Аттестационная работа №1

**Промежуточная аттестация Модуль 1 «Введение в разработку/Введение в Java»**

**Формулировка задания:**  
Необходимо реализовать приложение, принимающее список  
пользователей, продуктов и обрабатывающее покупку пользователя.

**Подробное описание функционала приложения**  
1. Создать классы Покупатель (Person) и Продукт (Product).  
Характеристики Покупателя: имя, сумма денег и пакет с продуктами  
(массив объектов типа Продукт). Имя не может быть пустой строкой и не может  
быть короче 3 символов. Деньги не могут быть отрицательным числом.  
Если Покупатель может позволить себе Продукт, то Продукт добавляется  
в пакет. Если у Покупателя недостаточно денег, то добавление не происходит.  
Характеристики Продукта: название и стоимость. Название продукта не  
может быть пустой строкой, оно должно быть. Стоимость продукта не может  
быть отрицательным числом.  
2. Поля в классах должны быть private, доступ к полям осуществляется  
через геттеры и сеттеры или конструктор класса.  
3. В классах переопределены методы toString(), equals(), hashcode().  
4. Создать в классе App метод main и проверить работу приложения.  
Данные Покупателей и Продукты вводятся с клавиатуры, для считывания  
данных потребуется использовать класс Scanner и его метод nextLine().  
Продукты в цикле выбираются покупателями по очереди и, пока не введено  
слово END, наполняется пакет.  
5. Обработать следующие ситуации:  
а. Если покупатель не может позволить себе продукт, то напечатайте  
соответствующее сообщение ("[Имя человека] не может позволить себе  
[Название продукта]").  
б. Если ничего не куплено, выведите имя человека, за которым  
следует "Ничего не куплено".

в. В случае неверного ввода - сообщение: "Деньги не могут быть  
отрицательными", пустого имени - сообщение: "Имя не может быть  
пустым" или длина имени менее 3 символов – сообщение: "Имя не может  
быть короче 3 символов".  
Программа реализуется в отдельной ветке git attestation/attestation01. При  
сохранении состояния программы (коммиты) пишется сообщение с описанием  
хода работы по задаче.  
В корне папки с программой должен быть файл .gitignore.  
Программа локально коммитится и публикуется в репозиторий GitHub на  
проверку.

**Тестовые данные**

Павел Андреевич = 10000; Анна  
Петровна = 2000; Борис = 10  
Хлеб = 40; Молоко = 60; Торт = 1000;  
Кофе растворимый = 879; Масло = 150  
Павел Андреевич - Хлеб  
Павел Андреевич - Масло  
Анна Петровна - Кофе растворимый  
Анна Петровна - Молоко  
Анна Петровна - Молоко  
Анна Петровна - Молоко  
Анна Петровна - Торт  
Борис - Торт  
Павел Андреевич - Торт  
END

**Ожидаемый результат :**  
Павел Андреевич купил Хлеб  
Павел Андреевич купил Масло  
Анна Петровна купил Кофе  
растворимый  
Анна Петровна купил Молоко  
Анна Петровна купил Молоко  
Анна Петровна купил Молоко  
Анна Петровна не может позволить себе  
Торт  
Борис не может позволить себе Торт  
Павел Андреевич купил Торт  
Павел Андреевич - Хлеб, Масло, Торт  
Анна Петровна - Кофе растворимый,  
Молоко, Молоко, Молоко  
Борис - Ничего не куплено

Женя = 0  
Мороженое = 200  
Женя - Мороженое  
END  
Женя не может позволить себе  
Мороженое  
Женя - Ничего не куплено  
Света = -3 Деньги не могут быть отрицательными  
Фа = 100 Имя не может быть короче 3 символов

Тестовые данные вводятся по очереди, сначала проверяется успешный  
кейс, потом не успешные, для неуспешных достаточно получить сообщение о  
невозможности покупки или ошибке валидации. После данного сообщения,  
сделать скриншот и завершить программу.  
Планируемый результат:  
1. Ссылка на программу в репозитории GitHub;  
2. Отчёт со скринами выполнения задач - постановка задачи, код  
задачи и результат в консоли Intellij Idea.  
Перечень инструментов, необходимых для реализации деятельности:  
Персональный компьютер, JDK 17/21 (либо OpenJDK 17/21), Intellij Idea  
для разработки на Java, GIT.

Решение:





