

## **Дисциплина «Программирование»**

### **Практическое задание №6**

**21–28 февраля**

**«Транспорт»**

Процесс выполнения этого задания состоит из трёх частей:

- 1) реализация программы, согласно описанным в условии требованиям;
- 2) оценивание работ других студентов;
- 3) период «споров».

#### ***Период реализации программы:***

После выдачи задания Вам необходимо выполнить его и загрузить архив (\*.zip) с решением задачи (полностью заархивировать решение, созданное средой разработки) до крайнего срока. В работе строго запрещается указывать ФИО, а также любую другую информацию, которая может выдать авторство работы. В случае выявления факта деанонимизации работы, работа может быть аннулирована.

#### ***Период взаимного оценивания:***

После окончания срока, отведенного на реализацию программы, начинается период взаимного оценивания. Вам будет необходимо проверить четыре работы других студентов, также выполнявших данное задание, согласно критериям оценивания. Проверка осуществляется анонимно: Вы не знаете, чью работу Вы проверяете, также, как и человек, кому принадлежит решение, не знает, кем была проверена его работа. Помимо оценки Вам необходимо указать комментарий к каждому из критериев. В случае, если Вы снижаете балл, необходимо подробно описать, за что именно была снижена оценка. Также рекомендовано писать субъективные комментарии, связанные с тонкостями программной реализации, предлагать автору работу более оптимальные на ваш взгляд решения. Снимать баллы за субъективные особенности реализации запрещено.

#### ***Период споров:***

По окончании периода взаимного оценивания, в случае если Вы не согласны с оценкой, выставленной одним из проверяющих, Вы можете вступить с этим студентом в анонимный диалог с целью уточнения причин выставления оценки по тому или иному критерию. В случае, если проверяющий не ответил Вам, или вы не пришли к обоюдному решению об изменении оценки, Вы можете поставить флаг, и работа будет рассмотрена одним из преподавателей. Флаги, поставленные

без предварительного обсуждения с проверяющим или после окончания периода выставления флагов, будут отклонены.

Также допустима перепроверка Вашей работы преподавателем. В таком случае оценки других проверявших работу не учитываются.

**Дедлайн загрузки работы:** 28 февраля 23:59

**Дедлайн проверки:** 3 марта 23:59

**Дедлайн обсуждения оценок:** 5 марта 23:59

**Дедлайн выставления флагов:** 6 марта 23:59

**Возможность поздней сдачи работы в этом задании предоставляться не будет.**

#### **Оценивание:**

$O_{\text{итог}} = 0,8 * O_{\text{задание}} + 0,2 * O_{\text{проверки}}$ , где  $O_{\text{задание}}$  – неокруглённая десятибалльная оценка за решение задания выставленная проверяющими с учётом возможной перепроверки преподавателем, а  $O_{\text{проверки}}$  неокруглённая десятибалльная оценка, выставленная студентами, чьи работы вы проверяли. Также в случае, если преподаватель обнаружит, что Вы проверили работы некачественно к Вам могут быть применены санкции в виде штрафа до 3 десятичных баллов от  $O_{\text{итог}}$  (в таком случае оценка вычисляется по формуле  $O_{\text{итог}} = O_{\text{задание}} - \text{Штраф}$ ).

Необходимо разработать библиотеку классов и консольное приложение согласно описанному заданию.

#### **Описание задания:**

В отдельной библиотеке классов **"EKRLib"** описать классы **Transport**, **Car**, **MotorBoat** и **TransportException**. Абстрактный класс **Transport** описывает абстрактное транспортное средство и содержит (можно создавать дополнительные члены для всех классов, но нельзя определять конструктор без параметров):

- 1) Строковое свойство **Model** (модель). Модель должна состоять только из заглавных латинских символов и цифр и длину ровно 5 символов. При несоответствии выбрасывать исключение типа **TransportException** с сообщением "Недопустимая модель <Model>".
- 2) Целочисленное свойство **Power** (мощность в лошадиных силах, тип `uint`), не может быть меньше 20 (выбрасывается исключение типа **TransportException** с сообщением "мощность не может быть меньше 20 л.с.").
- 3) Переопределенный метод **ToString()**, возвращающий строку формата "Model: <Model>, Power: <Power>".

