«Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчёт защищён с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель Савченко В.В.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г.

Отчёт

Лабораторной работе №1

«Code review»

Студент группы ПИ 92 В.М. Шульпов

Преподаватель Савченко В.В.

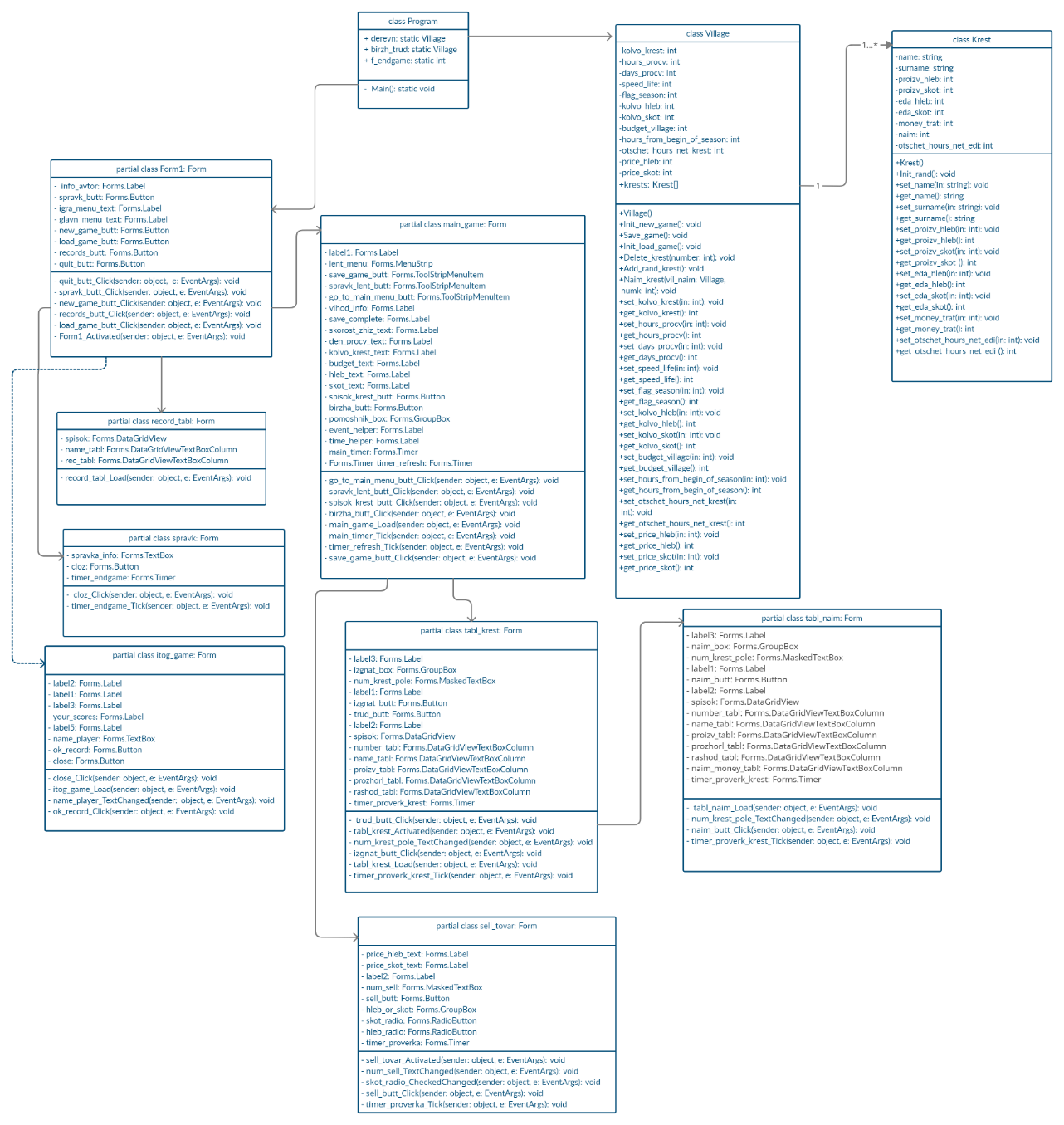
Барнаул 2022

**Задание**

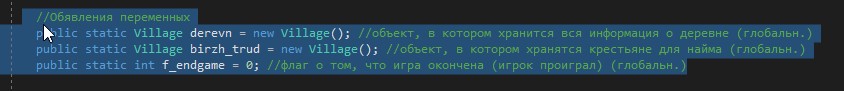
1. Для работы необходимо разделится на команды, и послать их список преподавателю.
2. Команда должна выбрать код прежнего готового проекта (например, курсовой работы по ООП) и нарисовать диаграмму классов для этого кода.
3. Далее необходимо закачать код проекта команды и диаграмму классов на общий репозиторий (адрес узнать у преподавателя).
4. Дождаться кода и диаграмм от других команд.
5. Результаты анализа кода записать в файл со своими фамилиями (doc, pdf)

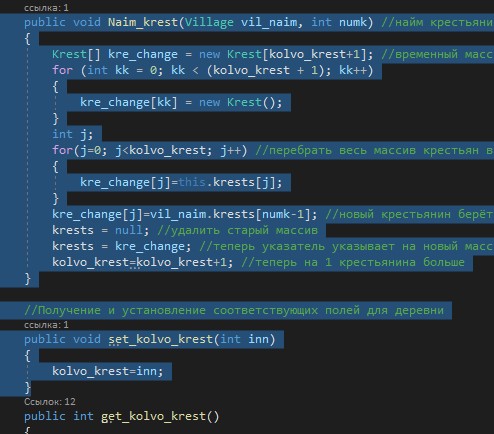
**Анализ студенческого кода (по курсовой работе на C# с использованием ООП и графического интерфейса)**

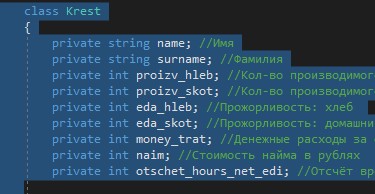
1. ✅ Наличие организации проекта в виде классов есть, диаграмма классов имеется



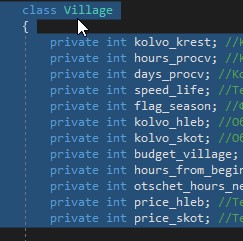
* 1. ❌ Методы классов не вынесены в интерфейсы
  2. ❌Не все названия классов осмысленные (Например: Form1), такое название не говорит о том, что это за форма
  3. В классе Village создаются и сохраняются в массиве экземпляры класса Krest, т.е. фактически есть композиция, но на диаграмме она показана как агрегация
  4. ✅Показаны связи: используется направленная агрегация, подписана мощность
  5. ✅Обозначение областей видимостей есть (+,-,#)
  6. И др.

1. ❌Отсутствие значимых комментариев к интерфейсу класса (интерфейсы не используются)
2. ✅Наличие комментариев к реализации методов
   1. ❌Излишняя информация в комментариях, длинные фразы: 
3. ✅Нет закомментированных участков кода в проекте, который передается на просмотр другим
4. ❌Есть противоречия следованию кода какому-то определенному соглашению о коде (code convention). Важно, чтобы код был написан единообразно, чтобы классы, методы и переменные назывались согласно единому стилю.
   1. ❌Одновременно используются разные стили именования функций (в C# лучше использовать camelCase или PascalCase для функций):

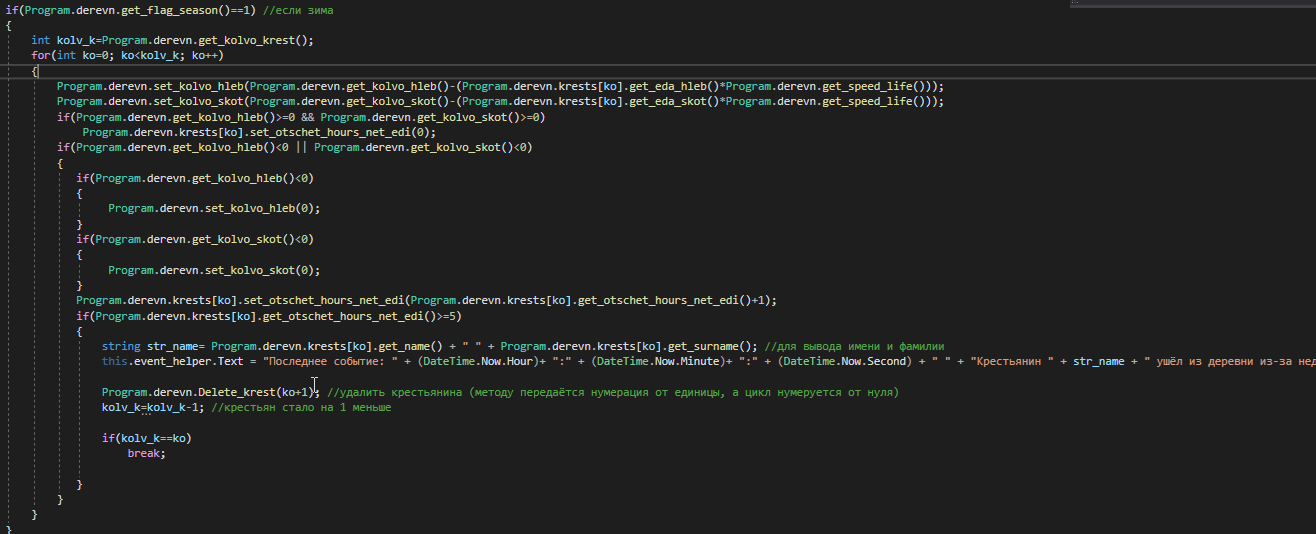


* 1. ❌ В качестве имён переменных используются русские слова («транслит»):

Смесь русских и английских слов в названиях переменных:



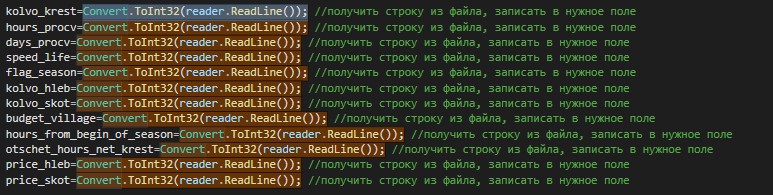
1. ❌Код сложно читать (огромное число вложенностей; перед тем, как производить арифметические операции было бы целесообразно записать каждое слагаемое в переменную для удобочитаемости)



1. ✅Наименования классов, полей, методов, параметров «говорящие», код описывает себя, но смесь русского и английского в названиях портят «общую картину»
2. ❌Используются краткие и не значащие ничего названия:



1. ❌Дублирование кода, длинные функции с повторяющимися параметрами можно было вынести в отдельную функцию:

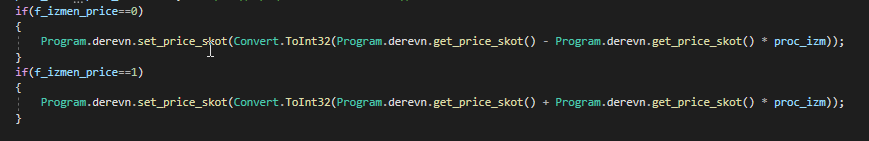


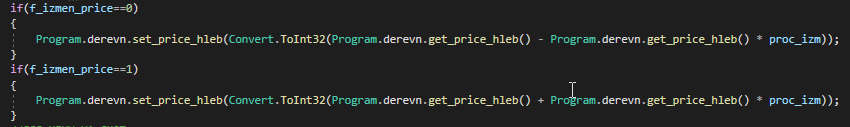
Создать функцию с параметром, куда уже передавать skot или hleb:



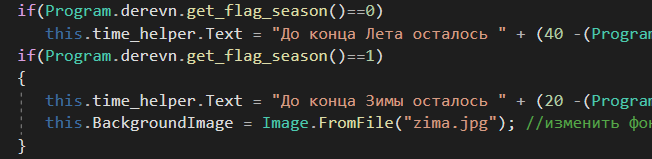


1. ❌Есть дублирование кода с небольшими изменениями

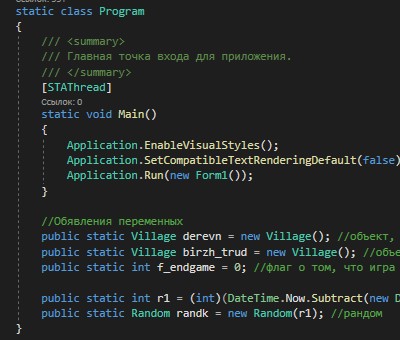




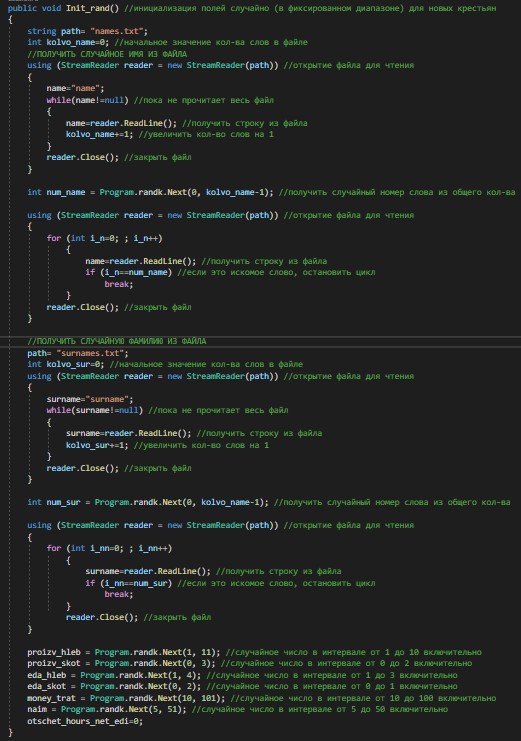
1. ❌Низкая связность не реализована, нет интерфейсов, абстракций, наследования.
2. ❌Есть магические константы



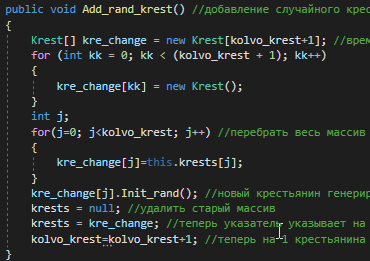
1. ✅Глобальные переменные только в одном главном классе:



1. ❌Есть методы слишком большого размера, их было целесообразно разбить на более мелкие приватные методы:



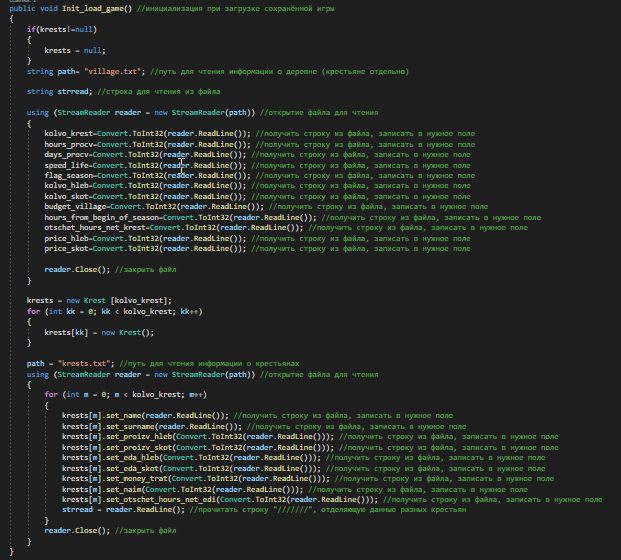
1. ❌Поля имеют правильные модификаторы доступа: те, что используются глобально public, а те, что используются глобальными функциями – private, protected в данной программе не используется.
2. ❌Высокое зацепление: классы имеют единственную зону ответственности. Это хорошо. Однако нет интерфейсов, абстракций, наследования.
3. ❌Не используются стандартные структуры данных, используются массивы:



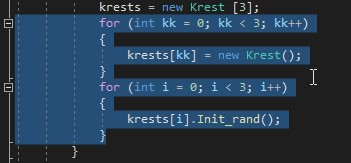
1. ❌Сторонние компоненты не использовались, стандартные библиотеки использовались в очень малом объеме, фреймворки не использовались.
2. ✅Документация по коду имеется в отчёте студента:

* Словесный алгоритм программы
* Структура программы
* Диаграмма классов
* Описание каждого метода
* Описание событий программы
* и другое

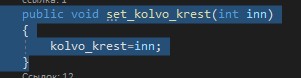
1. ✅ Динамическое определение типа (RTTI) не используется
2. ✅Не используется множественное наследование
3. ✅Графический интерфейс создавался через с помощью конструктора. Он находится в отдельных стандартных классах, унаследованных от класса Form. С логикой интерфейс не пересекается.
4. ✅Не используются сгенерированные методы и названия объектов
5. ❌Операции ввода вывода не везде отделены от логики

