Министерство образования и науки Российской Федерации

ФГБОУ ВО «Бурятский государственный университет имени Доржи Банзарова»

Институт математики и информатики

Кафедра информационных технологий

|  |  |
| --- | --- |
| рассмотрено и одобрено:  Председатель УМК ИМИ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Телешева Л.А.  заседание от «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 202\_\_ г.  протокол № \_\_\_ | УТВЕРЖДАЮ:  Зав. кафедрой ИТ  \_\_\_\_\_\_\_\_\_\_Цыбиков А.С.  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 202\_\_ г. |

**Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине**

**Объектно-ориентированное программирование**

наименование дисциплины

02.03.03 Математическое обеспечения и администрирование информационных систем

шифр и наименование направления

квалификация: бакалавр

Профиль подготовки

Информационные системы и базы данных

Составитель: Хабитуев Б.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 202\_\_ г.

Улан-Удэ

202\_\_

Оглавление

[I.Паспорт фонда оценочных средств 7](#_Toc100303152)

[II. Паспорт компетенций 7](#_Toc100303153)

[1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы 8](#_Toc100303154)

[2. Этапы формирования компетенций 9](#_Toc100303155)

[3. Показатели оценивания компетенций 10](#_Toc100303156)

[4. Шкала оценивания 11](#_Toc100303157)

[5. Компетентностно-ориентированные оценочные средства 13](#_Toc100303158)

[III. Оценочное средство Лабораторные работы 13](#_Toc100303159)

[Лабораторная работа №1. Основы C# 13](#_Toc100303160)

[Лабораторная работа №2. Классы в языке С# 13](#_Toc100303161)

[Лабораторная работа №3. Наследование 14](#_Toc100303162)

[Лабораторная работа №4. Композиция 14](#_Toc100303163)

[Лабораторная работа №5. Наследование и восходящее преобразование 14](#_Toc100303164)

[Лабораторная работа №6. Интерфейсы 14](#_Toc100303165)

[Лабораторная работа №7. Лямбда выражения 14](#_Toc100303166)

[Лабораторная работа №8. Коллекции и абстрактные классы 15](#_Toc100303167)

[Лабораторная работа №9. Параметризация 15](#_Toc100303168)

[Лабораторная работа №10. Исключительные ситуации 15](#_Toc100303169)

[Лабораторная работа №11. Паттерн проектирования Стратегия 15](#_Toc100303170)

[Лабораторная работа №12. Паттерн проектирования Наблюдатель 16](#_Toc100303171)

[Лабораторная работа №13. Паттерн проектирования Декоратор 16](#_Toc100303172)

[Лабораторная работа №14. Паттерн проектирования Команда 16](#_Toc100303173)

[Лабораторная работа №15. Паттерн проектирования Адаптер 16](#_Toc100303174)

[Лабораторная работа №16. Паттерн проектирования Фасад 16](#_Toc100303175)

[IV. Оценочное средство Посещение 17](#_Toc100303176)

[V. Оценочное средство Задание на СРС 17](#_Toc100303177)

[СРС №1. Ввод – вывод в C#. Работа с файлами 17](#_Toc100303178)

[СРС № 2. Сериализация объектов 18](#_Toc100303179)

[СРС №3. Reflection API. Динамическое определение в C# 20](#_Toc100303180)

[СРС №4. Работа с СУБД MS SQL 20](#_Toc100303181)

[СРС №5. Паттерн Domain model 20](#_Toc100303182)

[СРС №6. Паттерн Identity map 21](#_Toc100303183)

[СРС №7. Паттерн Lazy Load 21](#_Toc100303184)

[VII. Оценочное средство зачёт/экзамен 21](#_Toc100303185)

[Вопросы для подготовки к зачёту 21](#_Toc100303186)

[Вопросы для подготовки к экзамену 23](#_Toc100303187)

[Приложение №1. Лабораторные работы 25](#_Toc100303188)

[Формирование вариантов в лабораторных работах 1-10. 25](#_Toc100303189)

[Лабораторная работа №1 26](#_Toc100303190)

[Задача №1 26](#_Toc100303191)

[Задача №2 26](#_Toc100303192)

[Задача №3 26](#_Toc100303193)

[Задача №4 26](#_Toc100303194)

[Задача №5 26](#_Toc100303195)

[Задача №6 27](#_Toc100303196)

[Задача №7 27](#_Toc100303197)

[Задача №8 27](#_Toc100303198)

[Задача №9 27](#_Toc100303199)

[Задача №10 27](#_Toc100303200)

[Задача №11 28](#_Toc100303201)

[Задача №12 28](#_Toc100303202)

[Задача №13 28](#_Toc100303203)

[Задача №14 28](#_Toc100303204)

[Задача №15 28](#_Toc100303205)

[Задача №16 28](#_Toc100303206)

[Задача №17 28](#_Toc100303207)

[Задача №18 29](#_Toc100303208)

[Лабораторная работа №2 30](#_Toc100303209)

[Задача №1 30](#_Toc100303210)

[Задача №2 30](#_Toc100303211)

[Задача №3 30](#_Toc100303212)

[Задача №4 30](#_Toc100303213)

[Задача №5 30](#_Toc100303214)

[Задача №6 30](#_Toc100303215)

[Задача №7 30](#_Toc100303216)

[Задача №8 30](#_Toc100303217)

[Задача №9 31](#_Toc100303218)

[Задача №10 31](#_Toc100303219)

[Задача №11 31](#_Toc100303220)

[Задача №12 31](#_Toc100303221)

[Задача №13 31](#_Toc100303222)

[Задача №14 31](#_Toc100303223)

[Задача №15 31](#_Toc100303224)

[Задача №16 31](#_Toc100303225)

[Задача №17 32](#_Toc100303226)

[Задача №18 32](#_Toc100303227)

[Лабораторная работа №3. Наследование 33](#_Toc100303228)

[Задача №1 33](#_Toc100303229)

[Задача №2 33](#_Toc100303230)

[Задача №3 33](#_Toc100303231)

[Задача №4 33](#_Toc100303232)

[Задача №5 34](#_Toc100303233)

[Задача №6 34](#_Toc100303234)

[Задача №7 34](#_Toc100303235)

[Задача №8 34](#_Toc100303236)

[Задача №9 35](#_Toc100303237)

[Задача №10 35](#_Toc100303238)

[Лабораторная работа №4. Композиция 36](#_Toc100303239)

[Задача №1 36](#_Toc100303240)

[Задача №2 36](#_Toc100303241)

[Задача №3 36](#_Toc100303242)

[Задача №4 36](#_Toc100303243)

[Задача №5 36](#_Toc100303244)

[Задача №6 36](#_Toc100303245)

[Задача №7 36](#_Toc100303246)

[Задача №8 36](#_Toc100303247)

[Задача №9 37](#_Toc100303248)

[Задача №10 37](#_Toc100303249)

[Лабораторная работа №5. Наследование 38](#_Toc100303250)

[Задача №1 38](#_Toc100303251)

[Задача №2 39](#_Toc100303252)

[Задача №3 40](#_Toc100303253)

[Задача №4 41](#_Toc100303254)

[Задача №5 42](#_Toc100303255)

[Задача №6 43](#_Toc100303256)

[Задача №7 44](#_Toc100303257)

[Задача №8 45](#_Toc100303258)

[Задача №9 46](#_Toc100303259)

[Задача №10 47](#_Toc100303260)

[Лабораторная работа №6. Интерфейсы и абстрактные классы 48](#_Toc100303261)

[Задача №1 48](#_Toc100303262)

[Задача №2 48](#_Toc100303263)

[Задача №3 48](#_Toc100303264)

[Задача №4 48](#_Toc100303265)

[Задача №5 48](#_Toc100303266)

[Задача №6 48](#_Toc100303267)

[Лабораторная работа №7. Лямбда выражения 49](#_Toc100303268)

[Задача №1 49](#_Toc100303269)

[Задача №2 49](#_Toc100303270)

[Задача №3 49](#_Toc100303271)

[Задача №4 49](#_Toc100303272)

[Задача №5 50](#_Toc100303273)

[Задача №6 50](#_Toc100303274)

[Задача №7 50](#_Toc100303275)

[Задача №8 50](#_Toc100303276)

[Задача №9 51](#_Toc100303277)

[Задача №10 51](#_Toc100303278)

[Задача №11 51](#_Toc100303279)

[Задача №12 51](#_Toc100303280)

[Задача №13 51](#_Toc100303281)

[Задача №14 52](#_Toc100303282)

[Задача №15 52](#_Toc100303283)

[Задача №16 52](#_Toc100303284)

[Задача №17 52](#_Toc100303285)

[Задача №18 53](#_Toc100303286)

[Лабораторная работа №8. Коллекции и абстрактные классы 53](#_Toc100303287)

[Задача №1 53](#_Toc100303288)

[Задача №2 54](#_Toc100303289)

[Задача №3 56](#_Toc100303290)

[Задача №4 57](#_Toc100303291)

[Задача №5 58](#_Toc100303292)

[Задача №6 59](#_Toc100303293)

[Лабораторная работа №9. Параметризация 60](#_Toc100303294)

[Лабораторная работа №10. Исключительные ситуации 61](#_Toc100303295)

[Задача №1 61](#_Toc100303296)

[Задача №2 61](#_Toc100303297)

[Задача №3 61](#_Toc100303298)

[Задача №4 61](#_Toc100303299)

[Задача №5 61](#_Toc100303300)

[Задача №6 62](#_Toc100303301)

[Задача №7 62](#_Toc100303302)

[Задача №8 62](#_Toc100303303)

[Задача №9 62](#_Toc100303304)

[Задача №10 63](#_Toc100303305)

[Задача №11 63](#_Toc100303306)

[Задача №12 63](#_Toc100303307)

[Задача №13 63](#_Toc100303308)

[Задача №14 64](#_Toc100303309)

[Задача №15 64](#_Toc100303310)

[Задача №16 64](#_Toc100303311)

[Задача №17 65](#_Toc100303312)

[Задача №18 65](#_Toc100303313)

[Формирование варианта в лабораторных работах 11-13 66](#_Toc100303314)

[Лабораторная работа №11. Паттерн проектирования Стратегия 67](#_Toc100303315)

[Задача №1. RPG игра 67](#_Toc100303316)

[Задача №2. RPG игра 2 69](#_Toc100303317)

[Лабораторная работа №12. Наблюдатель 71](#_Toc100303318)

[Задача №1 71](#_Toc100303319)

[Задача №2 74](#_Toc100303320)

[Лабораторная работа №13. Декоратор 77](#_Toc100303321)

[Задача №1 77](#_Toc100303322)

[Задача №2 78](#_Toc100303323)

[Формирование варианта в лабораторной работе 14 80](#_Toc100303324)

[Лабораторная работа №14. Паттерн проектирования Команда 80](#_Toc100303325)

[Задача №1. Робот 80](#_Toc100303326)

[Задача №2. Умная теплица 82](#_Toc100303327)

[Задача №3. MEO GEO 83](#_Toc100303328)

[Лабораторная работа №15. Адаптер 85](#_Toc100303329)

[Лабораторная работа №16. Фасад 86](#_Toc100303330)

# I.Паспорт фонда оценочных средств

по дисциплине (модулю) **«**Объектно-ориентированное программирование»

ОПК-3. Способен применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения.

ПК-3. Способен применять современные технологии проектирования программного обеспечения.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | Оценочные средства | |
| № | Контролируемые разделы, темы, модули | Формируе-мые компе-тенции | Другие оценочные средства | |
| Вид | Балл |
| 1 | Основы C# | ОПК - 3 | Лабораторная работа  Посещение |  |
| 2 | ООП в С# | ОПК - 3 | Посещение |  |
| 3 | Разработка приложений в С# | ОПК - 3 | Посещение  Отчёт по СРС |  |
|  | 4 семестр |  | Зачёт | 40 |
|  |  | Всего |  | 100 |
| 4 | Принципы объектно-ориентированного программирования | ПК-3 | Лабораторная работа  Посещение |  |
| 5 | Паттерны для работы с Базами данных | ПК-3 | Отчёт по СРС |  |
|  | 5 семестр |  | Экзамен | 40 |
|  |  | Всего |  | 100 |

# II. Паспорт компетенций

## Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

в рамках учебной дисциплины «Объектно – ориентированное программирование**»**

Конечными результатами освоения дисциплины «Объектно – ориентированное программирование**»** являются сформированные на первом уровне когнитивные дескрипторы «знать», «уметь», «владеть», расписанные по отдельным компетенциям. Формирование этих дескрипторов происходит в течение одного семестра по этапам в рамках различного вида занятий и самостоятельной работы

|  |  |  |
| --- | --- | --- |
| ОПК-3. Способен применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения. | | |
| № п/п | Контролируемые разделы (темы) дисциплины | Перечень планируемых результатов обучения по дисциплине |
| 1 | Основы C# | в результате освоения дисциплины студент должен:  *Знать*:  - основы языка C#;  - стандартные библиотеки;  *Уметь:*  - создавать консольные приложения на языке C#;  - использовать базовые приёмы ООП в программном коде на C#;  *Владеть:*  - методологией и навыками решения практических задач;  - навыками использования технических и программных средств реализации информационных процессов. |
| 2 | ООП в С# |
| 3 | Разработка приложений в С# |

|  |  |  |
| --- | --- | --- |
| ПК-3. Способен применять современные технологии проектирования программного обеспечения. | | |
| № п/п | Контролируемые разделы (темы) дисциплины | Перечень планируемых результатов обучения по дисциплине |
| 1 | Принципы объектно-ориентированного программирования | в результате освоения дисциплины студент должен:  *Знать*:  - основные принципы объектно-ориентированного подхода;  - основные шаблоны проектирования;  - основные понятия языка UML.  *Уметь:*  - определять и применять различные шаблоны проектирования.  *Владеть:*  - навыками использования технических и программных средств реализации информационных процессов;  - методологией и основными приемами объектно-ориентированного программирования для решения задач с использованием языка C#;  - методологией и основными приемами алгоритмизации решения задач с использованием принципов ООП. |
| 2 | Паттерны для работы с Базами данных |
| 3 |  |

## Этапы формирования компетенций

|  |  |  |
| --- | --- | --- |
| Коды компетенций: ОПК-3. Способен применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения. | | |
| Индексы дисциплин | Название дисциплины | Семестр |
| Б1.О.04.01 | Программирование | 1-2 |
| Б1.О.04.02 | Архитектура компьютеров и операционные системы | 2 |
| Б1.О.04.03 | Компьютерные сети и коммутационное оборудование | 3 |
| Б1.О.04.05 | Структуры и алгоритмы компьютерной обработки данных | 3-4 |
| Б1.О.04.06 | Объектно-ориентированное программирование | 4-5 |
| Б1.О.04.07 | Функциональное и параллельное программирование | 6 |
| Б2.О.01(У) | Технологическая (проектно-технологическая) практика | 5 |
| Б2.О.02(П) | Технологическая (проектно-технологическая) практика | 7-8 |
| Б2.О.03(П) | Преддипломная практика | 8 |
| Б3.01(Г) | Подготовка к сдаче и сдача государственного экзамена |  |
| Б3.02(Д) | Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедура защиты |  |
| ФТД.01 | Олимпиадные задачи | 2 |
| ФТД.02 | Курс по программированию | 5 |

|  |  |  |
| --- | --- | --- |
| ПК-3. Способен применять современные технологии проектирования программного обеспечения. | | |
| Индексы дисциплин | Название дисциплины | Семестр |
| Б1.О.04.05 | Структуры и алгоритмы компьютерной обработки данных | 3-4 |
| Б1.О.04.06 | Объектно-ориентированное программирование | 4-5 |
| Б1.О.04.08 | Технологии разработки программного обеспечения | 7 |
| Б1.В.03.01 | Введение в базы данных | 4 |
| Б1.В.03.03 | Базы данных | 5-6 |
| Б1.В.03.05 | Web-программирование | 6 |
| Б1.В.04.02 | Моделирование информационных систем | 7 |
| Б1.В.ДВ.04.01 | Разработка на Java | 7-8 |
| Б1.В.ДВ.05.01 | Web-разработка | 7-8 |
| Б2.О.02(П) | Технологическая (проектно-технологическая) практика | 7-8 |
| Б2.О.03(П) | Преддипломная практика | 8 |
| Б3.01(Г) | Подготовка к сдаче и сдача государственного экзамена |  |
| Б3.02(Д) | Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедура защиты |  |

