Министерство образования Республики Беларусь

Белорусский Национальный Технический Университет

Факультет информационных технологий и робототехники

Кафедра «Программное обеспечение вычислительной техники

и автоматизированных систем»

**Отчёт**

по лабораторной работе № 11

по дисциплине ***«Объектно-ориентированное программирование»***

тема: «***Основы Объектно-Ориентированного Программирования в Java. Абстракция. Инициализация объектов***»

Название бригады: «In Gosling’s footsteps»

|  |  |  |
| --- | --- | --- |
| Исполнитель: |  | студенты группы 10702117  Бокшиц Е.А  Анисович М.А |
| Преподаватель: |  | Иванченко Виктор Викторович |

2018-2019 учебный год

**ЛАБОРАТОРНАЯ РАБОТА № 11**

Основы Объектно-Ориентированного Программирования в Java. Абстракция. Инициализация объектов

Цель работы:

Научиться работать с одномерными массивами в Java и закрепить приобретённые навыки на примере разработки интерактивных Java-приложений.

Общее задание:

1. Найти все элементы той части последовательности чисел Фибоначчи, значение последнего элемента которой не превосходит введённого пользователем значения.
2. Найти число трибоначчи по его порядковому номеру.

Числа трибоначчи – последовательность целых чисел *{tn}*, заданного с помощью рекуррентного соотношения: *t0 = 0, t1 = 0, t2 = 1, tn+3 = tn+2 + tn+1 + tn.* Пример ряда трибоначчи: *0, 0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149, …*

Условимся, что первые три числа ряда трибоначчи – это 0, 0 и 1, а если передаётся отрицательный индекс или ноль, то метод должен возвратить -1.

Требования к выполнению:

1. Необходимо спроектировать и реализовать алгоритмы решения заданий и UML-диаграмму взаимодействия классов и объектов разрабатываемой программной системы с отображением всех связей (отношений) между классами и объектами.
2. Каждая бригада берёт столько индивидуальных заданий, сколько человек находятся в ней. Индивидуальные задания в подгруппах не должны повторяться.
3. Основные классы системы должны быть самодостаточными, т.е. не зависеть, к примеру, от консоли! Любые типы отношений между классами должны применяться обосновано и лишь тогда, когда это имеет смысл.
4. На базе спроектированной программной системы реализуйте простейшее интерактивное консольное приложение. Используйте при реализации архитектурный шаблона проектирования Model-View-Controller, MVC.
5. Создаваемые классы необходимо грамотно разложить по соответствующим пакетам, которые должны иметь «адекватные» названия и быть вложены в указанные стартовые пакеты: by.bntu.fitr.povt.nameofteam.javalabs.lab11.
6. Попытайтесь реализовать все средства инициализации при создании соответствующих бизнес-объектов программной системы.
7. При выполнения задания необходимо по максимуму пытаться разрабатывать универсальный, масштабируемый и легко поддерживаемый и читаемый код.
8. Также рекомендуется придерживаться Single Responsibility Principle, SRP (принципа единственной ответственности): у каждого пакета, класса или метода должна быть только одна ответственность (цель), т.е. должна быть только одна причина изменить в дальнейшем соответствующий блок кода.
9. В соответствующих компонентах бизнес-логики необходимо предусмотреть «защиту от дурака».
10. Если логически не подразумевается или в задании иного не указано, то входными и выходными данными являются вещественные числа (числа с плавающей запятой).
11. Для задания размерности массива и конкретных его элементов необходимо использовать несколько способов (пользовательский ввод во время выполнения программы; генератор случайных чисел; …).
12. Для генерирования случайных чисел воспользуйтесь методами объекта класса java.util.Random, а для ввода пользовательских значений – соответствующими методами объекта класса java.util.Scanner.
13. Программа должна обязательно быть снабжена комментариями, в которых необходимо указать краткое предназначение программы, номер лабораторной работы и её название, версию программы, ФИО разработчиков, название бригады (если есть), номер группы и дату разработки. Исходный текст классов и демонстрационной программы рекомендуется также снабжать поясняющими краткими комментариями.
14. Программа должна быть снабжена дружелюбным и интуитивно понятным интерфейсом для взаимодействия с пользователем.
15. Интерфейс программы и комментарии должны быть на английском языке.
16. При проверки работоспособности приложения необходимо проверить все тестовые случаи.
17. Упаковать все классы программы в jar-архив, который должен быть запускаемым. Необходимо использовать два способа упаковки приложения в архив: с использованием IDE и стандартной Java-утилиты jar.
18. При выполнении задания не рекомендуется использовать интегрированные средства разработки (Integrated Development Environment, IDE). Лучше задействовать любой текстовый редактор и основные компоненты Java (компилятор – javac, утилиту для запуска JVM – java, утилиту для создания JAR-файлов - jar).
19. При разработке программ придерживайтесь соглашений по написанию кода на Java (Java Code-Convention) !!!

Результаты выполнения основного задания

Рисунок 1 – …

Рисунок 2 – …

…

**Результаты выполнения дополнительного задания**

Рисунок 1 – …

Рисунок 2 – …

…

Что мы узнали нового в процессе выполнения лабораторной работы (выводы)

1)

2)

3)

4)

5)

6)

…

*ПРИЛОЖЕНИЕ А*

Листинг исходного кода класса(ов)