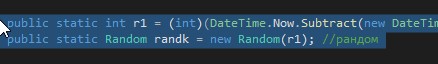


1. ❌Названия переменных для счётчиков циклов разные, когда можно было назвать одинаково (области видимости не пересекаются)

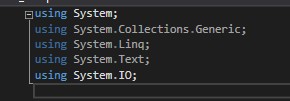


1. ❌Использование неинформативных названий переменных

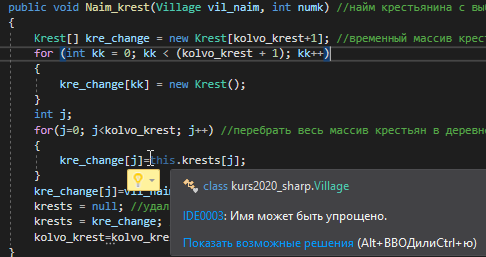


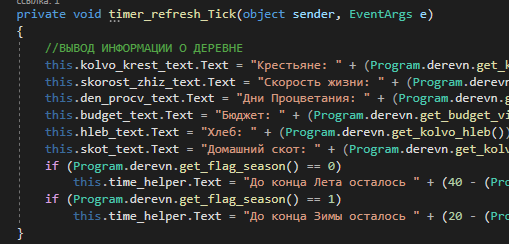


1. ❌Подключены неиспользуемые библиотеки

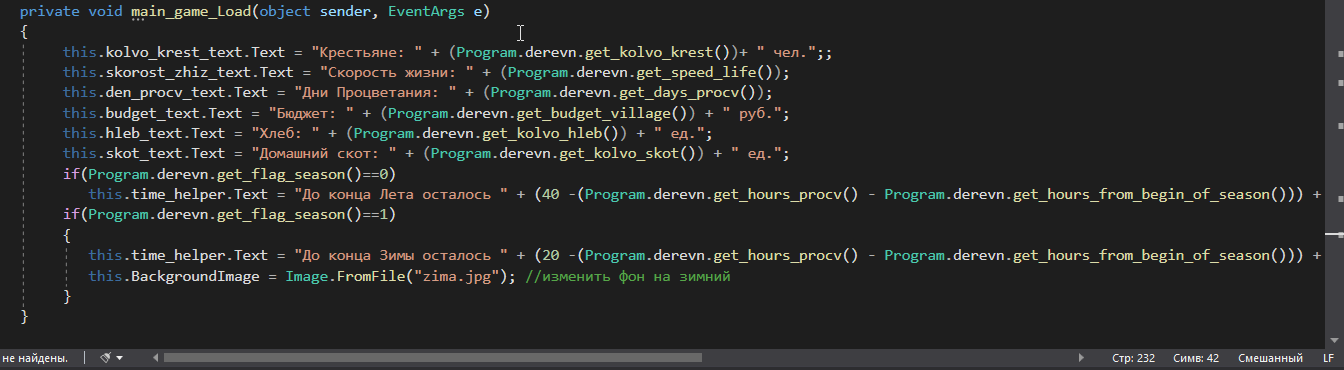


1. ❌Бессмысленное использование this:

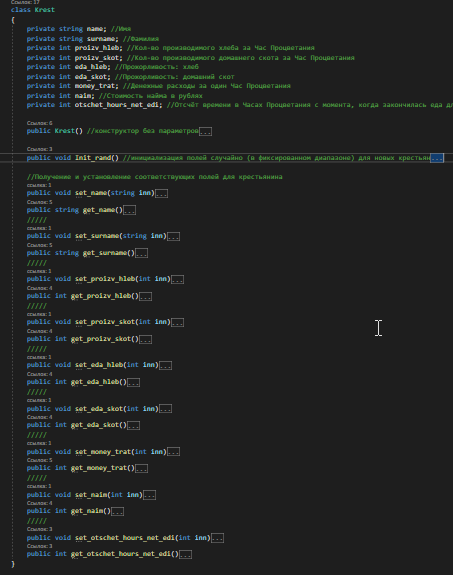




1. ❌Переносы длинных строк отсутствуют, приходится использовать прокрутку



1. ❌Классы, по сути своей, представляют собой лишь набор getter’ов и setter’ов, логика в них практически не реализуется:



1. Код, по моему мнению на 3 звезды из 5, однако это обосновывается малым опытом. Больше всего не понравилось смешивание русского и английского в именовании переменных и несоблюдение стиля кода.