## Показатели оценивания компетенций

|  |  |  |
| --- | --- | --- |
| Показатели оценивания компетенций | Уровень сформированности компетенций | Шкала оценивания |
| Обучаемый демонстрирует способность к полной самостоятельности (допускаются консультации с преподавателем по сопутствующим вопросам) в выборе способа решения неизвестных или нестандартных заданий в рамках учебной дисциплины с использованием навыков, знаний, умений и навыков, полученных как в ходе освоения данной учебной дисциплины, так и смежных дисциплин, следует считать компетенцию сформированной на высоком уровне. Присутствие сформированной компетенции на высоком уровне, способность к ее дальнейшему саморазвитию и высокой адаптивности практического применения к изменяющимся условиям профессиональной задачи | Высокий | 85-100 баллов оценка «отлично» |
| Способность обучающегося продемонстрировать самостоятельное применение знаний, умений и навыков при решении заданий, аналогичных тем, которые представлял преподаватель при потенциальном формировании компетенции ,и подтверждает наличие сформированной компетенции, причем на более высоком уровне. Наличие сформированной компетенции на повышенном уровне самостоятельности со стороны обучаемого при ее практической демонстрации в ходе заданий следует оценивать как положительное и устойчиво закрепленное в практическом навыке | Базовый | 70-84 баллов оценка «хорошо» |
| Если обучаемый демонстрирует самостоятельность в применении знаний, умений и навыков к решению учебных заданий в соответствии с образцом, данным преподавателем, по заданиям, решение которых было показано преподавателем, следует считать, что компетенция сформирована, но ее уровень недостаточно высок. Поскольку выявлено наличие сформированной компетенции, ее следует оценивать положительно, но на низком уровне | Пороговый | 60-69 баллов оценка «удовле-творительно» |
| Неспособность обучаемого самостоятельно продемонстрировать наличие знаний при решении заданий, которые были представлены преподавателем вместе с образцом их решения, отсутствие самостоятельности в применении умения к использованию методов освоения учебной дисциплины и неспособность самостоятельно проявить навык повторения решения поставленной задачи по стандартному образцу свидетельствуют об отсутствии сформированной компетенции. Отсутствие подтверждения наличия сформированности компетенции свидетельствует об отрицательных результатах освоения учебной дисциплины | Низкий | 0-59 баллов оценка «неудов-летворительно» |

Положительная оценка по дисциплине, может выставляться и при неполной сформированности компетенций в ходе освоения отдельной учебной дисциплины, если их формирование предполагается продолжить на более поздних этапах обучения, в ходе изучения других учебных дисциплин (в соответствии с разделом Место дисциплины в структуре ОП в Рабочей программе дисциплины).

## 4. Шкала оценивания

|  |  |  |  |
| --- | --- | --- | --- |
| **Планируемые результаты обучения** | **Показатели оценивания компетенций** | **Уровень сформированности компетенций** | **Шкала оценивания** |
| **Знать:** Теоретические основы объектно-ориентированного программирования.  **Уметь:** производить формализацию задачи, выделять и проектировать собственные программные классы.  **Владеть:** основными управляющими конструкциями языка C#, стандартными контейнерами (List, List, Dictionary) | Знает основные положения и принципы теории объектно-ориентированного программирования (инкапсуляция, наследование, полиморфизм).  Способен разрабатывать и реализовывать в виде программного кода собственные структуры данных (не связанные отношением наследования или композиции) в соответствии с поставленной задачей. | **Пороговый**  ( обязательный для всех студентов) | 60-69 баллов оценка «удовле-творительно» |
| **Знать:** Основные положения теории объектно-ориентированного программирования о расширении программного кода  **Уметь:**  Выбирать и реализовывать использование следующих механик и особенностей ОО языков:  - наследование;  - полиморфизм  - композиция;  - абстрактные классы;  - интерфейсы.  **Владеть:**  - методами формализации задач, выделения основных параметров классов, связей между классами. | Знает принцип расширения программного кода (композиция), способен сформулировать и привести примеры использования композиции, наследования, абстрактных классов и интерфейсов.  Способен реализовать на языке Java примеры использования: композиции, наследования, абстрактных классов и интерфейсов.  Способен реализовать в виде программного кода структуры классов реализующих полиморфизм и/или композицию. | **Базовый** | 70-84 баллов оценка «хорошо» |
| **Знать:** способы реализации и случаи применения основных шаблонов проектирования;  **Уметь:**  - реализовывать шаблоны проектирования на ОО языках;  **Владеть:**  - методами формализации задач, выделения связей между различными частями программного кода. | Знает структуру и случаи применения шаблонов проектирования (стратегия, наблюдатель, декоратор, фабрика, команда). Способен сформулировать условия применения паттернов представления данных в веб.  Умеет распознавать, проектировать и реализовывать шаблоны проектирования в виде программного кода при решении практических задач.  Способен комбинировать различные методы для решения задач. | **Высокий** | 85-100 баллов оценка «отлично» |

## 5. Компетентностно-ориентированные оценочные средства

**Средства оценивания:** Диагностирующий контроль освоению компетенции ОПК-3, ПК-3 по проводится в виде оценки посещения, коллоквиумов и комплектов задач по предмету «Объектно-ориентированное программирование».

В течение семестра проводятся: обучающимся выдаются задания на лабораторные и самостоятельную работу, на зачётном занятии обучающимся предлагается вопросы для зачёта/экзамена (устный коллоквиум) и задания для письменного решения (с использованием ЭВМ).

# III. Оценочное средство Лабораторные работы

по дисциплине «Объектно-ориентированное программирование»

В Лабораторных работах необходимо выполнить задание согласно варианту (см. Приложение №1). В заданиях на создание классов каждый из классов должен быть сохранён в отдельном файле. Функционал классов должен быть продемонстрирован в отдельном классе с методом – точкой доступа.

## Лабораторная работа №1. Основы C#

Задание подразумевает создание проекта в виде консольного приложения, в котором происходит решение задачи с выводом ответа (в консоль).

Входные данные должны быть получены из консоли (ввод с клавиатуры). При разработке программного кода рекомендуется обратить особое внимание на формат входных и выходных данных (требуемый формат приводится в условии задачи).

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код верно решает задачу, в консоль выводится ответ, соответствующий формату выходных данных. | 2 |
| Итого | 2 |

## Лабораторная работа №2. Классы в языке С#

Задание подразумевает создание класса. Класс должен реализовывать описанный функционал. Класс должен содержать конструктор класса и хотя бы один перегруженный метод.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Создан класс. Определены атрибуты и методы. Реализован конструктор класса. | 1 |
| Реализован класс пользователь, класс разрабатываемый по условию задачи протестирован. | 2 |
| Итого | 3 |

## Лабораторная работа №3. Наследование

Задание подразумевает разработку двух классов: родителя и наследника, класс наследник должен переопределить метод родителя. Также в родительском классе должен быть описан конструктор с непустым набором аргументов.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Созданы классы, удовлетворяющие условию задачи | 3 |
| Итого | 3 |

## Лабораторная работа №4. Композиция

Задание подразумевает создание класса и использование его в качестве атрибута для другого класса. Полученные классы нужно использовать для решения задачи.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Созданы классы, удовлетворяющие условию задачи | 3 |
| Итого | 3 |

## Лабораторная работа №5. Наследование и восходящее преобразование

Задание подразумевает разработку структуры классов, демонстрирующую наследование. Для демонстрации работы необходимо разработать класс пользователь, который должен содержать метод, демонстрирующий работу восходящего преобразования типов.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Разработана структура классов, удовлетворяющая заданию | 3 |
| Продемонстрировано восходящее преобразование при наследовании | 4 |
| Итого | 7 |

## Лабораторная работа №6. Интерфейсы

Для выполнения задания необходимо создать класс из лабораторной работы №2. Также необходимо разработать интерфейс согласно заданию. После чего необходимо разработать интерфейс и реализовать его в разработанном классе (согласно заданию).

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует основную задачу. | 4 |
| Итого | 4 |

## Лабораторная работа №7. Лямбда выражения

Разработать Лямбда функции согласно варианту. Создать делегат и при помощи групповой адресации реализовать описанный алгоритм. Решить задачу.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует задачу. | 2 |
| Итого | 2 |

## Лабораторная работа №8. Коллекции и абстрактные классы

Спроектировать структуру классов согласно задаче, базовый класс должен быть абстрактным и содержать абстрактные методы. Выполнить дополнительное задание.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует основную задачу. | 2 |
| Решена дополнительная задача | 1 |
| Итого | 3 |

## Лабораторная работа №9. Параметризация

Создать класс согласно заданию

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует основную задачу. | 2 |
| Итого | 2 |

## Лабораторная работа №10. Исключительные ситуации

Создать класс согласно заданию. Продумать исключительную ситуацию, связанную с использованием этого класса. Разработать код и продемонстрировать случаи обработки исключений.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует основную задачу. | 2 |
| Итого | 2 |

## Лабораторная работа №11. Паттерн проектирования Стратегия

Спроектировать структуру классов согласно задаче. Для решения нужно использовать паттерн проектирования стратегия. Выполнить дополнительное задание.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует основную задачу. | 3 |
| Решена дополнительная задача | 4 |
| Итого | 7 |

## Лабораторная работа №12. Паттерн проектирования Наблюдатель

Спроектировать структуру классов согласно задаче. Для решения нужно использовать паттерн проектирования наблюдатель. Выполнить дополнительное задание.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует основную задачу. | 3 |
| Решена дополнительная задача | 4 |
| Итого | 7 |

## Лабораторная работа №13. Паттерн проектирования Декоратор

Спроектировать структуру классов согласно задаче. Для решения нужно использовать паттерн проектирования декоратор. Выполнить дополнительное задание.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует основную задачу. | 3 |
| Решена дополнительная задача | 4 |
| Итого | 7 |

## Лабораторная работа №14. Паттерн проектирования Команда

Спроектировать структуру классов согласно задаче. Для решения нужно использовать паттерн проектирования команда. Выполнить дополнительное задание.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует основную задачу. | 3 |
| Решена дополнительная задача | 4 |
| Итого | 7 |

## Лабораторная работа №15. Паттерн проектирования Адаптер

Спроектировать структуру классов согласно задаче. Для решения нужно использовать паттерн проектирования Адаптер.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует основную задачу. | 4 |
| Итого | 4 |

## Лабораторная работа №16. Паттерн проектирования Фасад

Спроектировать структуру классов согласно задаче. Для решения нужно использовать паттерн проектирования Адаптер.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует основную задачу. | 4 |
| Итого | 4 |

# IV. Оценочное средство Посещение

по дисциплине «Объектно-ориентированное программирование»

***Максимально возможное количество баллов за посещение занятий в одном семестре: 6***

Количество баллов за посещение выводится по следующей формуле:



Где *x* – количество баллов за посещение;

*visited* – количество посещённых занятий в семестре;

*all* – общее число занятий в семестре.

Вычисленное значение округляется по стандартным правилам.

Например, если всего было проведено 20 занятий, а студент посетил 16 из них, то количество баллов за посещение будет равно:

, после округления и получается 5 баллов за посещение занятий.

# V. Оценочное средство Задание на СРС

по дисциплине «Объектно-ориентированное программирование»

## СРС №1. Ввод – вывод в C#. Работа с файлами

Разработать программный код, который будет решать лабораторной работы №1, согласно полученному варианту (см. Приложение №1). Программный код должен получать входные данные из файла, имя которого передаётся в качестве входного параметра консоли.

Файл с входными данными представляет собой текстовый файл следующей структуры:

*Директория выходного файла*

*Имя выходного файла*

*<Входные данные для задачи>*

Программный код должен проверить существует ли директория указанная во входных данных в текущей директории. Если такая директория не существует, то её нужно создать (в случае если директория выходного файла не указана, файл должен быть создан в текущей директории). После чего нужно проверить существует ли в указанной директории файл с именем указанным во входных данных. Если файл не существует, то его нужно создать файл, если файл существует, то его нужно будет перезаписать.

В выходном файле должен содержаться, отчёт следующей структуры:

Directory Name: *Директория выходного файла*

File Name: *Имя выходного файла*

Directory: *<параметр отчёта №1>*

File: *<параметр отчёта №2>*

*Решение задачи*

Пояснения: Директория выходного файла и имя выходного файла содержатся в входных данных.

Параметр отчёта №1: *created* – если директория была создана, *exists* – если директория существовала.

Параметр отчёта №2: *created* – если файл был создан, *exists* – если файл существовал.

Решение задачи содержит в себе ответ на задачу

**Пример задачи**

Дано два целых числа, вывести их сумму.

**Пример входного файла** *(input.txt)*

*aritmetics*

*sum.txt*

*4 5*

Согласно входным данным, файл-отчёт должен содержаться в директории *aritmetics* и его имя должно быть *sum.txt*. Допустим такая директория существует, а файл пришлось создать. В таком случае содержание выходного файла выглядит следующим образом:

*Directory Name: aritmetics*

*File Name: sum.txt*

*Directory: exists*

*File: created*

*9*

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует основную задачу. | 5 |
| Итого | 5 |

СРС № 2. Сериализация объектов

Разработать программный класс из лабораторной работы №2, соответствующей варианту (см. Приложение №1). Разработать программный код, который в зависимости от входных данных будет либо создавать объект с определёнными параметрами, и проводить его сериализацию, после чего записывать в файл, либо загружать сериализованный объект и писать данные о нём в отчётный файл.

Файл с входными данными представляет собой текстовый файл следующей структуры:

*Имя файла*

*Операция*

*<Входные данные для задачи>*

Пояснения:

В качестве параметра операция может быть либо строка *serialize*, либо строка *unserialize*.

*serialize –* означает, что нужно создать объект и заполнить его данными для задачи, после чего сохранить в файл с именем переданным указанным во входных данных.

*unserialize –* означает, что нужно создать объект и заполнить его данными полученными из файла указанного во входных данных. После чего вызвать вызывать методы объекта.

**Пример задачи**

Создать класс Cat, который будет содержать информацию о коте: Кличка, возраст. Класс должен содержать метод info(), который выводит в консоль информацию об объекте класса (кличку и возраст) в следующем формате: name: *<кличка>,* age: *<возраст>*.

Пример входного файла для сериализации:

*inputdata.dat*

*serialize*

*Tom 3*

Пояснение: в данном случае входные данные означают, что нужно создать объект класса Cat с параметрами Кличка = Tom, возраст = 3. После чего сереализовать его и записать в файл *inputdata.dat.*

Пример входного файла для загрузки данных:

*inputdata.dat*

*unserialize*

Пояснение: в данном случае входные данные означают, что нужно получить объект класса Сat из файла *inputdata.dat*. После чего вызвать у объекта метод *info().*

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует основную задачу. | 5 |
| Итого | 5 |

## СРС №3. Reflection API. Динамическое определение в C#

Придумать и разработать структуру классов, реализующую наследование. Базовый класс должен быть абстрактным. Каждый из производных классов (не менее 3) должны реализовывать методы, не заявленные в базовом классе (ниже будем называть их «особыми»).

В качестве задачи для разработки структуры классов взять задачу из лабораторной работы №8 (см. Приложение №1)

**Пример задачи:**

*Разработать структуру классов «Животные» - Animal. Абстрактным классом является класс Животное, субклассами являются классы Кошка («особый» метод – meow()), Собака («особый» метод – bark()), Лошадь («особый» метод – gallop()).*

Разработать класс, который будет в качестве атрибута содержать контейнер с классами, описанными выше (в случае с примером можно использовать List<Animal>).

Класс должен содержать метод, который заполнит контейнер произвольными объектами разработанных классов. Создание классов должно осуществляться при помощи методов встроенного класса Class.

Класс должен содержать метод, который перебирает все элементы контейнера и вызывает у них «особые» методы.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует основную задачу. | 5 |
| Итого | 5 |

## СРС №4. Работа с СУБД MS SQL

Разработать структуру базы данных согласно задаче из Лабораторной работы №2 (см. Приложение №1). Создать базу данных в СУБД MS SQL и соответствующие таблицы.

Написать программный код, который реализует операции для работы с БД: вставка, удаление, редактирование, чтение записи.

