип, имя). Создайте метод GetAge для получения значения поля age и метод GetName для получения значения поля name.

2.7. Создайте класс Point3D с целочисленными полями x, y, z. Реализуйте метод Show для отображения полей x, y, z в консольном окне.

2.8. Объявите класс Shop с строковым полем name (имя). В этом классе реализуйте метод GetName для получения значения поля name и метод SetName для установления нового значения полю name.

2 Конструкторы класса

3. *Ответьте на вопросы и выберите все правильные варианты ответов.*

3.1. Конструктор класса – это:

a. это специальный метод, который вызывается при создании нового объекта класса.

b. это специальный метод, который освобождает память, занимаемую объектом.

c. это специальный метод, который копирует состояние существующего объекта, не создавая новый экземпляр класса.

d. это специальный метод, который изменяет состояние объекта.

3.2. Можно ли в теле конструктора использовать оператор return?

a. Да.

b. Нет.

c. На усмотрение пользователя.

d. Если для конструктора указать тип возвращаемого значения, то следует указать оператор return.

3.3. Какое название можно использовать при объявлении конструктора класса?

a. Любое, состоящее из одного слова.

b. Название должно начинаться с заглавной буквой.

c. Название конструктора должно совпадать с названием класса.

d. Для названия конструктора нужно использовать слово constructor.

3.4. Какой тип возвращаемого элемента следует использовать в сигнатуре конструктора?

a. Название типа возвращаемого элемента должно совпадает с названием класса.

b. Любой.

c. В сигнатуре конструктора по построению не указывается тип возвращаемого элемента.

d. void.

3.5. Можно ли использовать модификатор доступа private при определении конструктора?

a. Да.

b. Нет.

3.6. Выберите правильные типы конструкторов.

a. Стандартный конструктор.

b. Конструктор создания.

c. Конструктор копирования.

d. Конструктор удаления.

e. Конструктор изменения.

f. Конструктор с параметрами.

g. Статический конструктор

3.7. Сколько параметров имеет стандартный конструктор?

a. 0.

b. 2.

c. 1.

d. По желанию пользователя.

3.8. Когда необходимо вызвать конструктор класса?

a. В момент освобождения памяти, выделенной под объект.

b. При создании нового экземпляра класса.

c. В момент запуска программы.

d. При компиляции приложения.

3.9. Какой модификатор доступа следует определять для статического конструктора?

a. private.

b. protected.

c. internal.

d. public.

3.10. Можно ли в статическом конструкторе получить доступ к данным, задаваемым на уровне определения экземпляра класса?

a. Да.

b. Нет.

3.11. В какой момент вызывается статический конструктор?

a. При запуске приложения.

b. При обращении к статическим членам класса.

c. При создании экземпляра класса.

d. При завершении работы программы.

3.12. При создании объекта какой конструктор вызывается раньше статический или стандартный?

a. Стандартный.

b. Статический.

c. Стандартный и статический конструкторы запускаются одновременно.

d. Порядок запуска конструкторов определяет программист.

3.13. Можно ли создавать несколько конструкторов с параметрами?

a. Да.

b. Нет.

3.14. Какой оператор следует использовать совместно с конструктором копирования при создании объекта класса?

a. +.

b. –.

c. new.

d. create.

e. add.

3.15. Сколько параметров следует определять в статическом конструкторе?

a. 1.

b. 0.

c. Нет ограничений.

d. 2.

4. *Используя [2; 6], напишите программный код следующих заданий.*

4.1. Создайте класс Student. Определите в этом классе стандартный конструктор.

4.2. Создайте класс Child с открытым стандартным конструктором. Создайте два экземпляра класса Child.

4.3. Создайте класс Car. Определите в этом классе целочисленное поле year (год выпуска) и общедоступный конструктор с параметрами для инициализации поля year.

4.4. Создайте класс Car со строковыми полями name (имя) и color (цвет). Определите публичный конструктор с параметрами для инициализации полей name, color. При создании объекта lada класса Car поле name должно быть проинициализировано значением «LADA VESTA», а поле color – значением «black». Затем создайте другой экземпляр bmw класса Car с присвоением полям name, color соответствующих значений «BMW X5», «white».

4.5. Создайте класс Product с защищенным строковым полем name. Реализуйте в этом классе открытый конструктор копирования для инициализации поля name создаваемого объекта.

4.6. Создайте класс Person с закрытым целочисленным полем age (возраст). Реализуйте в этом классе открытый метод Print для печати поля age в консольном окне и открытый стандартный конструктор. В стандартном конструкторе присвойте полю year значение 18. Создайте экземпляр класса Person и для этого экземпляра вызовите метод Print.

4.7. Объявите класс Manager с закрытыми полями age (целочисленный тип, возраст) и name (строковый тип, имя) и реализуйте открытый конструктор копирования с инициализацией полей age, name. Создайте два объекта класса Manager. Для первого объекта присвойте полю age значение 20, а полю name – значение «Дамир». Для второго объекта проинициализируйте поле age значением 18, а поле name – значением «Сара».

3 Свойства классов

5. *Ответьте на вопросы и выберите все правильные варианты ответов.*

5.1. Свойство – это:

a. метод создания объекта класса.

b. специальный метод копирования.

c. метод, предоставляющий доступ к чтению и изменению состояния объекта класса.

d. метод с упрощенной сигнатурой, предоставляющий доступ к чтению и изменению состояния объекта класса.

5.2. Какие блоки доступа определяются в свойствах?

a. get.

b. private.

c. public.

d. set.

e. protected.

f. add.

5.3. В каком блоке доступа свойств используется лексема value?

a. get.

b. set.

c. get и set.

d. Лексема value в свойствах не используется.

5.4. Блок доступа get – это:

a. область кода, предоставляющая доступ на чтение инкапсулированных данных.

b. область кода, предоставляющая доступ на запись инкапсулированных данных.

5.5. Блок доступа set – это:

a. область кода, предоставляющая доступ на чтение инкапсулированных данных.

b. область кода, представляющая доступ на изменение инкапсулированных данных.

5.6. Выберите верные типы свойств.

a. Свойство копирования.

b. Стандартное свойство.

c. Свойство с параметрами.

d. Автосвойство.

e. Свойство с ограниченным доступом.

f. Статическое свойство.

5.7. В каких свойствах можно использовать фильтр значений?

a. Автосвойсто.

b. Свойство с ограниченным доступом.

c. Свойство копирования.

d. Свойство с фильтром значений.

e. Стандартное свойство.

5.8. В каких свойствах отсутствует явное обращение к хранилищу данных?

a. Автосвойсто.

b. Свойство с ограниченным доступом.

c. Свойство копирования.

d. Свойство с фильтром значений.

e. Стандартное свойство.

5.9. Можно ли не объявлять блок доступа get в стандартных свойствах?

a. Да.

b. Нет.

5.10. Можно ли не объявлять блок доступа set в автосвойствах?

a. Да.

b. Нет.

5.11. В каком блоке доступа можно получить доступ к лексеме value?

a. get.

b. set.

c. get и set.

d. Лексема value в свойствах не используется.

5.12. Сколько параметров можно указывать в свойствах?

a. 1.

b. Ограничений нет.

c. В сигнатуре свойств параметры не предусмотрены.

d. На усмотрение пользователя.

5.13. Можно ли не объявлять блок доступа set в свойствах с ограниченным доступом?

a. Да.

b. Нет.

5.14. Укажите предпосылки возникновения синтаксиса объектной инициализации значений свойств.

a. Уменьшение количества конструкторов с параметрами.

b. Упрощение сигнатуры инициализации исходных значений свойств.

c. Уменьшение количества свойств.

d. Для задания стандартного значения свойств.

5.15. Можно ли объявлять свойства класса с одинаковыми именами?

a. Да.

b. Нет.

6. *Используя [5], напишите программный код следующих заданий.*

6.1. Создайте класс Student. Определите в этом классе строковое автосвойство Name (имя).

6.2. Создайте класс Child с открытыми стандартным конструктором и целочисленным автосвойством Age (возраст). Создайте два экземпляра класса Child с инициализацией автосвойства Age.

6.3. Создайте класс Car. Определите в этом классе открытое целочисленное стандартное свойство Year, закрытое целочисленное поле year (год выпуска) и общедоступный стандартный конструктор. В свойстве Year доступ к полю year осуществляется без ограничений, а изменение значения поля year допустимо на положительные числа.

6.4. Определите класс Car с открытыми строковыми автосвойствами Name (имя) и Color (цвет) и с открытым стандартным конструктором. Используйте объектную инициализацию при создании экземпляра класса Car. В создаваемом объекте свойству Name присвойте значение "KIA SOUL", а свойству Color – значение «green».

6.5. Создайте класс Product с защищенным строковым полем name. В этом классе реализуйте открытое свойство Name с ограниченным доступом. В свойстве Name невозможно изменять значение поля name, но можно получить доступ на чтение к полю name. Кроме того, создайте открытый стандартный конструктор с присвоением полю name значения «Рамиль».

4 Наследование

7. *Ответьте на вопросы и выберите все правильные варианты ответов.*

7.1. Наследование – это:

a. принцип ООП, при котором определяется способность языка программирования скрывать излишние детали реализации от пользователей объекта.

b. способность языка программирования строить новые определения классов на основе определений существующих классов.

c. способность языка программирования к изменению (переопределению) функционала, унаследованного от базового класса, в дочерних классах.

d. способность языка программирования создавать экземпляра класса с помощью конструкторов.

7.2. Разрешено ли в языке C# множественное наследование классов?

a. Да.

b. Нет.

c. В языке C# нет таких ограничений.

d. В языке C# каждый класс может иметь только два родительских классов.

7.3. С помощью какой команды можно запечатать наследование класса?

a. class.

b. partial class.

c. sealed.

d. override.

7.4. С помощью какого символа определяется признак наследования между двумя классами?

a. «.».

b. «=».

c. «:».

d. «+».

e. «<».

7.5. В каком порядке вызываются конструкторы базового и дочернего классов при наследовании?

a. В случайном порядке.

b. Порядок определяет программист.

c. Сначала вызывается конструктор дочернего класса, а потом базового класса.

d. Сначала вызывается конструктор базового класса, а потом дочернего класса.

7.6. С помощью какой команды вызывается конструктор базового класса в производном классе?

a. main.

b. parent.

c. base.

d. child.

e. class.

7.7. Какой модификатор доступа следует использовать для инкапсуляции члена базового класса в рамках контекстов базового и дочернего классов?

a. public.

b. internal.

c. private.

d. protected.

7.8. Можно ли создавать наследников, если при объявлении базового класса использовалось ключевое слово sealed?

a. Нет.

b. Да.

7.9. Пусть пять дочерних классов определены для некоторого базового класса. Можно ли для этого базового класса создать еще один производный класс?

a. Да.

b. Нет.

c. Предположение ошибочно, поскольку множественное наследование запрещено в языке C#.

7.10. Пусть класс B является производным от класса A. Можно ли для класса B создать дочерний класс C?

a. Да.

b. Нет.

c. Предположение ошибочно, поскольку множественное наследование запрещено в языке C#.

7.11. Пусть класс A является родительским для класса B. Можно ли для класса B определить еще один родительский класс?

a. Да.

b. Нет.

7.12. Пусть класс A является наследником класса B, а класс C является производным класса A. Выберите порядок вызова конструкторов при создании экземпляра класса C.

a. A → C → B.

b. B → A → C.

c. C → A → B.

d. A → B → C.

7.13. Пусть некоторый член класса A имеет модификатор доступа private, а класс B является наследником класса A. Можно ли получить доступ к этому члену в контексте класса B?

a. Да.

b. Нет.

7.14. Пусть некоторый член класса A имеет модификатор доступа protected, а класс B является наследником класса A. Можно ли получить доступ к этому члену в классе B?

a. Да.

b. Нет.

7.15. Пусть некоторый член класса A имеет модификатор доступа private, а класс B является наследником класса A. Какой модификатор доступа следует использовать для инкапсуляции этого члена в контекстах классов A и B?

a. public.

b. open.

c. protected.

d. internal.

e. sealed.

8. *Используя [1; 3], напишите программный код следующих заданий.*

8.1. Создайте два класса Person и Student. Класс Student является дочерним, а класс Person – базовым. Установите корректное отношение наследования между классами Person и Student.

8.2. Создайте три класса Animal, Cat, Dog. Класс Animal является родительским для классов Cat, Dog. Установите правильные отношения наследования между этими тремя классами.

8.3. Создайте два класса Entity, Product. Entity является базовым классом, а Product – производным. Установите корректное отношение наследования между двумя классами Entity, Product.

8.4. Пусть класс Dishes является родительским, а Cup – дочерним классом. Установить корректное отношение наследования между двумя классами Dishes, Cup.

8.5. Создайте три класса Entity, Staff, Manager. Класс Manager наследуется от класса Staff, а класс Staff от класса Entity. Установите правильные отношения наследования между этими классами.

8.6. Реализуйте два класса Animal, Predator, где Animal – родительский класс, а Predator – дочерний. Создайте целочисленное поле age. Доступ к полю age возможен в контекстах классов Animal, Predator.

8.7. Пусть класс Transport является базовым, а SpaceShuttle – дочерний класс. Инкапсулируйте строковое поле nate в рамках классов Transport, SpaceShuttle.

5 Полиморфизм

9. *Ответьте на вопросы и выберите все правильные варианты ответов.*

9.1. Полиморфизм – это:

a. принцип ООП, при котором определяется способность языка программирования скрывать излишние детали реализации от пользователей объекта.

b. способность языка программирования строить новые определения классов на основе определений существующих классов.

c. способность языка программирования к изменению (переопределению) функционала, унаследованного от базового класса, в дочерних классах.

d. способность языка программирования моделировать предметную область.

9.2. Полиморфный интерфейс – это:

a. именованный набор абстрактных членов.

b. набор членов базового класса, который доступен всем наследникам.

c. набор членов дочернего класса, который доступен родительскому классу.

d. совокупность членов класса.

9.3. Виртуальный член – это:

a. член базового класса, определяющий стандартную реализацию, который может быть изменен в производном классе.

b. член производного класса, определяющий стандартную реализацию, который может быть изменен в родительском классе.

c. это член базового класса, который не предусматривает стандартную реализацию, а предлагает только сигнатуру.

d. это любой член класса, который доступен в дочернем классе.

9.4. Абстрактный член – это:

a. член базового класса, определяющий стандартную реализацию, который может быть изменен в производном классе.

b. член производного класса, определяющий стандартную реализацию, который может быть изменен в родительском классе.

c. это член базового класса, который не предусматривает стандартную реализацию, а предлагает только сигнатуру.

d. это любой член класса, который доступен в дочернем классе.

9.5. Какое ключевое слово следует использовать в производном классе при переопределении виртуальноro члена базового класса?

a. virtual.

b. abstract.

c. override.

d. sealed.

e. class.

f. set.

g. get.

9.6. Какое ключевое слово следует использовать в производном классе при переопределении абстрактного члена базового класса?

a. virtual.

b. abstract.

c. override.

d. sealed.

e. class.

f. set.

g. get.

9.7. Для запечатывания возможности переопределения виртуального члена базового класса в производных классах какое ключевое слово следует добавить в сигнатуру этого виртуального члена?

a. virtual.

b. abstract.

c. override.

d. sealed.

e. protected.

f. partial.

g. private.

9.8. Можно ли создавать экземпляры абстрактного класса?

a. Да.

b. Нет.

9.9. Какое ключевое слово следует использовать для обозначения класса абстрактным?

a. virtual.

b. abstract.

c. override.

d. sealed.

e. partial.

f. class.

9.10. Выберите правильные утверждения.

a. Абстрактный член может иметь стандартную реализацию.

b. Абстрактный член не может иметь стандартную реализацию.

c. Виртуальный член имеет стандартную реализацию.

d. Виртуальный член не может иметь стандартную реализацию.

9.11. Выберите правильные утверждения.

a. В производных классах в обязательном порядке следует переопределять все виртуальные члены базового класса.

b. В производных классах в обязательном порядке следует переопределять все абстрактные члены базового класса.

c. Абстрактные члены нельзя объявлять с модификатором доступа private.

d. Виртуальные члены нельзя объявлять с модификатором доступа private.

9.12. Интерфейс – это:

a. член базового класса, определяющий стандартную реализацию, который может быть изменен в производном классе.

b. это член базового класса, который не предусматривает стандартную реализацию, а предлагает только сигнатуру.

c. это именованный набор абстрактных членов.

d. это совокупность членов класса.

e. это абстрактный класс.

f. это объект абстрактного класса.

9.13. Какое ключевое слово следует использовать для обозначения метода виртуальным?

a. virtual.

b. abstract.

c. override.

d. sealed.

9.14. Какое ключевое слово следует использовать для обозначения метода абстрактным?

a. virtual.

b. abstract.

c. override.

d. sealed.

9.15. Пусть определены интерфейсы I1, I2 и класс C. Можно ли в классе C реализовать одновременно все члены интерфейсов I1, I2?

a. Нет, множественное наследование интерфейсов запрещено.

b. Нет, множественное наследование классов запрещено.

c. Да.

d. Нет, классы не могут переопределять члены интерфейсов.

10. *Используя [4], напишите программный код следующих заданий.*

10.1. Создайте класс Strategy с виртуальным методом Display. Метод Display распечатывает на консольном окне сообщение «Strategy».

10.2. Создайте класс Weather с виртуальным методом Show. Метод Show отображает на консольном окне сообщение «My Weather». Создайте экземпляр класса Weather и для созданного экземпляра класса вызовите метод Show.

10.3. Создайте два класса Strategy и ConservativeStrategy. Класс Strategy является базовым, а ConservativeStrategy – дочерним. Реализуйте виртуальный метод Display в классе Strategy и переопределите его в дочернем классе ConservativeStrategy.

10.4. Пусть Animal является базовым классом, а Cat – дочерним. В классе Animal создайте закрытое строковое поле type, а в классе Cat закрытое целочисленное поле age. Для классов Animal, Cat реализуйте стандартные конструкторы. В стандартном конструкторе класса Animal полю type присваивается значение "My Type", а в стандартном конструкторе класса Cat поле age инициализируется значением 5. Создайте виртуальным метод Print в классе Animal. Методом Print в классе Animal распечатывается значение поля

type, а в классе Cat распечатывается значение поля age. Создайте по одному объекту классов Animal, Cat. Для каждого объекта вызовите метод Print.

10.5. *Пусть Entity является абстрактным классом. В классе Entity реализуйте абстрактный метод Display.*

10.6. *Пусть Entity является абстрактным базовым классом, а Product – производным классом. Абстрактный метод Display объявлен в классе Entity, а в классе Product этот метод переопределен для отображения в консольном окне сообщения «My Product». Создайте объект класса Product и для этого объекта вызовите метод Display.*

10.7. *Создайте интерфейс IPrintable. В этом интерфейсе определите метод Display без параметров. Затем определите класс ConsolePrinting с реализацией методов интерфейса IPrintable для печати сообщения «My Console» на консольном окне. Создайте объект класса ConsolePrinting и вызовите для этого объекта метод Display.*

Литература

1. *Microsoft, Learn.* Документация по C#. Вложенные типы. — 2024. — URL: https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/classes-and-structs/nested-types (дата обр. 27.02.2024).
2. *Microsoft, Learn.* Документация по C#. Конструкторы. — 2024. — URL: https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/classes-and-structs/constructors (дата обр. 27.02.2024).
3. *Microsoft, Learn.* Документация по C#. Наследование. — 2024. — URL: https://docs.microsoft.com/ru-ru/dotnet/csharp/fundamentals/object-oriented/inheritance (дата обр. 27.02.2024).
4. *Microsoft, Learn.* Документация по C#. Полиморфизм. — 2024. — URL: https://docs.microsoft.com/ru-ru/dotnet/csharp/fundamentals/object-oriented/polymorphism (дата обр. 27.02.2024).
5. *Microsoft, Learn.* Документация по C#. Свойства. — 2024. — URL: https://docs.microsoft.com/ru-ru/dotnet/csharp/properties (дата обр. 27.02.2024).
6. *Microsoft, Learn.* Документация по C#. Стандартные блоки программы. — 2024. — URL: https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/program-building-blocks (дата обр. 27.02.2024).
7. *Microsoft, Learn.* Документация по C#. Типы и члены. — 2024. — URL: https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/types (дата обр. 27.02.2024).

Учебное издание

Яруллин Рашид Саматович
Заботин Игорь Ярославич

СБОРНИК ЗАДАЧ ПО КУРСУ «ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ АНАЛИЗ И ПРОГРАММИРОВАНИЕ»

задачник