4) Абстрактный метод **string StartEngine()**, переопределяемый в производных классах для получения звука (в виде строки), издаваемого транспортным средством.

5) Конструктор с параметрами для свойств **Model** и **Power**.

Класс **Car** с параметрическим конструктором, описывающий автомобиль (наследник класса **Transport**). Требования:

1) Переопределенный метод **ToString()**, приписывает слева строку "Car. " к строке, возвращаемой из одноименного метода базового класса.

2) Переопределенный метод **StartEngine()**, возвращающий строку "<Model>: Vroom", где <Model> - модель.

Класс **MotorBoat** с параметрическим конструктором, описывающий моторную лодку (наследник класса **Transport**). Требования:

1) Переопределенный метод **ToString()**, приписывает слева строку "MotorBoat. " к строке, возвращаемой из одноименного метода базового класса.

2) Переопределенный метод **StartEngine()**, возвращающий строку "<Model>: Brrrbrr", где <Model> - модель.

В консольной программе:

1) Создать массив типа **Transport[]**, состоящий из случайного (в диапазоне [6; 10)) количества элементов и заполнить его ссылками на случайным образом сгенерированные объекты **Car** и **MotorBoat** (выбор типа объекта равновероятен): **Model** и **Power** генерировать случайным образом (**Power** - в диапазоне [10; 100), **Model** - согласно спецификации). При возникновении исключения – повторить попытку создания объекта (текст сообщения из исключения выводить на экран). После создания объекта (используя **WriteLine**) выводить на экран результат вызова **StartEngine()**.

2) Информацию о транспортных средствах (возвращаемую методом **ToString()**) из сформированного массива записать в текстовые файлы "**Cars.txt**" и "**MotorBoats.txt**" (кодировка utf-16, расположение: папка с файлом решения), содержащие только сведения о машинах (**Cars.txt**) и моторных лодках (**MotorBoats.txt**) соответственно (информация о каждом транспортном средстве – с новой строки).

**Дополнительные требования (критерии оценивания):**

1. Для проверки в систему PeerGrade должен быть загружен архив с решением. Ожидается, что проверка работы будет проводиться, в среде разработки Visual Studio 2019, поэтому в случае выполнения задания с использованием другой среды разработки настоятельно рекомендуется проверить возможность открытия и запуска проекта в этой среде разработки.
2. Текст программы должен быть отформатирован согласно кодстайлу нашего курса<sup>1</sup>. Для автоматического форматирования в среде Visual Studio достаточно нажать Ctrl+K, D.

---

<sup>1</sup> [https://edu.hse.ru/pluginfile.php/1167815/mod\\_resource/content/2/C%20Sharp%20Coding%20Conventions.pdf](https://edu.hse.ru/pluginfile.php/1167815/mod_resource/content/2/C%20Sharp%20Coding%20Conventions.pdf)

3. Программа не должна завершаться аварийно (или уходить в бесконечный цикл) при любых входных данных. При некорректных входных данных программа должна выводить сообщение об ошибке и запрашивать ввод заново.
4. Программа должна быть декомпозирована. Каждый из логических блоков должен быть выделен в отдельный метод. Не строго, но желательно, чтобы каждый метод по длине не превышал 40 строк.
5. Интерфейс программы должен быть понятен. Пользователю должны выводиться подсказки о возможных дальнейших действиях и иные необходимые сообщения. Предполагается, что для успешного использования программы не требуется обращения к исходному коду программы.
6. Текст программы должен быть документирован. Необходимо писать, как комментарии перед методами, так и комментарии, поясняющие написанный внутри метода код. Названия переменных и методов должны быть на английском языке и отражать суть хранимых значений / выполняемых действий.
7. Работа должна быть выполнена в соответствии с заданием, критерии оценивания (“Rubric”) доступны для предпросмотра в системе “PeerGrade”.
8. Также оценивается общее впечатление, которое производит работа (как с точки зрения пользовательского интерфейса, так и с точки зрения написания кода программы). Эта часть оценки остаётся на усмотрение проверяющего.