Дополнительная задача: разработать интерфейс с использованием windows forms.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует основную задачу. | 4 |
| Создан интерфейс для работы с базой (таблица со списком, форма добавления) | 4 |
| Итого | 8 |

## СРС №5. Паттерн Domain model

Для таблицы, разработанной в СРС №4 разработать структуру классов, реализующую паттерн Domain model и Data Mapper.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует основную задачу. | 6 |
| Итого | 6 |

## СРС №6. Паттерн Identity map

Для классов из СРС №5 разработать структуру классов, реализующую паттерн Identity map.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует основную задачу. | 6 |
| Итого | 6 |

## СРС №7. Паттерн Lazy Load

Для классов из СРС №5 разработать структуру классов, реализующую паттерн Lazy load.

|  |  |
| --- | --- |
| **Критерий** | **Максимальная оценка** |
| Программный код реализует основную задачу. | 6 |
| Итого | 6 |

# 

# VII. Оценочное средство зачёт/экзамен

по предмету «Объектно-ориентированное программирование»

Билет для сдачи зачёта/экзамена состоит из теоретического вопроса и решения задач. В качестве условий задач участник выбирает один из вариантов, предложенных в лабораторных работах. Список теоретических вопросов приведён ниже.

## Вопросы для подготовки к зачёту

1. Классы в языке C#. Определение класса, методы, аргументы.
2. Конструкторы и инициализация
3. Перегрузка и замещение функций
4. Сборка мусора в C#
5. Наследование
6. Восходящее преобразование
7. Замещение функций при наследовании
8. Конструкторы и наследование
9. Полиморфизм
10. Конструкторы и полиморфизм
11. Абстрактные классы
12. Интерфейсы
13. Параметризация в языке C#
14. Коллекции объектов (List, List, Итераторы, Стек.)
15. Исключения в C#
16. Лямбда функции
17. Наследование от внутренних классов
18. Пространства имён

## Вопросы для подготовки к экзамену

1. Классы в языке C#. Определение класса, методы, аргументы.
2. Конструкторы и инициализация
3. Перегрузка и замещение функций
4. Сборка мусора в C#
5. Наследование
6. Восходящее преобразование
7. Замещение функций при наследовании
8. Конструкторы и наследование
9. Полиморфизм
10. Конструкторы и полиморфизм
11. Абстрактные классы
12. Интерфейсы
13. Параметризация в языке C#
14. Коллекции объектов (List, List, Итераторы, Стек.)
15. Исключения в C#
16. Лямбда функции
17. Наследование от внутренних классов
18. Пространства имён
19. Паттерн проектирования Стратегия
20. Паттерн проектирования Наблюдатель
21. Паттерн проектирования Декоратор
22. Паттерн проектирования Простая Фабрика
23. Паттерн проектирования Фабричный метод
24. Паттерн проектирования Абстрактная фабрика
25. Паттерн проектирования Команда
26. Паттерн проектирования Команда. Дополнительные возможности (отмена действий, очередь запросов)
27. Паттерн проектирования Фасад
28. Паттерн проектирования Адаптер
29. Паттерн Data Mapper и Domain Model
30. Паттерн Identity Map
31. Паттерн Lazy Load

**Критерии оценки ответа на теоретический вопрос**

|  |  |  |
| --- | --- | --- |
| **Критерий** | **Требования** | **Баллы** |
| Знание и понимание теоретического материала | Рассматриваемые понятия определяются четко и полно, приводятся соответствующие примеры | 5 |
| Используемые понятия строго соответствуют теме | 4 |
| Анализ и оценка информации | Объясняются альтернативные взгляды на рассматриваемую проблему |  |
| 4 |
| Дается личная оценка проблеме | 3 |
| Построение суждений | Изложение ясное и четкое | 2 |
| Приводятся различные точки зрения и их личная оценка (примеры применения, плюсы и минусы) | 2 |
|  | **Итого** | **20** |

Проект макета Фонда оценочных средств

утвержден Учёным Советом ИМИ

от «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 202\_\_ г., протокол № \_\_\_\_

Учёный секретарь \_\_\_\_\_\_\_\_\_\_\_\_ Телешева Л.А.

# Приложение №1. Лабораторные работы

## Формирование вариантов в лабораторных работах 1-10.

Формирование вариантов происходит согласно таблице №1. Для каждого варианта указан номер задания.

Таблица №1. Формирование вариантов

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вар.** | **Лаб.раб. №1** | **Лаб.раб. №2** | **Лаб.раб. №3** | **Лаб.раб. №4** | **Лаб.раб. №5** | **Лаб.раб. №6** | **Лаб.раб. №7** | **Лаб.раб. №8** | **Лаб.раб. №9** | **Лаб.раб. №10** |
| 1 | 1 | 1 | 1 | 10 | 1 | 1 | 1 | 1 | 1.1, 1.4 | 1 |
| 2 | 2 | 2 | 2 | 9 | 2 | 2 | 2 | 2 | 1.2, 1.4 | 2 |
| 3 | 3 | 3 | 3 | 8 | 3 | 3 | 3 | 3 | 2.1, 2.4 | 3 |
| 4 | 4 | 4 | 4 | 7 | 4 | 4 | 4 | 4 | 2.2,2.3 | 4 |
| 5 | 5 | 5 | 5 | 6 | 5 | 5 | 5 | 5 | 1.1, 1.3 | 5 |
| 6 | 6 | 6 | 6 | 5 | 6 | 6 | 6 | 6 | 1.2,1.3 | 6 |
| 7 | 7 | 7 | 7 | 4 | 7 | 1 | 7 | 1 | 2.2,2.3 | 7 |
| 8 | 8 | 8 | 8 | 3 | 8 | 2 | 8 | 2 | 2.1,2.3 | 8 |
| 9 | 9 | 9 | 9 | 2 | 9 | 3 | 9 | 3 | 1.2,1.3 | 9 |
| 10 | 10 | 10 | 10 | 1 | 10 | 4 | 10 | 4 | 2.2,2.3 | 10 |
| 11 | 11 | 11 | 1 | 10 | 1 | 5 | 11 | 5 | 2.1, 2.4 | 11 |
| 12 | 12 | 12 | 2 | 9 | 2 | 6 | 12 | 6 | 2.2,2.3 | 12 |
| 13 | 13 | 13 | 3 | 8 | 3 | 1 | 13 | 1 | 1.1, 1.4 | 13 |
| 14 | 14 | 14 | 4 | 7 | 4 | 2 | 14 | 2 | 1.2, 1.4 | 14 |
| 15 | 15 | 15 | 5 | 6 | 5 | 3 | 15 | 3 | 1.1, 1.3 | 15 |
| 16 | 16 | 16 | 6 | 5 | 6 | 4 | 16 | 4 | 1.2,1.3 | 16 |
| 17 | 17 | 17 | 7 | 4 | 7 | 5 | 17 | 5 | 2.2,2.3 | 17 |
| 18 | 18 | 18 | 8 | 3 | 8 | 6 | 18 | 6 | 2.1, 2.4 | 18 |

## Лабораторная работа №1

Решить задачу, вывод должен быть реализован согласно примеру выходных данных.

### Задача №1

Дано целое число, нужно вывести следующее и предыдущее число, согласно формату вывода.

Пример входных данных: *3*

Пример выходных данных:

*Next number for number 3 is 4.*

*Previous number for number 3 is 2.*

### Задача №2

Дано целое число (число <10000). Вывести квадрат числа согласно формату вывода

Пример входных данных: *4*

Пример выходных данных: *Square for the number 4 = 16*

### Задача №3

Пешеход вышел из пункта А и шёл со скоростью V в течении t часов. Найти расстояние, которое прошёл пешеход.

Пример входных данных: *3 5*

Пример выходных данных: *S=15*

### Задача №4

Дано число - сторона квадрата, найти площадь и периметр квардата.

Пример входных данных: *3*

Пример выходных данных:

*Square = 9*

*Perimeter = 12*

### Задача №5

В равнобедренном треугольнике дано основание и длина боковой стороны, найти периметр треугольника.

Пример входных данных: *3 4*

Пример выходных данных: Perimeter = *11*

### Задача №6

Дана длина стороны куба, найти объём куба.

Пример входных данных: *5*

Пример выходных данных: Volume =*125*

### Задача №7

Дана длина стороны треугольника и высота, проведённая к этой стороне, найти площадь треугольника.

Пример входных данных: *5 2*

Пример выходных данных: Square= *5*

### Задача №8

Прямая задаётся уравнением ax+b=y. Даны координаты точки A(ax,by). Вывести на экран Yes, если точка лежит на прямой, NO если точка не лежит на прямой. Данные передаются в следующей последовательности: a, b – коэффициенты прямой, ax, by – координаты точки А. Разделителем является пробел.

Пример входных данных: *1 0 2 3*

Пример выходных данных: *NO*

### Задача №9

Парабола задаётся уравнением ax2+bx+c=y. Даны координаты точки A(ax,by). Вывести на экран Yes, если точка лежит на параболе, NO если точка не лежит на параболе. Данные передаются в следующей последовательности: a, b,с – коэффициенты прямой, ax, by – координаты точки А. Разделителем является пробел.

Пример входных данных: *1 0 2 -1 0*

Пример выходных данных: *NO*

### Задача №10

Даны длины сторон прямоугольника, найти площадь и периметр прямоугольника.

Пример входных данных: *3 4*

Пример выходных данных

*Square = 12*

*Perimeter = 14*

### Задача №11

Дано два целых числа a,b (a<b). Вывести все чётные числа от а до b (включая числа a и b).

Пример входных данных: *-5 3*

Пример выходных данных: *-4 -2 0 2*

### Задача №12

Дано два целых числа a,b (a<b). Вывести все нечётные числа от а до b (включая числа a и b).

Пример входных данных: *-5 3*

Пример выходных данных: *-5 -3 -1 1 3*

### Задача №13

Дано два целых числа a,b (a<b). Найти сумму всех целых чисел от a до b (включая числа a и b).

Пример входных данных: *0 3*

Пример выходных данных: *6*

### Задача №14

Дано три числа, найти максимальное из них.

Пример входных данных: *0 3 1*

Пример выходных данных: *3*

### Задача №15

Дано три числа, найти минимальное из них.

Пример входных данных: *0 3 1*

Пример выходных данных: *0*

### Задача №16

Дано два числа, если первое больше второго вывести >, если первое меньше второго вывести <, если числа равны вывести =.

Пример входных данных: *0 5*

Пример выходных данных: <

### Задача №17

Дано два целых числа a,b (a<b). Найти все числа от а до b, кратные 5 (включая числа a и b).

Пример входных данных: *0 19*

Пример выходных данных: *5 10 15*

### Задача №18

Дано два целых числа a,b (a<b). Найти все числа от а до b, не кратные 3 (включая числа a и b).

Пример входных данных: *0 7*

Пример выходных данных: *1 2 4 5 7*

## Лабораторная работа №2

Создайте класс согласно задаче, определите конструктор (в конструкторе должны задаваться атрибуты класса), протестируйте класс в программе. Класс должен содержать конструктор класса и хотя бы один перегруженный метод.

### Задача №1

Создайте класс погодная станция. Класс должен содержать атрибуты: температура, влажность, давление (все атрибуты — числа) и методы для их изменения. Класс также должен содержать метод, который выводит информацию (значения атрибутов) на экран (в консоль).

### Задача №2

Создайте класс Товар. Класс должен содержать атрибуты: название (строка), цена (число) и методы для их изменения. Класс также должен содержать метод, который выводит информацию (значения атрибутов) на экран (в консоль).

### Задача №3

Создайте класс Робот. Класс должен содержать атрибуты: координаты местоположения в двумерном пространстве (целочисленные координаты на плоскости). Класс должен содержать метод передвижение, аргументы этого метода — направление, которое задаётся числовым кодом (1 - влево, 2 - вправо, 3 - вниз, 4 — вверх) и количество шагов. Класс должен содержать метод который выводит на экран текущие координаты робота.

### Задача №4

Создайте класс Прямоугольник. Класс должен содержать атрибуты: координаты левого верхнего и правого нижнего угла в декартовой системе координат. Класс должен содержать методы вычисления площади и периметра прямоугольника.

### Задача №5

Создайте класс Треугольник. Класс должен содержать атрибуты: длины сторон. Класс должен содержать методы вычисления площади и периметра треугольника.

### Задача №6

Создайте класс Круг. Класс должен содержать атрибут: радиус круга. Класс должен содержать методы вычисления длины окружности, которая является границей круга и вычисления площади круга.

### Задача №7

Создайте класс Вектор. Класс должен содержать атрибуты: координаты вектора в декартовой системе координат. Класс должен содержать метод вычисления длины вектора.

### Задача №8

Создайте класс Телевизор. Класс должен содержать атрибуты: текущее состояние (включен/выключен), текущий канал. Класс должен содержать метод включения, выключения (изменение атрибута текущее состояние), также класс должен содержать метод изменения текущего канала и метод выводящий информацию о текущем состоянии телевизора (если телевизор выключен вывести просто выключен, если включен то вывести текущий канал).

### Задача №9

Создайте класс Радиоприёмник. Класс должен содержать атрибуты: текущее состояние (включен/выключен), текущий канал. Класс должен содержать метод включения, выключения (изменение атрибута текущее состояние), также класс должен содержать метод изменения текущего канала и метод выводящий информацию о текущем состоянии приёмника (если приёмник выключен вывести просто выключен, если включен то вывести текущий канал).

### Задача №10

Создайте класс Ячейка камеры хранения. Класс должен содержать атрибуты: текущее состояние (свободна/занята), предмет в ячейке (строка). Класс должен содержать метод, который позволит положить в ячейку предмет и метод выводящий информацию о текущем состоянии ячейки(если ячейка пуста вывести empty, иначе вывести название предмета в ячейке).

### Задача №11

Создайте класс Равнобедренный Треугольник. Класс должен содержать атрибуты: высота и основание. Класс должен содержать методы вычисления площади и периметра треугольника.

### Задача №12

Создайте класс равнобедренный Треугольник. Класс должен содержать атрибуты: высота и боковая сторона. Класс должен содержать методы вычисления площади и периметра треугольника.

### Задача №13

Создайте класс Кинофильм. Класс должен содержать атрибуты: категория (три категории А, В, С), цена билета. Класс должен содержать метод вычисления цены билета от количества дней со дня премьеры. Цена на билеты фильмов категории С снижается на 30% через неделю после премьеры, В на 20% через 10 дней после премьеры, А на 15% через 2 неделе после премьеры.

### Задача №14

Создайте класс Дробь. Класс должен содержать атрибуты: числитель и знаменатель. Класс должен содержать метод сокращения дроби и/или метод добавления к дроби другой дроби.

### Задача №15

Создайте класс Персональный компьютер. Класс должен содержать текстовое описание и цену составляющих компьютера: Системный блок, Монитор, Клавиатура, Мышь. Класс должен содержать методы вывода общей цены и общего описания (через пробел описание всех составляющих).

### Задача №16

Создайте класс Комплексное число. Класс должен содержать атрибуты: действительная и мнимая часть. Класс должен содержать метод вычисления модуля комплексного числа, метод умножения на скаляр и метод вывода на экран (пример: *4-2i*).

### Задача №17

Создайте класс Будильник. Класс должен содержать атрибуты: значение минут, значение часов, а также время, на которое поставлен будильник. Класс должен содержать методы: проверки, сколько осталось до срабатывания будильника, метод подачи звукового сигнала (вывести в консоль ring).

### Задача №18

Создайте класс Треугольник. Класс должен содержать атрибуты: значение двух углов треугольника и стороны. Класс должен содержать метод, который определяет тип треугольника (прямоугольный, остроугольный, тупоугольный, равносторонний, равнобедренный, не существует), а также метод, который определяет его площадь.

## Лабораторная работа №3. Наследование

В задании необходимо разработать два класса: родитель и наследник, класс наследник должен переопределить метод родителя. В родительском классе должен быть описан конструктор с непустым набором аргументов.

### Задача №1

Разработать базовый класс: товар. Товар содержит атрибуты: базовая стоимость товара, название товара. Товар содержит метод вывода информации о товаре, метод должен выводить в консоль информацию о товаре. Пример вывода информации:

Персональный компьютер ………… 1000 рублей.

Разработать наследника класса Товар: Продукты питания. Продукты питания содержат дополнительный атрибут: количество дней до истечения срока годности. Итоговая стоимость продукта питания снижается на 20%, когда до истечения срока годности остаётся меньше 10 дней, и ещё на 50% когда до истечения срока годности остаётся меньше 4 дней.

### Задача №2

Разработать базовый класс: товар. Товар содержит атрибуты: базовая стоимость товара, название товара. Товар содержит метод вывода информации о товаре, метод должен выводить в консоль информацию о товаре. Пример вывода информации:

Персональный компьютер ………… 1000 рублей.

Разработать наследника класса Товар: Товар в строительном магазине. Товары в строительном магазине содержат дополнительные атрибуты: сезонная скидка (в процентах) и тип материала (три типа материалов: стройматериалы, электротехника, сантехника). В зависимости от типа материала скидка работает по разному: стройматериалы – полная скидка, электротехника – половина скидки, сантехника – треть скидки (то есть если сезонная скидка равна 30% то на стройматериалы будет 30%, на электротехнику – 15%, на сантехнику 10%).

### Задача №3

Разработать базовый класс: товар. Товар содержит атрибуты: базовая стоимость товара, название товара. Товар содержит метод вывода информации о товаре, метод должен выводить в консоль информацию о товаре. Пример вывода информации:

Персональный компьютер ………… 1000 рублей.

Разработать наследника класса Товар: системный блок. Под базовой ценой системного блока понимается цена корпуса. Также системный блок содержит дополнительные атрибуты: стоимость материнской платы, стоимость жёсткого диска, стоимость видеокарты, стоимость оперативной памяти.

### Задача №4

Разработать базовый класс: товар. Товар содержит атрибуты: базовая стоимость товара, название товара. Товар содержит метод вывода информации о товаре, метод должен выводить в консоль информацию о товаре. Пример вывода информации:

Персональный компьютер ………… 1000 рублей.

Разработать наследника класса Товар: бытовая техника. Бытовая техника содержит дополнительный атрибут – дополнительная гарантия. Стоимость бытовой техники складывается из базовой стоимости и стоимости дополнительной гарантии. При этом год гарантии стоит 3% от стоимости товара, два года – 5%, три года – 6%.

### Задача №5

Разработать базовый класс: товар. Товар содержит атрибуты: базовая стоимость товара, название товара. Товар содержит метод вывода информации о товаре, метод должен выводить в консоль информацию о товаре. Пример вывода информации:

Персональный компьютер ………… 1000 рублей.

Разработать наследника класса Товар: DVD диск. DVD диск содержит дополнительные атрибуты: количество дней с даты выхода в продажу, категория (три категории: A, B, C).

Категория С получает скидку в 30% через 30 дней после выхода в продажу, категория В получает скидку в 25% через 45 дней после выхода в продажу, категория А получает скидку в 10% через 60 дней после выхода в продажу.

### Задача №6

Разработать базовый класс: транспортное средство. Класс содержит атрибуты: базовая стоимость аренды в час, название. Класс содержит метод вывода информации о стоимости аренды (на вход метод принимает время аренды в часах), метод должен выводить информацию в консоль. Пример вывода информации:

Грузовой автомобиль Камаз ……….. 15000 рублей.

Разработать наследника класса: легковой автомобиль. Легковой автомобиль содержит дополнительные атрибуты: комплектация автомобиля (эконом – 100% от стоимости, бизнес – 120 % от стоимости, премиум – 200% от стоимости) и стоимость дополнительных услуг. Стоимость аренды легкового автомобиля считается как сумма аренды (в зависимости от комплектации) и стоимости дополнительных услуг.

### Задача №7

Разработать базовый класс: транспортное средство. Класс содержит атрибуты: базовая стоимость аренды в час, название. Класс содержит метод вывода информации о стоимости аренды (на вход метод принимает время аренды в часах), метод должен выводить информацию в консоль. Пример вывода информации:

Грузовой автомобиль Камаз ……….. 15000 рублей.

Разработать наследника класса: грузовой автомобиль. Грузовой автомобиль содержит дополнительный атрибут: вес груза. Стоимость аренды считается как сумма аренды и сумма за вес груза (за каждый килограмм 100 рублей).

### Задача №8

Разработать базовый класс: транспортное средство. Класс содержит атрибуты: базовая стоимость аренды в час, название. Класс содержит метод вывода информации о стоимости аренды (на вход метод принимает время аренды в часах), метод должен выводить информацию в консоль. Пример вывода информации:

Грузовой автомобиль Камаз ……….. 15000 рублей.

Разработать наследника класса: Автобус. Автобус содержит дополнительные атрибуты: количество пассажиров и количество мест в автобусе. Стоимость аренды считается как сумма базовой стоимости и комиссионного сбора. В случае наполненности автобуса на 30% и менее комиссионный сбор составляет 2000 рублей, если наполненность менее 50% то 3500 рублей, от 50% до 75% - 4000 рублей, если более 75% то 4500 рублей.

### Задача №9

Разработать базовый класс: транспортное средство. Класс содержит атрибуты: базовая стоимость аренды в час, название. Класс содержит метод вывода информации о стоимости аренды (на вход метод принимает время аренды в часах), метод должен выводить информацию в консоль. Пример вывода информации:

Грузовой автомобиль Камаз ……….. 15000 рублей.

Разработать наследника класса: Вертолёт. Вертолёт содержит дополнительные атрибуты: количество пассажиров (не более 5) и вес груза. Каждый пассажир платит 5000 рублей, за каждый килограмм груза доплачивается 1000 рублей.

### Задача №10

Разработать базовый класс: транспортное средство. Класс содержит атрибуты: базовая стоимость аренды в час, название. Класс содержит метод вывода информации о стоимости аренды (на вход метод принимает время аренды в часах), метод должен выводить информацию в консоль. Пример вывода информации:

Грузовой автомобиль Камаз ……….. 15000 рублей.

Разработать наследника класса: Катер. Катер содержит дополнительный атрибут: расстояние до пункта назначения. Стоимость аренды считается следующим образом: базовая стоимость аренды умножается на 1.5, за каждый километр расстояния до пункта назначения платят 450 рублей.

## Лабораторная работа №4. Композиция

Реализовать описанный в задании класс. На основании класса реализовать класс решающий задачу.

### Задача №1

Разработать класс точка в декартовой системе координат (n=3). На основе этого класса разработать класс Треугольник. Треугольник описывается тремя точками - вершинами. Класс должен содержать метод вычисления площади треугольника.

### Задача №2

Разработать класс точка в декартовой системе координат (n=3). На основе этого класса разработать класс Треугольник. Треугольник описывается тремя точками - вершинами. Класс должен содержать метод вычисления периметра треугольника.

### Задача №3

Разработать класс точка в декартовой системе координат (n=3). На основе этого класса разработать класс Треугольник. Треугольник описывается тремя точками - вершинами. Класс должен содержать метод определения типа треугольника (равносторонний, равнобедренный, разносторонний).

### Задача №4

Разработать класс точка в декартовой системе координат (n=3). На основе этого класса разработать класс Треугольник. Треугольник описывается тремя точками - вершинами. Класс должен содержать метод определения типа углов треугольника.

### Задача №5

Разработать класс точка в декартовой системе координат (n=3). На основе этого класса разработать класс Прямоугольник. Прямоугольник описывается тремя точками – левая нижняя, правая верхняя вершина и третья точка для определения прямоугольника в пространстве. Класс должен содержать метод вычисления площади прямоугольника.

### Задача №6

Разработать класс точка в декартовой системе координат (n=3). На основе этого класса разработать класс Прямоугольник. Прямоугольник описывается двумя точками – левая нижняя, правая верхняя вершина и третья точка для определения прямоугольника в пространстве. Класс должен содержать метод вычисления периметра прямоугольника.

### Задача №7

Разработать класс точка в декартовой системе координат (n=3). На основе этого класса разработать класс Параллелограмм. Прямоугольник описывают четыре точки – вершины параллелограмма. Класс должен содержать метод вычисления площади параллелограмма.

### Задача №8

Разработать класс точка в декартовой системе координат (n=3). На основе этого класса разработать класс Параллелограмм. Параллелограмм описывают четыре точки – вершины параллелограмма. Класс должен содержать метод вычисления периметра параллелограмма.

### Задача №9

Разработать класс точка в декартовой системе координат (n=3). На основе этого класса разработать класс Трапеция. Трапеция описывают четыре точки – вершины Трапеции. Класс должен содержать метод вычисления периметра Трапеции.

### Задача №10

Разработать класс точка в декартовой системе координат (n=3). На основе этого класса разработать класс Трапеция. Трапеция описывают четыре точки – вершины Трапеции. Класс должен содержать метод вычисления площади Трапеции.

## Лабораторная работа №5. Наследование

Разработать структуру классов, моделирующую описанную задачу. Тестирование классов произвести в отдельном классе с методом – точкой входа в приложение, в этом классе также реализовать дополнительный метод, функционал метода описан в задаче.

### Задача №1

Создать иерархию классов для Транспортного средства. Все классы в иерархии имеют следующие параметры: модель (название транспорта), вес, скорость. Все классы должны иметь следующие методы:

1. вычисление стоимости аренды транспорта в час, без учёта стоимости топлива
2. вычисление затрат топлива в час;

Создать производный класс Грузовик, характеризуемый также грузоподъемностью (в тоннах) и весом перевозимого груза. Стоимость аренды Грузовика зависит от грузоподъёмности (10 рублей за каждый килограмм в час). Затраты топлива зависят от текущего груза (10 литров за каждые 100 килограмм в час).

Создать производный класс Автобус, характеризуемый также вместимостью (количество сидячих мест) и коэффициентом комфортабельности. Стоимость аренды Автобуса зависит от его вместимости (15 рублей за каждое место) и умножается на коэффициент комфортабельности. Затраты топлива зависят от времени (10 литров на каждые 50 километров).

Создать класс, содержащий метод, который позволит вычислить стоимость аренды транспорта с учётом стоимости топлива. На вход данный метод должен принимать объект транспорта, время аренды (в часах) и стоимость литра топлива. Метод должен возвращать число – стоимость аренды конкретного транспорта.

### Задача №2

Создать иерархию классов для моделирования геометрических фигур. Каждая фигура закрашивается по определённым правилам (области не смешиваются). Для закрашивания доступно три цвета: красный, синий или зелёный. Граница фигуры имеет постоянную толщину и закрашивается в белый цвет. Все классы имеют методы:

1. вычисления площади фигуры;
2. вычисления длины границы фигуры.

Создать производный класс Треугольник. Треугольник характеризуется длинами сторон. При покраске Треугольник разделяют на три равные (по площади) части и красят в три доступных цвета.

Создайте класс Круг. Класс должен содержать атрибут: радиус круга. Круг красят в один из доступных цветов, для определения цвета в круге нужно определить дополнительный атрибут класса.

Создать класс, содержащий метод, который позволит вычислить количество миллилитров каждой краски (красная, синяя, зелёная и белая). На вход данный метод должен принимать объект фигуры, затраты краски на 1 кв. см. Метод должен вывести в консоль четыре числа – необходимое количество красной, синей, зелёной и белой краски.

### Задача №3

Создать иерархию классов для моделирования геометрических фигур. Каждая фигура закрашивается по определённым правилам (области не смешиваются). Для закрашивания доступно три цвета: красный, синий или зелёный. Граница фигуры имеет постоянную толщину и закрашивается в белый цвет. Все классы имеют методы:

1. вычисления площади фигуры;
2. вычисления длины границы фигуры.

Создать производный класс Прямоугольник. Прямоугольник характеризуется длинами сторон, также прямоугольник имеет дополнительные атрибуты – процентное количество синего, красного и зелёного цвета. Согласно этим значениям происходит покраска прямоугольника.

Создайте класс Равнобедренный треугольник. Класс должен содержать атрибут: боковая сторона и основание. Равнобедренный треугольник делят на две равные части и красят в два цвета зелёный и синий.

Создать класс, содержащий метод, который позволит вычислить количество миллилитров каждой краски (красная, синяя, зелёная и белая). На вход данный метод должен принимать объект фигуры, затраты краски на 1 кв. см. Метод должен вывести в консоль четыре числа – необходимое количество красной, синей, зелёной и белой краски.

### Задача №4

Создать иерархию классов для моделирования геометрических фигур. Каждая фигура закрашивается по определённым правилам (области не смешиваются). Для закрашивания доступно три цвета: красный, синий или зелёный. Граница фигуры имеет постоянную толщину и закрашивается в белый цвет. Все классы имеют методы:

1. вычисления площади фигуры;
2. вычисления длины границы фигуры.

Создать производный класс квадрат. Квадрат характеризуется длиной стороны. При покраске квадрат делится на 4 равные части. Три из них красят в доступные цвета (красный, синий, зелёный), одну оставляют незакрашенной.

Создайте класс треугольник. Класс должен содержать атрибут: два угла и сторона. Треугольник красят в зависимости от его типа. Равносторонний – делят на три равные части (каждую часть красят в разные цвета), равнобедренный на две части (их красят зелёным и синим цветом), прямоугольный (красят полностью в красный цвет), в другом случае треугольник не красят.

Создать класс, содержащий метод, который позволит вычислить количество миллилитров каждой краски (красная, синяя, зелёная и белая). На вход данный метод должен принимать объект фигуры, затраты краски на 1 кв. см. Метод должен вывести в консоль четыре числа – необходимое количество красной, синей, зелёной и белой краски.

### Задача №5

Создать иерархию классов для моделирования геометрических фигур. Каждая фигура закрашивается по определённым правилам (области не смешиваются). Для закрашивания доступно три цвета: красный, синий или зелёный. Граница фигуры имеет постоянную толщину и закрашивается в белый цвет. Все классы имеют методы:

1. вычисления площади фигуры;
2. вычисления длины границы фигуры.

Создать производный класс Треугольник. Треугольник характеризуется длиной двух сторон и углом между ними. Треугольник красят в зависимости от его типа. Равносторонний – делят на три равные части (каждую часть красят в разные цвета), равнобедренный на две части (их красят зелёным и синим цветом), в другом случае треугольник красят в красный цвет.

Создайте класс Трапеция. Класс должен содержать атрибут: длины оснований и высота. трапецию красят в один из доступных цветов, для определения цвета в трапеции нужно определить дополнительный атрибут класса.

Создать класс, содержащий метод, который позволит вычислить количество миллилитров каждой краски (красная, синяя, зелёная и белая). На вход данный метод должен принимать объект фигуры, затраты краски на 1 кв. см. Метод должен вывести в консоль четыре числа – необходимое количество красной, синей, зелёной и белой краски.

### Задача №6

Создать иерархию классов для Транспортного средства. Все классы в иерархии имеют следующие параметры: модель (название транспорта), вес, максимальная скорость. Все классы должны иметь следующие методы:

1. вычисление стоимости аренды транспорта в час, без учёта стоимости топлива
2. вычисление затрат топлива в час;

Создать производный класс Легковой автомобиль, характеризуемый также набором прилагающихся услуг и базовой стоимостью аренды. Стоимость аренды Грузовика зависит от набора услуг (эконом – 100% от стоимости, бизнес – 120 % от стоимости, премиум – 200% от стоимости). Затраты топлива зависят от скорости автомобиля (менее 150 – 1 литр в час, менее 200 – 1,5 литра в час, менее 300 – 2 литра в час).

Создать производный класс Вертолёт, характеризуемый также вместимостью (количество мест) и общим весом (пассажиры и груз). Стоимость аренды Вертолёта зависит от количества пассажиров (4000 рублей за каждое место). Затраты топлива зависят от веса (10 литров в час на каждые 10 килограмм веса).

Создать класс, содержащий метод, который позволит вычислить стоимость аренды транспорта с учётом стоимости топлива. На вход данный метод должен принимать объект транспорта, время аренды (в часах) и стоимость литра топлива. Метод должен возвращать число – стоимость аренды конкретного транспорта.

### Задача №7

Создать иерархию классов для моделирования объёмных геометрических фигур. Каждая фигура закрашивается по определённым правилам (области не смешиваются). Для закрашивания доступно три цвета: красный, синий или зелёный. Внутри фигура заполняется белой краской. Все классы имеют методы:

1. вычисления площади поверхности фигуры;
2. вычисления объёма фигуры.

Создать производный класс правильный тэтраэдр. Тэтраэдр характеризуется длиной ребра. Три грани тэтраэдра красят доступными цветами, одну не красят.

Создайте класс Куб. Класс должен содержать атрибут: длина стороны. Также куб имеет дополнительные атрибуты – процентное количество синего, белого и зелёного цвета. Покраска осуществляется согласно процентному количеству соответствующего цвета.

Создать класс, содержащий метод, который позволит вычислить количество миллилитров каждой краски (красная, синяя, зелёная и белая). На вход данный метод должен принимать объект фигуры, затраты краски на 1 кв. см. Метод должен вывести в консоль четыре числа – необходимое количество красной, синей, зелёной и белой краски.

### Задача №8

Создать иерархию классов для моделирования объёмных геометрических фигур. Каждая фигура закрашивается по определённым правилам (области не смешиваются). Для закрашивания доступно три цвета: красный, синий или зелёный. Внутри фигура заполняется белой краской. Все классы имеют методы:

1. вычисления площади поверхности фигуры;
2. вычисления объёма фигуры.

Создать производный класс Сфера. Сфера характеризуется длиной радиуса, также сфера содержит информацию о цвете, в который она будет покрашена (сфера полностью красится в этот цвет).

Создайте класс Прямоугольный параллепипед. Класс должен содержать атрибут: длины сторон. Площадь параллепипеда окрашивается во все три света в равных пропорциях.

Создать класс, содержащий метод, который позволит вычислить количество миллилитров каждой краски (красная, синяя, зелёная и белая). На вход данный метод должен принимать объект фигуры, затраты краски на 1 кв. см. Метод должен вывести в консоль четыре числа – необходимое количество красной, синей, зелёной и белой краски.

### Задача №9

Создать иерархию классов для моделирования объёмных геометрических фигур. Каждая фигура закрашивается по определённым правилам (области не смешиваются). Для закрашивания доступно три цвета: красный, синий или зелёный. Внутри фигура заполняется белой краской. Все классы имеют методы:

1. вычисления площади поверхности фигуры;
2. вычисления объёма фигуры.

Создать производный класс прямой цилиндр. Цилиндр характеризуется длиной радиуса основания и высотой. Основания цилиндра красят в синий и зелёный цвет, цилиндрическую поверхность красят в красный цвет.

Создайте класс Сфера. Сфера характеризуется длиной радиуса. Внутри сферы есть три секции для красной, зелёной и синей краски, объём секций задаётся в виде процентов для каждой сферы.

Создать класс, содержащий метод, который позволит вычислить количество миллилитров каждой краски (красная, синяя, зелёная и белая). На вход данный метод должен принимать объект фигуры, затраты краски на 1 кв. см. Метод должен вывести в консоль четыре числа – необходимое количество красной, синей, зелёной и белой краски.

### Задача №10

Создать иерархию классов для моделирования объёмных геометрических фигур. Каждая фигура закрашивается по определённым правилам (области не смешиваются). Для закрашивания доступно три цвета: красный, синий или зелёный. Внутри фигура заполняется белой краской. Все классы имеют методы:

1. вычисления площади поверхности фигуры;
2. вычисления объёма фигуры.

Создать производный класс Конус. Цилиндр характеризуется длиной радиуса основания и высотой. Основание цилиндра красят в синий цвет, боковую поверхность красят в красный цвет.

Создать производный класс правильный тэтраэдр. Тэтраэдр характеризуется длиной ребра. Также тэтраэдр имеет дополнительные атрибуты – процентное количество синего, белого и зелёного цвета. Три грани тэтраэдра красят доступными цветами, четвёртую грань красят в соответствии со значением атрибутов.

Создать класс, содержащий метод, который позволит вычислить количество миллилитров каждой краски (красная, синяя, зелёная и белая). На вход данный метод должен принимать объект фигуры, затраты краски на 1 кв. см. Метод должен вывести в консоль четыре числа – необходимое количество красной, синей, зелёной и белой краски.

## Лабораторная работа №6. Интерфейсы и абстрактные классы

Для выполнения задания необходимо создать класс из лабораторной работы №2. Также необходимо разработать интерфейс согласно заданию. После чего необходимо разработать интерфейс и реализовать его в разработанном классе (согласно заданию).

### Задача №1

Разработать интерфейс TXTFileSaver. Интерфейс должен содержать метод save для загрузки значений атрибутов класса в текстовый файл (каждый атрибут на новой строке).

### Задача №2

Разработать интерфейс CSVFileSaver. Интерфейс должен содержать метод save(char c) для загрузки значений атрибутов класса в csv файл (атрибуты класса записываются в одной строке, через разделитель - *c*).

### Задача №3

Разработать интерфейс TXTFileLoader. Интерфейс должен содержать метод load для загрузки значений атрибутов класса из текстового файла (каждый атрибут на новой строке).

### Задача №4

Разработать интерфейс CSVFileLoader. Интерфейс должен содержать метод load (char c) для загрузки значений атрибутов класса из csv файла (атрибуты класса записываются в одной строке, через разделитель - *c*).

### Задача №5

Разработать интерфейс ObjectInput. Интерфейс должен содержать метод input () метод должен выводить на экран консоли запрос на ввод значений атрибутов класса.

### Задача №6

Разработать интерфейс ObjectOutput. Интерфейс должен содержать метод output () метод должен выводить на экран консоли параметры класса согласно примеру:

|  |  |
| --- | --- |
| **Общий шаблон вывода** | **Пример вывода класса Cat** |
| ----- <Имя класса> -----------  | <attr 1 name>: <attr value> |  | <attr 2 name>: <attr value> |  ----------------------------------- | ----- Cat -----  | name:Tom |  | age:10 |  ---------------- |

## Лабораторная работа №7. Лямбда выражения

Разработать Лямбда функции согласно варианту. Создать делегат и при помощи групповой адресации реализовать описанный алгоритм. Решить задачу.

### Задача №1

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. прибавь 1**

**2. умножь на 2**

**3. возведи в квадрат**

Исполнитель выполнил программу 1, 3, 2, 1, 2 пять раз. Какой результат будет получен для числа 24.

### Задача №2

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. прибавь 3**

**2. увеличь число десятков на 1**

Примечание: команда 2 преобразует число 23 в 33. Если перед выполнением команды 2 вторая с конца цифра равна 9, она не изменяется, то есть число 91 останется 91, также, как и 191 останется 191.

Исполнитель выполнил программу 2, 1, 1, 2, 2 семь раз. Какой результат будет получен для числа 142.

### Задача №3

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. прибавь 7**

**2. увеличь первые три разряда числа на 1**

Примечание: команда 2 преобразует число 23 в 34, а 129 в 239 (так как младший разряд нельзя увеличить). Если перед выполнением команды 2 какая-либо цифра равна 9, она не изменяется.

Исполнитель выполнил программу 2, 2, 1, 1, 2 шесть раз. Какой результат будет получен для числа 45.

### Задача №4

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. прибавь 31**

**2. умножь на 1,5**

Примечание: команда 2 умножает чётные числа на 1,5, нечётные числа команда 2 не меняет. То есть 2 будет преобразовано в 3, а 3 «преобразуется» в 3.

Исполнитель выполнил программу 1, 1, 2, 2, 1 десять раз. Какой результат будет получен для числа 134.

### Задача №5

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. прибавь 1**

**2. сделай чётное**

**3. сделай нечётное**

Примечание: команда 1 увеличивает число на 1, вторая умножает это число на 2, третья переводит число x в число 2x + 1. Например, вторая команда переводит число 10 в число 20, а третья переводит число 10 в число 21.

Исполнитель выполнил программу 1, 3, 3, 2, 2,1,1 пять раз. Какой результат будет получен для числа 17.

### Задача №6

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. прибавь 1**

**2. прибавь 2**

**3. прибавь 5**

Исполнитель выполнил программу 3, 1, 3, 2, 3,1,1 двенадцать раз. Какой результат будет получен для числа 371.

### Задача №7

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. прибавь 1**

**2. прибавь 2**

**3. прибавь предыдущее**

Примечание: команда 3 прибавляет к числу N, число N-1. То есть к число 3 будет преобразовано в 5 (3+2).

Исполнитель выполнил программу 1, 1, 3, 2, 3,1,2 семь раз. Какой результат будет получен для числа 31.

### Задача №8

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. Прибавить 1**

**2. Умножить на 2**

**3. Прибавить 3**

Исполнитель выполнил программу 1, 3, 1, 2, 3,1,2 тринадцать раз. Какой результат будет получен для числа 311.

### Задача №9

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. Прибавить 1**

**2. Умножить на 2**

**3. Умножить на 3**

Исполнитель выполнил программу 3, 2, 1, 3, 1,2,2 двенадцать раз. Какой результат будет получен для числа 131.

### Задача №10

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. Прибавить 1**

**2. Прибавить 2**

**3. Умножить на 3**

Исполнитель выполнил программу 1, 2, 2, 3, 1,3,2 семь раз. Какой результат будет получен для числа 461.

### Задача №11

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. Прибавить 1**

**2. Сделай нечётное**

Примечание: вторая команда преобразует число N в число *2N+1*

Исполнитель выполнил программу 1, 2, 2, 1, 1,2,2 девять раз. Какой результат будет получен для числа 61.

### Задача №12

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. Вычесть 1**

**2. Вычесть 3**

**3. Разделить нацело на 3**

Примечание: При выполнении команды 3 выполняется деление нацело (остаток отбрасывается).

Исполнитель выполнил программу 1, 3, 2, 3, 1,3,2 девять раз. Какой результат будет получен для числа 456.

### Задача №13

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. Вычесть 1**

**2. Вычесть 3**

**3. Вычесть остаток от деления на 43**

Примечание: Команда 3 выполняется только для чисел, больших, чем 43. Если число меньше 43, то команда не изменяет число.

Исполнитель выполнил программу 1, 2, 3, 3, 1,3,2 девять раз. Какой результат будет получен для числа 345.

### Задача №14

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. Вычесть 3**

**2. Вычесть 10**

**3. Разделить нацело на 2**

Исполнитель выполнил программу 1, 2, 3, 3, 1,3,2 пять раз. Какой результат будет получен для числа 1245.

### Задача №15

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. Прибавить 3**

**2. Умножить на 2**

**3. Прибавить остаток от деления на 43**

Примечание: Третья команда добавляет к числу остаток деления этого числа на 43

Исполнитель выполнил программу 1, 3, 2, 1, 1, 2, 2 пять раз. Какой результат будет получен для числа 115.

### Задача №16

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. Прибавить 2**

**2. Прибавить предыдущее**

**3. Прибавить следующее**

Примечание: Вторая команда прибавляет к числу N число N-1, Третья команда добавляет к числу N число N+1.

Исполнитель выполнил программу 1, 3, 2, 1, 2, 1, 3 семь раз. Какой результат будет получен для числа 127.

### Задача №17

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. Прибавить 2**

**2. Прибавить 3**

**3. Дописать к числу справа 1**

Примечание: Третья команда дописывает к числу 1 справа, например, число 123 она изменит на 1231.

Исполнитель выполнил программу 1, 2, 2, 3, 2, 1, 3 семь раз. Какой результат будет получен для числа 27.

### Задача №18

Исполнитель реализует функции, которые принимают на вход целое число и возвращают целое число. Функциям присвоены номера:

**1. Прибавить 2**

**2. Дописать к числу справа 3**

**3. Вычесть из числа 3**

Примечание: вторая команда дописывает к числу 3 справа, например, число 34 она изменит на 343.

Исполнитель выполнил программу 3, 1, 2, 3, 2, 1, 3 семь раз. Какой результат будет получен для числа 67.

## Лабораторная работа №8. Коллекции и абстрактные классы

Спроектировать структуру классов согласно задаче, базовый класс должен быть абстрактным и содержать абстрактные методы. Выполнить дополнительное задание.

### Задача №1

Создать иерархию классов для Автомобилей. Все классы в иерархии имеют следующие параметры: модель (название автомобиля), вес, скорость. Все классы должны иметь следующие методы:

1. переопределение атрибутов;
2. вычисление стоимости аренды автомобиля в час (для каждого конкретного автомобиля свой метод);
3. вычисление затрат топлива в час (для каждого конкретного автомобиля свой метод);
4. вычисление аренды с учётом затрат топлива, при известном времени аренды (для каждого конкретного автомобиля свой метод).

Создать производный класс Грузовик, характеризуемый также грузоподъемностью (в тоннах) и весом перевозимого груза.

Стоимость аренды Грузовика зависит от грузоподъёмности (10 рублей за каждый килограмм в час). Затраты топлива зависят от текущего груза (10 литров за каждые 100 килограмм в час)

Создать производный класс Автобус, характеризуемый также вместимостью (количество сидячих мест) и коэффициентом комфортабельности.

Стоимость аренды Автобуса зависит от его вместимости (15 рублей за каждое место) и умножается на коэффициент комфортабельности.

Затраты топлива зависят от времени (10 литров на каждые 50 километров).

Создать класс Автопарк, который будет содержать в себе объекты вышеописанных классов. Класс автопарк должен содержать методы:

1. Поиск грузовиков с минимальной стоимостью аренды в час;
2. Поиск самого быстрого Автобуса.

**Дополнительная задача**

Добавить в класс Автопарк метод, который позволит найти и вывести данные об объекте по его модели.

### Задача №2

В зоомагазине продаются животные и аксессуары для содержания животных.

Ниже приводится описание товаров зоомагазина

У всех товаров есть общие атрибуты: название, цена

**Животные**, общий атрибут - пол:

Рыбки – Тип (морская, речная); Цвет

Грызуны – Порода (крыса, хомяк, белка);

Рептилии – Вид (черепаха, змея, ящерица);

**Аксессуары** – общие атрибуты: высота; ширина; длина

Клетки – наличие кормушки.

Аквариумы – наличие компрессора;

Нужно реализовать корзину товаров, в которую можно добавлять товары и выводить чек вида:

1. *Аквариум …….. 1000*
2. *Барбус …………. 100*

*Итого: 1100*

**Дополнительная задача**

Реализовать вывод полной информации о товаре (включая все его специфические параметры)

### Задача №3

В компьютерной стратегии войска делятся на два вида: живая сила (пехота, инженеры) и техника (танки, самолёты).

У всех войск есть общие атрибуты: сила атаки, название, вес

**Живая сила** имеет специфические атрибуты – уровень здоровья, звание:

Пехота имеет дополнительный параметр – выносливость;

Инженеры имеют дополнительный параметр – уровень мастерства (целое число).

**Техника** имеет специфические атрибуты – толщина брони и скорость передвижения:

Танки имеют дополнительный атрибут – наличие противоракетных модулей;

Самолёты имеют дополнительный атрибут - наличие противотанковых ракет.

Нужно реализовать класс – транспортник, который будет вмещать в себя войска (живая сила и техника) по весу и выводить на экран текущий состав (список).

**Дополнительная задача**

Реализовать возможность помещения нескольких объектов в транспортник сразу. Например, поместить сразу 3 танка.

### Задача №4

Учёт оргтехники

Разработать программу учёта оргтехники

У оргтехники есть общие параметры: цена, дата приёма на учёт, ответственное лицо, модель

Типы оргтехники: персональные компьютеры (центральный процессор, RAM, HDD); Принтеры (тип - цветной или ч/б, струйный или лазерный); проекторы (тип - обычный/короткофокусный, разрешение)

Реализовать программный класс, который будет собирать информацию об оргтехнике и выдавать сводную информацию:

Список оргтехники с атрибутами

**Дополнительная задача**

Реализовать метод вывода списка ответственных лиц с общей суммы оргтехники записанной на них (при выполнении данного задания считать, что есть только три отв. Лица: Ivanov, Petrov, Sidorov)

### Задача №5

Разработать классы для спортсменов ДЮСШ. У всех спортсменов есть общие параметры: ФИО, возраст, пол, лучший результат (место на соревнованиях).

В ДЮСШ занимаются следующие спортсмены:

Борцы, у борцов есть атрибуты – количество проведённых схваток, количество побед;

Легкоатлеты, у легкоатлетов нужно хранить - лучший результат на 100 м, 400 м, 1 км;

Пауэрлифтеры , специфические атрибуты – максимальный вес лёжа, количество отжиманий, количество подтягиваний

Реализовать класс, который будет собирать в себе объекты всех трёх типов, и выводить информацию о спортсменах ДЮСШ.

**Дополнительная задача**

Реализовать метод, который выведет лучшего борца, пауэрлифтера и легкоатлета

### Задача №6

Разработать систему классов для хранения информации о командах по разным видам спорта. Для всех видов команд есть общие атрибуты: Тренер, название, количество побед, ничей, поражений.

Для каждого вида команд есть специфические атрибуты:

Футбольная команда – количество забитых голов, количество пропущенных голов;

Хоккейная команда – количество удалений, количество забитых шайб, количество пропущенных;

Волейбольная команда – Количество удачных подач, количество удачных приёмов.

Реализовать класс, который будет собирать в себе объекты всех трёх типов, и выводить информацию о командах.

**Дополнительная задача**

Реализовать метод, который найдёт лучшую команду каждого типа по количеству очков (победа – 3 очка, ничья – 1 очко, поражение – 0 очков).

## Лабораторная работа №9. Параметризация

**Постановка задачи №1. Программа для хранения записей о научной литературе.**

В качестве научной литературы могут быть книги и статьи. Для каждого вида научной литературы реализовать свой класс, классы должны наследоваться от общего суперкласса. Определить уникальные атрибуты для классов, при этом классы должны содержать общие атрибуты: количество страниц, ФИО автора.

Разработать параметризированный класс ScienceList для хранения записей типа научная литература. В классе должен быть атрибут – коллекция литературы. Также должен быть метод добавления элементов в коллекцию.

**Задача 1.** Создайте метод, который вернёт параметризованную коллекцию со списком литературы по автору.

**Задача 2.** Добавьте параметр – научная отрасль для каждого представителя литературы. Создайте метод, который вернёт параметризованную коллекцию по теме.

**Задача 3.** Добавьте метод, который вернёт параметризованную коллекцию со статистикой по авторам (ФИО и количество страниц по каждому автору).

**Задача 4.** Создайте дополнительный класс – пара, один атрибут класса будет объектом научной литературы, второй – число (статус: в наличии, на руках). Измените класс ScienceList: добавьте метод, который выведет всю литературу, которая на руках.

**Постановка задачи №2. Программа для хранения информации в ветеринарной клинике.**

В ветеринарной клинике должны храниться сведения о кошках и собаках. Для каждого вида реализовать свой класс, классы должны наследоваться от общего суперкласса. Определить уникальные атрибуты для классов, при этом классы должны содержать общие атрибуты: ФИО хозяина, возраст.

Разработать параметризированный класс PetList для хранения записей о животных. В классе должен быть атрибут – коллекция животных. Также должен быть метод добавления элементов в коллекцию.

**Задача 1.** Создайте метод, который принимает на вход натуральное число. Метод должен вернуть параметризованную коллекцию со списком животных одного возраста.

**Задача 2.** Добавьте суммарная стоимость лечения для каждого вида животных. Создайте метод, который вернёт параметризованную коллекцию с пятью животными с наибольшей суммой, потраченной на лечение.

**Задача 3.** Добавьте метод, который вернёт параметризованную коллекцию со статистикой по владельцам (ФИО и количество животных).

**Задача 4.** Добавьте суммарная стоимость лечения для каждого вида животных. Создайте метод, который вернёт параметризованную коллекцию со статистикой по суммам, затраченным владельцами (ФИО и общая затраченная сумма).

## Лабораторная работа №10. Исключительные ситуации

Разработать класс согласно задаче, из лабораторной работы №2. Доработать задачу согласно условию. Добавить метод, который будет выбрасывать исключение (см. условие задачи). Реализовать приложение, в котором будет создаваться объект класса, и вызываться метод. В случае выброса исключения произвести обработку и вывести на экран информацию о том, что было выброшено исключение, а также предложить варианты решения.

### Задача №1

Создайте класс погодная станция. Добавьте метод, который будет вычислять величину «ИТХ». Эта величина вычисляется по формуле: давление/ (температура - влажность). Будем считать ситуацию, когда в знаменателе 0 – исключительной.

### Задача №2

Создайте класс Товар. Добавьте атрибут «скидка на товар». Добавьте метод вычисления итоговой прибыли магазина. Для вычисления итоговой прибыли магазина нужно вычесть из стоимости товара скидку, налог (13%) и накладные расходы магазина (10%). Исключительной считается ситуация, когда итоговая прибыль меньше 5%. В данной задаче под процентом понимается процент от начальной стоимости товара.

### Задача №3

Создайте класс Робот. Класс должен содержать атрибуты: координаты местоположения в двумерном пространстве (целочисленные координаты на плоскости). Класс должен содержать метод передвижение, аргументы этого метода — направление, которое задаётся числовым кодом (1 - влево, 2 - вправо, 3 - вниз, 4 — вверх) и количество шагов. Класс должен содержать метод, который выводит на экран текущие координаты робота.

Добавьте атрибут: целое число – радиус действия робота. Будем считать, что область действия робота круг с центром в начале координат и радиусом. Исключительной ситуацией считается выполнение метода передвижения, при котором робот выйдет за пределы области действия.

### Задача №4

Создайте класс Прямоугольник. Класс должен содержать атрибуты: координаты левого верхнего и правого нижнего угла в декартовой системе координат. Класс должен содержать методы вычисления площади и периметра прямоугольника.

Создайте класс «корзину», который будет содержать прямоугольники в коллекции (List), класс должен содержать метод добавления прямоугольника в коллекцию. Исключительной будем считать ситуацию, когда при добавлении новый прямоугольник пересекается с одним из прямоугольников в коллекции.

### Задача №5

Создайте класс Треугольник. Класс должен содержать атрибуты: длины сторон. Класс должен содержать методы вычисления площади и периметра треугольника.

Создайте класс «корзину», который будет содержать треугольники в коллекции (List), класс должен содержать метод добавления треугольника в коллекцию. Исключительной будем считать ситуацию, когда при добавлении нового треугольника в коллекцию его площадь будет больше чем суммарная площадь всех треугольников в коллекции (правило работает, когда в коллекции уже есть больше трёх треугольников).

### Задача №6

Создайте класс Круг. Класс должен содержать атрибут: радиус круга. Класс должен содержать методы вычисления длины окружности, которая является границей круга и вычисления площади круга.

Добавьте два атрибута – координаты центра круга (на плоскости).

Создайте класс «корзину», который будет содержать круги в коллекции (List), класс должен содержать метод добавления круга в коллекцию. Исключительной будем считать ситуацию, когда при добавлении центр нового круга находится в одном из кругов из коллекции.

### Задача №7

Создайте класс Вектор. Класс должен содержать атрибуты: координаты вектора в декартовой системе координат. Класс должен содержать метод вычисления длины вектора.

Создайте класс «корзину», который будет содержать векторы в коллекции (List), класс должен содержать метод добавления вектора в коллекцию. Исключительной будем считать ситуацию, когда при добавлении новый вектор коллинеарен хотя бы одному вектору из коллекции.

### Задача №8

Создайте класс Телевизор. Класс должен содержать атрибуты: текущее состояние (включен/выключен), текущий канал. Класс должен содержать метод включения, выключения (изменение атрибута текущее состояние), также класс должен содержать метод изменения текущего канала и метод выводящий информацию о текущем состоянии телевизора (если телевизор выключен вывести просто выключен, если включен то вывести текущий канал).

Добавьте атрибут – массив каналов. Этот атрибут должен содержать массив целых чисел с номерами каналов, для общности будем считать, что у всех каналов есть уникальный числовой пятизначный номер. Метод переключения должен теперь принимать этот номер для переключения. Исключительной считается ситуация попытки переключения на несуществующий номер.

### Задача №9

Создайте класс Радиоприёмник. Класс должен содержать атрибуты: текущее состояние (включен/выключен), текущий канал. Класс должен содержать метод включения, выключения (изменение атрибута текущее состояние), также класс должен содержать метод изменения текущего канала и метод выводящий информацию о текущем состоянии приёмника (если приёмник выключен вывести просто выключен, если включен то вывести текущий канал).

Добавьте атрибут – массив частот. Этот атрибут должен содержать массив строк с частотами каналов (“101 FM” и тд). Метод переключения должен теперь принимать строку (частоту) для переключения. Исключительной считается ситуация попытки переключения на несуществующую частоту.

### Задача №10

Создайте класс Ячейка камеры хранения. Класс должен содержать атрибуты: текущее состояние (свободна/занята), предмет в ячейке (строка). Класс должен содержать метод, который позволит положить в ячейку предмет и метод выводящий информацию о текущем состоянии ячейки (если ячейка пуста вывести empty, иначе вывести название предмета в ячейке).

Создайте класс камера хранения. Класс должен содержать коллекцию (List) ячеек. Число ячеек заранее определено и не должно меняться. Класс должен содержать метод, который будет принимать на вход порядковый номер ячейки и объект для помещения в ячейку. Исключительной считается ситуация при которой осуществляется попытка помещения в несуществующую ячейку.

### Задача №11

Создайте класс Треугольник. Класс должен содержать атрибуты: высота и основание. Класс должен содержать методы вычисления площади и периметра треугольника.

Создайте класс «корзину», который будет содержать треугольники в коллекции (List), класс должен содержать метод добавления треугольника в коллекцию. Исключительной будем считать ситуацию, когда при добавлении нового треугольника в коллекцию его периметр будет больше чем суммарный периметр всех треугольников в коллекции (правило работает, когда в коллекции уже есть больше трёх треугольников).

### Задача №12

Создайте класс равнобедренный Треугольник. Класс должен содержать атрибуты: высота и боковая сторона. Класс должен содержать методы вычисления площади и периметра треугольника.

Создайте класс «корзину», который будет содержать треугольники в коллекции (List), класс должен содержать метод добавления треугольника в коллекцию. Исключительной будем считать ситуацию, когда при добавлении нового треугольника в коллекцию его площадь будет больше чем удвоенная средняя площадь всех треугольников в коллекции (правило работает, когда в коллекции уже есть больше трёх треугольников).

### Задача №13

Создайте класс Кинофильм. Класс должен содержать атрибуты: категория (три категории А, В, С), цена билета. Класс должен содержать метод вычисления цены билета от количества дней со дня премьеры. Цена на билеты фильмов категории С снижается на 30% через неделю после премьеры, В на 20% через 10 дней после премьеры, А на 15% через 2 неделе после премьеры.

Создайте класс Анализатор. Класс содержит метод который получает на вход объект класса Кинофильм и количество проданных на него билетов. Метод вычисляет общую прибыль с билетов и суммирует в атрибут класса. Кроме того, метод «запоминает» общее количество проданных билетов, а также минимальную цену.

Добавьте метод анализа, этот метод вычисляет коэффициент прибыли по следующей формуле:

Берётся 50% от общей прибыли, это число делится на количество проданных билетов. Исключительной считается ситуация когда полученное число меньше чем минимальная цена билета.

### Задача №14

Создайте класс Дробь. Класс должен содержать атрибуты: числитель и знаменатель. Класс должен содержать метод сокращения дроби и/или метод добавления к дроби другой дроби.

Создайте класс вычислитель. Этот класс должен содержать атрибут «начальное значение», которое является представителем класса Дробь с числителем и знаменателем равными 1.

Класс должен содержать метод, который принимает на вход объект класса дробь, и изменяет «начальное значение» применив к нему арифметическую операцию (операция произвольная – воспользуйтесь методом генерации произвольного числа от 1 до 4). Исключительной является ситуация, когда после выполнения такой операции «начальное значение» станет равным 0.

### Задача №15

Создайте класс Персональный компьютер. Класс должен содержать текстовое описание и цену составляющих компьютера: Системный блок, Монитор, Клавиатура, Мышь. Класс должен содержать методы вывода общей цены и общего описания (через пробел описание всех составляющих).

Создайте класс «корзину», который будет содержать Компьютеры в коллекции (List), класс должен содержать метод добавления компьютера в коллекцию. Исключительной будем считать ситуацию, когда при добавлении нового компьютера в коллекцию его цена будет больше чем утроенная средняя цена всех компьютеров в коллекции (правило работает, когда в коллекции уже есть больше трёх компьютеров).

### Задача №16

Создайте класс Комплексное число. Класс должен содержать атрибуты: действительная и мнимая часть. Класс должен содержать метод вычисления модуля комплексного числа, метод умножения на скаляр и метод вывода на экран (пример: *4-2i*).

Создайте класс вычислитель. Этот класс должен содержать атрибут «начальное значение», которое является представителем класса Комплексное число с действительной и мнимой частью равными 1.

Класс должен содержать метод, который принимает на вход объект класса действительное число, и изменяет «начальное значение» применив к нему арифметическую операцию (операция произвольная – воспользуйтесь методом генерации произвольного числа от 1 до 4). Исключительной является ситуация, когда после выполнения такой операции «начальное значение» станет равным -*i*.

### Задача №17

Создайте класс Будильник. Класс должен содержать атрибуты: значение минут, значение часов, а также время, на которое поставлен будильник. Класс должен содержать методы: проверки, сколько осталось до срабатывания будильника, метод подачи звукового сигнала (вывести в консоль ring).

Добавьте три атрибута: текущий день, месяц и год. При установке будильника также нужно указать дату, на которую ставится будильник.

Исключительной будем считать ситуацию, когда будильник устанавливается на уже прошедшую дату.

### Задача №18

Создайте класс Треугольник. Класс должен содержать атрибуты: значение двух углов треугольника и стороны. Класс должен содержать метод, который определяет тип треугольника (прямоугольный, остроугольный, тупоугольный, равносторонний, равнобедренный, не существует), а также метод, который определяет его площадь.

Создайте класс «корзину», который будет содержать треугольники в коллекции (List), класс должен содержать метод добавления треугольника в коллекцию. В коллекцию должны добавляться только существующие треугольники

Исключительной будем считать ситуацию, когда при добавлении нового треугольника в коллекцию его площадь будет больше чем удвоенная средняя площадь всех равнобедренных в коллекции (правило работает, когда в коллекции уже есть больше трёх равнобедренных треугольников).

## Формирование варианта в лабораторных работах 11-13

Задача состоит из основной и дополнительной задачи. При выполнении задачи требуется реализовать классы согласно заданию, выполнение дополнительной задачи может потребовать изменения классов. Итоговые баллы начисляются за основную и дополнительную задачу.

|  |  |  |
| --- | --- | --- |
| **Вариант** | **Задача** | **Дополнительная задача** |
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 1 | 3 |
| 4 | 1 | 4 |
| 5 | 1 | 5 |
| 6 | 2 | 1 |
| 7 | 2 | 2 |
| 8 | 2 | 3 |
| 9 | 2 | 4 |
| 10 | 2 | 5 |
| 11 | 1 | 6 |
| 12 | 2 | 6 |
| 13 | 1 | 7 |
| 14 | 1 | 8 |
| 15 | 2 | 7 |
| 16 | 2 | 8 |

## Лабораторная работа №11. Паттерн проектирования Стратегия

### Задача №1. RPG игра

В игре планируются следующие игровые классы: Рыцарь (Knight), Вор (Rogue), Маг (Mage). Все персонажи обладают характеристиками «количество здоровья», «мана». Каждый игровой класс обладает первичным атрибутом: ловкость (вор), интеллект (маг), сила (рыцарь).

Персонаж каждого класса может быть оснащён оружием: посох (staff), меч (sword), кинжал (dagger). Персонажи могут использовать оружие, величина урона оружия увеличивается, в зависимости от атрибутов (урон меча увеличивается от силы, посоха от интеллекта, кинжала от ловкости). По умолчанию урон оружия увеличивается в два раза от первичного атрибута оружия.

Задача: организовать структуру классов таким образом, чтобы она поддерживала следующие свойства:

1. Каждому конкретному персонажу можно дать любое оружие;
2. Существует возможность динамической замены оружия во время игры (выполнения программы);
3. Добавление новых видов оружия не влечёт за собой изменений связанных с классами персонажей.
4. Реализовать метод info(), который будет отображать информацию о текущем состоянии персонажа.
5. Реализовать метод attack(), который проводит атаку имеющимся оружием и возвращает значение – нанесённый урон.

**Дополнительная задача №1**

добавить игровой класс: Друид (Druid), специальный показатель друида – дух (увеличивает урон для оружия, которое зависит от ловкости и/или интеллекта). Друид может принимать различные формы, которые увеличивают его атрибуты: форма медведя (увеличение силы), форма кошки (увеличение ловкости), форма совуха (увеличение интеллекта). Переключение форм тратит ману.

**Дополнительная задача №2**

добавить новый вид оружия: Заражённый меч. Заражённый меч увеличивает урон в зависимости от текущего объёма здоровья, при этом каждая атака должна отнимать 5% от текущего здоровья персонажа.

Дополнительная задача №3:

Добавить второй слот для оружия и разделить оружие на два типа: двуручное и одноручное. При этом должны выполняться следующие правила:

1. Если надеть два одноручных оружия, то оружие во второй руке наносит урон в полтора раза меньше от своего значения
2. Нельзя надеть два двуручных оружия

**Дополнительная задача №4**

Добавить игровой класс: варлок. Варлоки умеют одевать только посохи. Урон Варлока равен пятикратному урону, посоха. При каждой атаке посохом варлок расходует 10% маны. Варлок не может атаковать, если у него нет маны. Варлоки могут восстанавливать ману за счёт своего здоровья – нужно реализовать метод lifeTap, который переводит 10% здоровья в ману.

**Дополнительная задача №5**

Изменить метода attack() таким образом, чтобы он принимал на вход объект персонажа по которому проводится атака. Персонаж, которого атакуют должен получать урон от атаки (уменьшение здоровья), при этом если здоровье становиться равным нулю персонаж «умирает».

**Дополнительная задача №6**

Добавить новый вид оружия: меч тысячи истин. Перед каждой атакой меч увеличивает силу персонажа на 10 единиц, но при этом понижает интеллект персонажа на 5 (Интеллект не может быть снижен ниже нуля). Урон меча рассчитывается по формуле: 15 х Сила.

**Дополнительная задача №7**

Добавить игровой класс: Шаман. Шаманы имеют два первичных атрибута – ловкость и интеллект. Шаман может установить один из тотемов (увеличение силы, ловкости или интеллекта), установка тотема тратит 5 единиц маны.

**Дополнительная задача №8**

Добавить новый вид оружия: Молот Громовая ярость. Урон молота рассчитывается по формуле: 15 х (Интеллект + Ловкость + Сила). Каждая атака добавляет молоту одну единицу ярости, если число единиц ярости равно пяти, то атака молота усиливается в 5 раз, при этом счётчик ярости обнуляется.

### Задача №2. RPG игра 2

В ролевой игре персонажи обладают характеристиками: количество здоровья, мана и выносливость. Кроме того, у персонажей есть атрибуты: удача, магическая сила, мастерство. Все персонажи могут одевать различные виды оружия, и производить атаку этими видами оружия.

Лук – урон рассчитывается по формуле 10 x Мастерство, каждая атака луком тратит 10 единиц выносливости. Если выносливости недостаточно, урон уменьшается в 4 раза.

Меч - урон рассчитывается по формуле 5 x (Мастерство+3 x Удача), каждая атака тратит 8 единиц выносливости. Если выносливости недостаточно, атака не производится.

Магический посох – урон рассчитывается по формуле 8 х Магическая сила, каждая атака тратит 7 единиц маны. Если маны недостаточно, атака не производится, при этом посох восстанавливает персонажу 10 единиц маны.

Все персонажи могут производить одну из специальных атак, при этом свойства атаки зависят от экипированного оружия:

- Огненная атака, расходует ману (15 единиц на атаку). Урон рассчитывается по формуле: 15 х Магическая сила. Если маны недостаточно, то атака не проводится. Если персонаж экипирован магическим посохом, то количество используемой маны сокращается на 5 единиц.

- Физическая атака, расходует здоровье персонажа (20 единиц на атаку). Урон рассчитывается по формуле: 10 х (Мастерство + 3 х Удача). Если величина здоровья не позволяет провести атаку, то атака не проводится. Если персонаж экипирован мечом, то урон увеличивается в полтора раза, а расходуемое здоровье также увеличивается на 5 единиц.

Задача: организовать структуру классов таким образом, чтобы она поддерживала следующие свойства:

1. Каждому конкретному персонажу можно дать любое оружие;
2. Существует возможность динамической замены оружия во время игры (выполнения программы);
3. Реализовать метод info(), который будет отображать информацию о текущем состоянии персонажа и экипированном оружии.
4. Реализовать метод attack(), который проводит атаку имеющимся оружием и возвращает значение – нанесённый урон.
5. Добавление нового вида оружия/специальной атаки не должно влиять на существующие классы
6. Реализовать метод specialAttack(), который проводит специальную атаку и возвращает значение – нанесённый урон.

**Дополнительная задача №1**

Изменить метода attack() таким образом, чтобы он принимал на вход объект персонажа по которому проводится атака. Реализовать специальную атаку: Сжигание маны. Атака расходует здоровье персонажа (15 единиц), при этом уменьшает количество маны цели на 20 единиц маны. Если здоровья недостаточно, атака не проводится. Если у цели недостаточно маны, то за каждую недостающую единицу атака наносит урон кратный 10.

**Дополнительная задача №2**

Добавить атрибут для персонажа – максимальное количество маны. Реализовать новый вид оружия – копьё. Урон копья рассчитывается по формуле: 7 х (4 х Мастерство + Магическая сила). Каждая атака копьём восстанавливает персонажу 10 единиц маны. Атака не должна увеличивать количество маны больше максимально возможного количества.

**Дополнительная задача №3**

Добавить атрибут для персонажа – максимальное количество здоровья. Реализовать новый вид оружия – кинжал. Урон кинжал рассчитывается по формуле: 3 х (4 х Мастерство + 2 х Удача). Каждая атака восстанавливает персонажу 20 единиц здоровья. Атака не должна увеличивать количество здоровья больше максимально возможного количества.

**Дополнительная задача №4**

Реализовать специальную атаку: Усиление. Атака не наносит урона, но увеличивает все атрибуты персонажа на 10%.

**Дополнительная задача №5**

Изменить метод attack() таким образом, чтобы он принимал на вход объект персонажа по которому проводится атака. Разделить весь урон (оружия и специальных атак) на физический и магический. Для персонажей реализовать возможность добавления слабости к определённому виду урона. При получении такого урона, персонаж должен получать урон в пятикратном размере.

**Дополнительная задача №6**

Изменить метод attack() таким образом, чтобы он принимал на вход объект персонажа по которому проводится атака. Реализовать специальную атаку: Ледяная атака. Атака расходует ману персонажа (15 единиц) и наносит 10 х (3 х Удача + 4 х Магическая сила) урона. Если здоровье цели после атаки становится меньше 50%, то атака дополнительно наносит 20 единиц урона и восполняет 15 единиц маны.

**Дополнительная задача №7**

Реализовать специальную атаку: Атака ветром. Атака ветром расходует ману (20 единиц) персонажа и наносит урон 12 х (Магический урон + 4 х Удача). Каждая атака добавляет 1 единицу силы ветра. Если число единиц силы ветра равно 3, то атака ветром восполняет персонажу 35 единиц маны и обнуляет счётчик силы ветра.

**Дополнительная задача №8**

Реализовать новый вид оружия – жезл арканы. Жезл должен иметь два режима атаки. В первом режиме жезл расходует ману, каждая атака добавляет один уровень арканы (максимум 4). Урон считается по формуле: 4 х (Магическая сила + 10 х Уровень арканы), расход маны считается по формуле 8 х Уровень арканы.

Второй режим не расходует ману, наносимый урон вычисляется по формуле 15 х Уровень арканы + 10. Атаки во втором режиме обнуляют счётчик арканы.

## Лабораторная работа №12. Наблюдатель

### Задача №1

Вы работаете в Гринвичской обсерватории (Royal Greenwich Observatory). В ваши обязанности входит сбор данных о всех космических объектах, летящих в сторону Земли. Каждый объект имеет массу (Mass), скорость (Speed), площадь (Square).

Пример данных, которые можно получить от обсерватории о трех космическим х объектах:

1) (40, 19, 40)

2) (60, 20, 90)

3) (466, 334, 325)

Клуб астрономов хочет получать информацию о массе и скорости объектов. Агентство NASA хочет получать полную информацию об объектах.

Вашей задачей является: Разработать структуру классов, которая позволит получать данные и производить оповещение наблюдателей. Разработанное решение должно удовлетворять следующим критериям:

1. Позволяет добавлять новых наблюдателей без изменения кода;
2. Позволяет удалять наблюдателей;
3. Считывание данных нужно производить из текстового файла с задержкой в 5 секунд;
4. Для тестирования информация о получении наблюдателями оповещений должна выводится в консоль (если в задании не сказано другого). Например, при получении информации должно выводится сообщение вида «NASA получило информацию: 60, 20, 90».

**Дополнительная задача №1**

Создать получателя данных, который будет производить классификацию объектов на 4 типа и в зависимости от типа выводить на экран 4 разных изображения.

Типы объектов:

1. Быстрые объекты – скорость больше 40
2. Крупные объекты – площадь больше 100
3. Тяжёлые объекты – масса больше 200
4. Другие объекты считаются нормальными

**Дополнительная задача №2**

Информацию об объектах хочет получать научно исследовательская группа, которая разработала следующую формулу:

Mass\*10+Speed\*5-Square\*12.

Если результат вычислений больше нуля, то объект признаётся аномальным, иначе обычным. Научно исследовательская группа планирует публиковать информацию о всех объектах (вывод в консоль), помимо характеристик они хотя публиковать класс объектов (general – обычный, anomaly - аномалия).

Кроме того, данными группы интересуются две организации: XCom и MIB, в дальнейшем стоит ожидать, что интерес будет расти.

MIB отслеживает все аномальные объекты (выводит информацию на экран).

XCom отслеживает аномальные объекты скорость которых превышает 20 единиц, если такой объект обнаружен то, должна срабатывать тревога (вывод в консоль).

**Дополнительная задача №3**

Центр статистики и исследований хочет получать информацию и фиксировать её в базе данных. Для тестирования реализуйте запись в текстовый файл данных в формате:

Mass, Speed, Square, <Системная дата и время>

**Дополнительная задача №4**

Книга рекордов Гиннеса решила составить список уникальных объектов, зафиксированных обсерваторией. Кроме прочего, они решили сделать топ объектов (максимум 10) по убыванию в следующем порядке: масса, скорость, площадь.

**Дополнительная задача №5**

Религиозная организация Свидетели Солнца верит, что как только общая масса объектов (о которых они знают) превысит 300 произойдёт конец света. В связи с этим они считают, что если такое произойдёт, то получать информацию в дальнейшем смысла нет. Реализовать такую возможность (при выполнении условия нужно удалять из списка рассылки).

**Дополнительная задача №6**

Обсерватория фиксирует объекты на расстоянии 300 тыс км от земли. Научная группа из МГУ хочет получать эти данные для рассчёта времени за которое объект достигнет земли. Эту информацию они планируют публиковать в формате:

Mass, Speed, Square, <Время в формате>

Время публикуется в формате <Число дней>, <Часы>, <Минуты> (секунды округляются)

**Дополнительная задача №7**

Специальное подразделение руководства полётами, хочет получать данные обсерватории и вычислять индекс опасности объекта. Индекс опасности вычисляется по следующей формуле:

Speed\*Mass/Square

Если полученное значение будет больше 5 то, объект признаётся опасным. Информацию об опасных объектах планируется публиковать в формате:

<Индекс>, Mass, Speed, Square

**Дополнительная задача №8**

Клуб юных астрономов также решил получать данные обсерватории с целью составления списка уникальных характеристик объектов (то есть, две разных кометы с характеристиками 10,12,13 и 10,12,13 считаются одинаковыми). Разработайте для них решение, которое позволит получать информацию и формировать список (нужно реализовать метод) уникальных объектов.

### Задача №2

Информационное агентство "Ulan-Ude news" аккумулирует различную информацию после чего публикует её. Информация агентства имеет следующую структуру:

Сообщение, дата в формате YYYY-mm-dd (например, 2020-09-30), тип сообщения (целое число).

Типы сообщения: 1 - информация о погоде, 2 - курс валют, 3 - новости города.

Газета информ полис и телеканал Ариг Ус хотят получать свежую информацию.

Вашей задачей является: Разработать структуру классов, которая позволит получать данные и производить оповещение наблюдателей. Разработанное решение должно удовлетворять следующим критериям:

* Позволяет добавлять новых наблюдателей без изменения кода;
* Позволяет удалять наблюдателей;
* Считывание данных нужно производить из текстового файла с задержкой в 5 секунд;
* Для тестирования информация о получении наблюдателями оповещений должна выводится в консоль (если в задании не сказано другого). Например, при получении информации должно выводится сообщение вида «Информ полис получил информацию: <Текст сообщения>, <Дата>, <Тип сообщения>».

Пример сообщений:

*-15 1 20; 2020-11-14; 1*

*dollar 74; 2020-11-14; 2*

*euro 90; 2020-11-14; 2*

*Курс доллара опять растёт он стал 74;2020-11-14;3*

Первое сообщение означает, что температура -15, вид осадков – 1, скорость ветра 20. Дата – 2020-11-14 и само сообщение является информацией о погоде (тип 1)

Второе сообщение означает, что курс доллара 74, Дата – 2020-11-14 и само сообщение является курсов валют (тип 2)

Третье сообщение означает, что курс евро 90, Дата – 2020-11-14 и само сообщение является курсов валют (тип 2)

Четвёртое сообщение: «*Курс доллара опять растёт он стал 74*», Дата – 2020-11-14 и само сообщение новостью города (тип 3)

**Дополнительная задача №1**

Приложение "Погода в Улан-Удэ" интересуют только новости погоды. Поэтому вы задали чёткий формат сообщений о погоде - в этих сообщениях передаётся три числа: температура, вид осадков, скорость ветра (например, можно считать что формат сообщения выглядит следующим образом: <температура>, <вид осадков>, <скорость ветра>).

Виды осадков: 1 - ясное небо, 2 - дождь, 3 - снег.

Приложение должно получать информацию и выводить на экран изображение, которое демонстрирует изображение с соответсвующими осадками.

**Дополнительная задача №2**

Сайт НовостиБурятии.рф интересуют все сообщения, при получении сообщения этот сайт каким-то образом формирует рейтинг сообщения (для тестирования назначайте произвольное число) и выставляет его в виде новости.

Сайт НовостиРоссии интересуют новости рейтинг которых выше 10. Сайт НовостиМира интересуют новости рейтинг которых выше 20. Разработайте решение, которое позволит реализовать подобную ситуацию.

**Дополнительная задача №3**

Центр статистики и исследований хочет получать сообщения и фиксировать её в базе данных. Для тестирования реализуйте запись в текстовый файл данных в формате:

<Текст сообщения>, <Дата>, <Тип сообщения>, <Системная дата и время>

**Дополнительная задача №4**

Агрегатор новостей компании Гугл вычисляет информационную мощность сообщения следующим образом:

<Коэффициент типа>\*<Количество символов в сообщении>/100

Где коэффициент типа равен

0.83- информация о погоде

0.57 - курс валют

0.95 - новости города.

Кроме прочего, в любой момент времени хотят формировать топ 5 новостей (по коэффициенту, если коэффициент равен, то сначала нужно вывести более раннее сообщение).

**Дополнительная задача №5**

Группу финансовых аналитиков интересует только курс валют. Допустим, что в сообщениях передаётся информацию о двух типах валют - евро и доллары.

В группе аналитиков уверены, что если курс доллара превысит 80 рублей или курс евро превысит 100 рублей, то дальнейшее получение сообщений бессмысленно. Реализовать такую возможность (при выполнении условия нужно удалять из списка рассылки).

**Дополнительная задача №6**

Исследовательская группа экономистов фиксирует рост доллара. Допустим, что в сообщениях передаётся информацию о двух типах валют - евро и доллары. Задачей группы является фиксировать всю информацию о курсе и выдавать оповещение всякий раз, когда курс любой из валют изменяется более чем на 10%.

**Дополнительная задача №7**

Комитет городской безопасности оценивает всю информацию в медиапространстве и пытается найти странные сообщения. Для этих целей в комитете хотят получать информацию и планируют анализировать новости города.

Аналитики комитета хотят выделять подозрительные дни. Подозрительным день считается если в течении дня было два или более сообщения в котором встречалось число 13.

В этом случае комитет должен производить запись в файл:

<Дата>, количество встреченных чисел 13.

**Дополнительная задача №8**

Клуб юных метеорологов очень сильно интересуется данными о погоде. Поэтому вы задали чёткий формат сообщений о погоде - в этих сообщениях передаётся три числа: температура, вид осадков, скорость ветра (например, можно считать, что формат сообщения выглядит следующим образом: <температура>, <вид осадков>, <скорость ветра>).

Виды осадков: 1 - ясное небо, 2 - дождь, 3 - снег.

Помимо публикации материалов в Клубе решили собрать информацию сколько дней была ясная погода, сколько дней шёл дождь или снег.

В разрабатываемом классе необходимо реализовать метод, который выведет эту информацию.

## Лабораторная работа №13. Декоратор

### Задача №1

Универсальный химический обогатитель вопреки всем законам физики и химии способен создавать стабильные вещества из молекул различных элементов.

Суть его работы крайне проста, на вход подаётся базовое вещество, которое представляет собой некоторое число молекул (например, H2O – две молекулы водорода и одна молекула кислорода). Далее вещество проходит через различные фильтры и на выходе вы получаете новое вещество.

Вы разрабатываете приложение для обогатителя (на основе паттерна Декоратор). Для описания исходного вещества вы используете не химическую формулу, а строку, которая представляет собой набор латинских символов с описанием элементов. Например: H2O будет записано как «H H O», а C2H5OH как «C C H H H H H O H».

При этом каждое исходное вещество является отдельным классом, например для воды и перекиси водорода нужно будет создавать отдельные классы.

Нужно разработать несколько фильтров:

1) «Кислородный фильтр», который будет добавлять молекулу кислорода, то есть строка «H H O» после этого фильтра примет вид «H H O O».

2) «Водородный фильтр», который будет добавлять молекулу водорода, то есть строка «H H O» после этого фильтра примет вид «H H O H».

Решение должно позволять добавлять новые фильтры, кроме того должно позволять различные виды исходного вещества и комбинации фильтров.

**Дополнительная задача №1**

Разработать специальный фильтр, который «убирает» весь кислород из вещества.

**Дополнительная задача №2**

Разработать справочник (например, массив) с ценами за молекулу каждого вещества. Доработать решение так, чтобы оно позволяло считать итоговую стоимость вещества.

**Дополнительная задача №3**

Доработать решение так, чтобы итоговая формула полученного вещества была приведена к «нормальной форме». То есть, например, «C C H H H H H O H» преобразуется в «С2 H5 O H».

**Дополнительная задача №4**

Разработать специальный фильтр, который преобразует вещество следующим образом: вычисляется среднее число молекул в веществе. После чего преобразуется в вещество, в котором количество каждой из молекул, будет равно округлённому до целого среднему числу.

Например, «C C H H H H H O H» - всего 9 молекул, разных по 3, среднее 3. Поэтому после фильтрации будет «C C C H H H O O O».

**Дополнительная задача №5**

Разработать фильтр «Выделитель воды», то есть выделяет всю воду («H H O») из вещества, после чего возвращает вещество без воды.

**Дополнительная задача №6**

Разработать фильтр «Максимизатор». Этот фильтр находит те молекулы, которых больше всего в веществе и выделяет только эти молекулы.

**Дополнительная задача №7**

Разработать справочник (например, массив) с ценами за молекулу каждого вещества. Разработать фильтр «Минимизатор». Этот фильтр находит самые дорогие молекулы (по итоговой стоимости) и удаляет их из вещества.

**Дополнительная задача №8**

Доработать решение: создать подсистему логирования, которая будет записывать действия всех фильтров в текстовый файл.

### Задача №2

В кондитерской фабрике решили оптимизировать производство и разбили производство тортов на несколько этапов. Для начала изготавливается основа (прямоугольная, круглая). Каждая основа характеризуется высотой и набором параметров (например, прямоугольная имеет ширину и длину). Кроме этого, к торту прилагается строка описания.

Для создания полноценного торта основа оформляется при помощи различных инструментов оформления:

Кремовой добавки – состоит из названия крема и добавочной высоты.

Кремовой добавки для боковых граней – состоит из названия крема и количества, которое наносится (то есть, например, для прямоугольной основы прибавка к ширине и длине).

Украшение – различные фигурки, которые добавляют высоту.

Необходимо разработать программное решение (на основе паттерна Декоратор). Решение должно позволять добавлять виды оформления, кроме того должно позволять создавать различные виды основ и комбинации инструментов оформлений.

**Дополнительная задача №1**

Разработать инструмент оформления «удалить крем», который удалит текущий крем с торта (если он есть).

**Дополнительная задача №2**

Доработать решение, так чтобы оно позволяло считать итоговую стоимость изделия как сумму от стоимости компонентов. При этом учесть возможность скидки (5% если, количество компонентов превышает 5).

**Дополнительная задача №3**

Разработать инструмент «Выравниватель», который позволяет создать квадратное изделие (берёт минимальное из ширины и длины в качестве стороны квадрата).

**Дополнительная задача №4**

Разработать инструмент «Нарезчик», который нарезает изделие на куски и возвращает один кусок (с соответствующими размерами).

**Дополнительная задача №5**

Доработать решение, так чтобы оно позволяло считать итоговый вес изделия. При этом решение должно выводить описание в виде чека (то есть вес каждой компоненты).

**Дополнительная задача №6**

Разработать инструмент «Выравниватель2», который позволяет создать квадратное изделие (берёт максимальное из ширины и длины в качестве стороны квадрата). Данный инструмент должен дописать сколько нужно добавить для выравнивания

**Дополнительная задача №7**

Доработать решение так, чтобы для каждого компонента можно было задать количество калорий, после чего посчитать общее число и вывести список элементов с калориями.

**Дополнительная задача №8**

Доработать решение: создать подсистему логирования, которая будет записывать действия всех инструментов в текстовый файл.

## Формирование варианта в лабораторной работе 14

|  |  |  |
| --- | --- | --- |
| **Вариант** | **Задача** | **Дополнительная задача** |
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 1 | 3 |
| 4 | 1 | 4 |
| 5 | 1 | 5 |
| 6 | 1 | 6 |
| 7 | 2 | 1 |
| 8 | 2 | 2 |
| 9 | 2 | 3 |
| 10 | 2 | 4 |
| 11 | 2 | 5 |
| 12 | 2 | 6 |
| 13 | 3 | 1 |
| 14 | 3 | 2 |
| 15 | 3 | 3 |
| 16 | 3 | 4 |

Задача состоит из основной и дополнительной задачи. При выполнении задачи требуется реализовать классы согласно заданию, выполнение дополнительной задачи может потребовать изменения классов. Итоговые баллы начисляются за основную и дополнительную задачу.

## Лабораторная работа №14. Паттерн проектирования Команда

### Задача №1. Робот

Роботизированная плафрома представляет собой уникальную плату с универсальными интерфейсами подключения различных частей. Изначально можно выделить несколько видов таких частей:

1. Средства передвижения – на данный момент доступны колёса, гусеницы и воздушная подушка.
2. «Органы чувств» - камеры, датчики тепла, эхолоты.
3. Манипуляторы – механические «руки», краны и т.д.

Подобная конструкция популярна и позволяет подключать любое устройство, имеющее интерфейс подключения. В связи с этим многие производители решили создавать устройства для платформы.

Для примера необходимо реализовать части от двух производителей: BugsyTools и RosentalElectronics. Классы для управления устройствами BugsyTools являются наследниками базового класса, а методы для управления конкретными устройствами описываются в виде camelCase на Английском языке (например, для гусениц они реализуют команды moveForward, moveBack). Классы для управления устройствами RoseintalElectronics не имеют общих родителей, и все функции управления обозначаются числовыми кодами (instruction1, instruction2 и т.п.).

Задача: разработать программное решение для пульта управления роботизированной платформой. Решение должно удовлетворять требованиям:

1. Пульт должен поддерживать назначение управляющих инструкций на 8 кнопок;
2. Пульт должен поддерживать возможность добавления различных устройств;
3. Управляемые устройства не обязательно должны быть связаны одним родителем или интерфейсом.
4. Для тестирования: при нажатии любой кнопки необходимо выводить в консоль соответствующую информацию.

**Дополнительная задача №1**

Разработать следующие макрокоманды:

1. Включение робота, включение шасси, включение датчиков.
2. Проверка эхолотом расстояния до ближайшей стены, если расстояние больше 10 см, движение вперёд на 10 сантиметров. Если расстояние меньше 10 см, то движение назад на 5 сантиметров.

**Дополнительная задача №2**

Разработать решение для создания и записи лога робота. Лог должен записываться в текстовый файл и содержать информацию в следующем виде:

Дата, название выполненной операции, результат выполнения операции

**Дополнительная задача №3**

Разработать решение для кнопки отмены последней команды. Под отменой понимается, выполнение действия обратного последней команде (если это возможно). В случае если такое действие не возможно (например, нельзя «обратить» проверку расстояния до стены) вывести на экран соответствующее сообщение. Если действий не было произведено, при нажатии на кнопку написать сообщение.

**Дополнительная задача №4**

Для организации многопоточности реализовать очередь команд. Очередь должна работать по следующей схеме: при попытке выполнить любую операцию она попадает в очередь, как только количество операций в очереди будет равно пяти эти операции выполняются последовательно в порядке добавления каждая с задержкой в 2 секунды.

**Дополнительная задача №5**

Доработать решение – создать объект, который будет отслеживать состояние робота, в зависимости от применённых операций (например, после передвижения координаты робота в пространстве изменяются). Данное решение (программный код) должно быть не изменяемым и не зависеть от добавления новых видов устройств.

**Дополнительная задача №6**

Доработать решение – создать «виртуальный пульт», данный пульт не должен иметь физического ограничения на количество кнопок, то есть количество кнопок должно быть расширяемым в ходе выполнения программы. При этом каждой команде пульта ставиться в соответствие уникальная текстовая метка (MoveForward, UseHand). Управление роботом должно осуществляться через командную строку вводом соответствующих команд.

### Задача №2. Умная теплица

«Умные» устройства становятся всё доступнее. Поэтому мы решили разрабатывать наборы для дачников. Одним из таких наборов является «Умная» теплица. Мы планируем ставить в теплицу следующие виды устройств:

1. Датчики влажности воздуха и почвы.
2. Системы полива.
3. Датчик температуры.
4. Управление вентиляцией.

Для примера нужно реализовать управляющие классы для устройств. Датчики должны быть разработана компанией GreenCorporations и все они реализуют интерфейс *Estimator*. Системы полива разрабатываются компанией SantoroLimited, данная компания выпускает как системы обычного полива, так и капельного. Контора NickyAndSons занимается системами вентиляции.

Задача: разработать программное решение для пульта управления умной теплицей. Решение должно удовлетворять требованиям:

1. Пульт должен поддерживать назначение управляющих инструкций на 6 кнопок;
2. Пульт должен поддерживать возможность добавления различных устройств;
3. Управляемые устройства не обязательно должны быть связаны одним родителем или интерфейсом.
4. Для тестирования: при нажатии любой кнопки необходимо выводить в консоль соответствующую информацию.

**Дополнительная задача №1**

Разработать следующие макрокоманды:

1. Проверить влажность воздуха, проверить влажность почвы, проверить температуру в теплице.
2. Проверить влажность почвы, если она меньше 30% включить систему полива.

**Дополнительная задача №2**

Разработать решение для создания и записи лога выполненных команд. Лог должен записываться в текстовый файл и содержать информацию в следующем виде:

Дата, название выполненной операции, результат выполнения операции

**Дополнительная задача №3**

Разработать решение для кнопки отмены последней команды. Под отменой понимается, выполнение действия обратного последней команде (если это возможно). В случае если такое действие не возможно (например, нельзя «обратить» проверку влажности) вывести на экран соответствующее сообщение. Если действий не было произведено, при нажатии на кнопку написать сообщение.

**Дополнительная задача №4**

Для организации многопоточности реализовать очередь команд. Очередь должна работать по следующей схеме: при попытке выполнить любую операцию она попадает в очередь, как только количество операций в очереди будет равно пяти эти операции выполняются последовательно в порядке добавления каждая с задержкой в 2 секунды.

**Дополнительная задача №5**

Доработать решение – создать объект, который будет отслеживать состояние теплицы, в зависимости от применённых операций (например, после полива изменяется влажность почвы и воздуха). Данное решение (программный код) должно быть не изменяемым и не зависеть от добавления новых видов устройств.

**Дополнительная задача №6**

Доработать решение – создать «виртуальный пульт», данный пульт не должен иметь физического ограничения на количество кнопок, то есть количество кнопок должно быть расширяемым в ходе выполнения программы. При этом каждой команде пульта ставиться в соответствие уникальная текстовая метка (TestHumidity, TestTemperature). Управление теплицей должно осуществляться через командную строку вводом соответствующих команд.

### Задача №3. MEO GEO

На волне успехов ретрогейминга группа энтузиастов решила возродить старые хиты и разработала универсальную игровую платформу MEO GEO. Данная платформа комплектуется стандартным джойстиком, который имеет 8 положений, а также шестью кнопками. Кроме того, на джойстике располагается кнопка включения игры. Нашей задачей является написать программный код для связи джойстика и прилагаемых игр.

Проблемы начались с самого начала. Выяснилось, что игры компании SNK используют коды для обозначения действий персонажей (например, 1 – движение вперёд, 2 – движение вниз и тд), а компания Midway использует текстовые коды (move forward, move back, upperpunch).

Задача: разработать программное решение для джойстика. Решение должно удовлетворять требованиям:

1. Джойстик должен поддерживать назначение управляющих инструкций каждой игры на кнопки, при этом не должен меняться программный код джойстика.
2. Джойстик должен поддерживать подключение любых игр;
3. Игры не обязательно должны быть связаны одним родителем или интерфейсом.
4. Для тестирования: при нажатии любой кнопки необходимо выводить в консоль соответствующую информацию.

**Дополнительная задача №1**

Доработать решение, чтобы оно позволяло создавать макросы – комбинации кнопок. Каждое действие в макросе должно выполняться с секундной задержкой.

Протестировать на следующих примерах:

«Влево, вниз, влево, А,А»

«назад, назад, х»

**Дополнительная задача №2**

Разработать решение для создания и записи лога выполненных команд. Лог должен записываться в текстовый файл и содержать информацию в следующем виде:

Дата, название выполненной операции, результат выполнения операции

**Дополнительная задача №3**

Разработать решение для кнопки отмены последней команды. Под отменой понимается, выполнение действия обратного последней команде (если это возможно). В случае если такое действие не возможно (например, нельзя «отменить» включение автомата) вывести на экран соответствующее сообщение. Если действий не было произведено, при нажатии на кнопку написать сообщение.

**Дополнительная задача №4**

Для организации многопоточности реализовать очередь команд. Очередь должна работать по следующей схеме: при попытке выполнить любую операцию она попадает в очередь, как только количество операций в очереди будет равно пяти эти операции выполняются последовательно в порядке добавления каждая с задержкой в 2 секунды.

## Лабораторная работа №15. Адаптер

Разработать интерфейс *JSONImportable*, данный интерфейс должен описывать метод *public String GetJson()*.

Классы, реализующие интерфейс должны при реализации метода getJson возвращать строку с описанием объекта вида {title:<информация>,description:<информация>}.

Разработать класс *Cat*, который реализует интерфейс *JSONImportable.* Метод *GetJson* должен возвращать следующую информацию: {title: Имя кота,description: возраст кота}.

Создайте класс *JsonCollector*, с методом *void GetInfo(JSONImportable ji)*, который принимает на вход объекты и записывает в файл их представление в виде JSON.

Разработайте класс для второй лабораторной работы (согласно варианту). После чего реализуйте класс - адаптер для этого класса, который позволит использовать его с классом *JsonCollector.*

## Лабораторная работа №16. Фасад

Возьмите свой класс для лабораторной работы №2. В этом классе реализуйте (если этого не сделали) метод Print для вывода информации об этом классе в консоль. Метод должен выводить значения атрибутов класса.

Создайте класс-фасад ConsoleWorker. Класс содержит ссылку на управляемый объект и несколько методов (см. ниже). Для примера рассмотрим класс Cat (класс содержит атрибут name: string, age: int).

* + - 1. CreateObject – метод ждёт ввода через консоль для всех атрибутов класса, после чего создаёт объект клаcc. Созданный объект будем хранить в специальном атрибуте класса ConsoleWorker. Если мы уже создали объект, то выводим «объект уже создан» и ничего не делаем.

То есть в случае Cat он должен запросить на ввод число (возраст) и строку (имя). После чего создать объект класса Cat.

* + - 1. PrintObject – метод выводит в консоль объект класса. То есть вызывает метод класса (в примере Cat) – Print. Если объект не был создан, то пишем «Объект ещё не создан».
      2. EraseObject – стирает созданный объект, если объект не создавался то не делает ничего.
      3. PrintMenu – метод должен выводить в консоль меню для работы. После чего он должен ждать команду от пользователя. Команды имеют числовые коды. Пункты меню:

Создать объект класса

Вывести объект класса

Стереть объект

Пояснение: после вывода меню если пользователь введёт 1 – то вызывается метод CreateObject, 2- PrintObject, 3 – PrintMenu. При любом другом вводе пишем: «Неверная команда» и заново выводим меню.

Примечание: Каждый из методов 1-3 также выводит меню.