Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

Московский приборостроительный техникум

ПРАКТИЧЕСКАЯ РАБОТА № 6

Дисциплина: «МДК 01.04 Системное программирование»

Тема: «Работа в паре»

Специальность: 09.02.07 Информационные системы и программирование

|  |  |
| --- | --- |
| Преподаватель  Никонова А.А.  «\_\_\_» \_\_\_\_\_\_\_\_\_\_2024 г. | Выполнили студентки  Группы П50-5-22  Овцунова А.А. и Гладкова А.А. |

**Цель работы**: в данной практической работе необходимо написать программу, которая соответствует базовым принципам ООП и содержит в себе как минимум 3 класса и 6 функций.

**Ход работы**:

1. Объявляем класс Practicant, который представляет участника фестиваля. Он содержит приватные поля name, balance и ссылку на объект MerchCollection, чтобы реализации покупки мерча из коллекции.

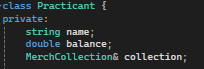


Рисунок 1. Создание класса

1. Далее создадим конструктор класса, который принимает в качестве аргументов имя участника, баланс и ссылку на коллекцию мерча.

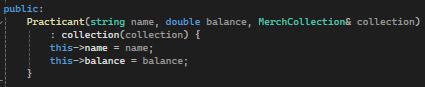


Рисунок 2. Конструктор

1. Создадим метод, с помощью которого будем получать текущий баланс.



Рисунок 3. Функция взятия баланса

1. Сначала создается строковая переменная product и переменная index. Если в коллекции мерча нет, выводится сообщение о том, что мерча нет, и метод завершается. Если же нет, то запускается цикл do-while внутри которого выводится список доступного мерча из коллекции и запрашивается у пользователя выбор товара.

После выбора товара проверяется, достаточно ли у участника средств на балансе для его покупки.

Если участник имеет достаточно средств, цена товара вычитается из его баланса, и статус товара изменяется на "куплено" в коллекции мерча.

Если участник не имеет достаточно средств, выводится сообщение о недостатке средств для покупки выбранного товара.



Рисунок 4. Функция покупки мерча

1. Сначала метод проверяет, есть ли косплееры в векторе cosplayers. Если вектор пуст, выводится сообщение "Пока что никого нет..." и выполнение метода завершается в ином случае – выводит список доступных косплееров.

После вывода списка косплееров пользователю предлагается ввести номер косплеера. Далее проверяется, что введенное значение является числом, и находится в допустимом диапазоне от 1 до размера вектора cosplayers.

После выбора косплеера пользователю предлагается выбрать действие для взаимодействия с ним. После выполнения выбранной операции пользователю снова предлагается выбрать действие.



Рисунок 5. Функция для взаимодействия с косплеерами

1. Далее создадим 3 типовых метода Photoshoot(), Chat() и ParticipateInContest() основная задача которых – реализовать действия, которые может выполнять участник во время взаимодействия с косплеерами.

Для того, чтобы понять принцип работы данных функций рассмотрим одну из них:

Метод Photoshoot() позволяет участнику фестиваля устроить фотосессию с выбранным косплеером. Сначала метод выводит сообщение о начале фотосессии, затем участнику предлагается выбрать позу для фотосессии посредством вывода списка доступных поз.

После выбора позы выводится сообщение о завершении фотосессии, поздравляя участника с завершением фотосессии и отмечая успешное взаимодействие.

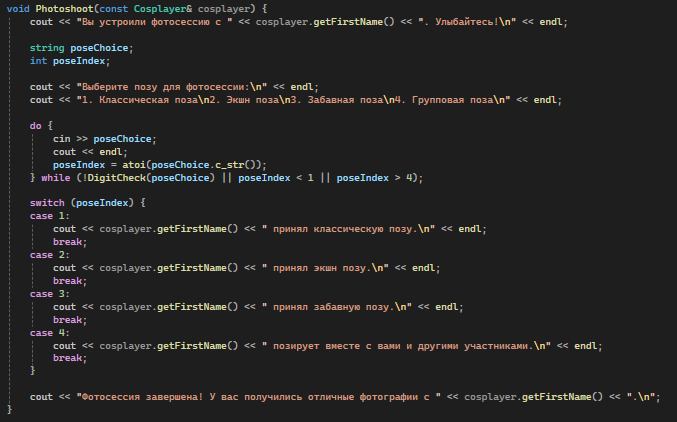


Рисунок 6. Функция для фотосессии с косплеерами

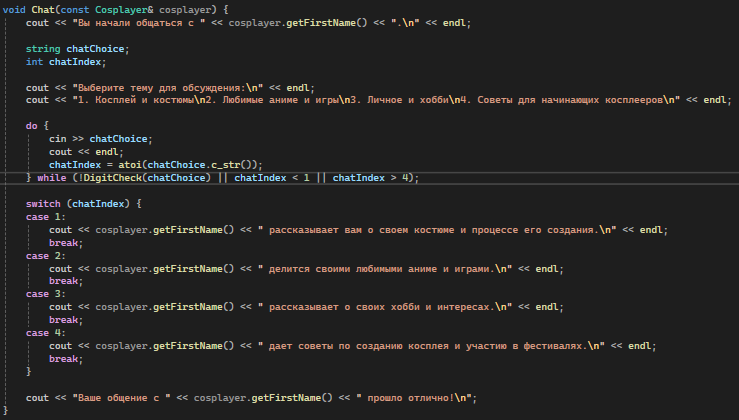


Рисунок 7. Функция для общения с косплеерами

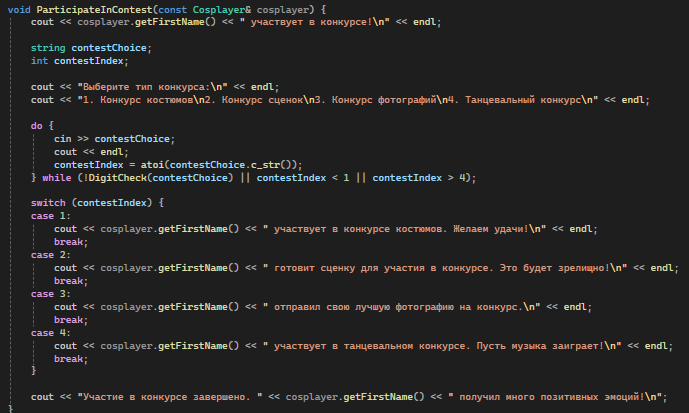


Рисунок 8. Функция для участия в конкурсах с косплеерами

1. Объявим функцию PracticantAct, которая принимает на вход объект MerchCollection и ссылку на вектор cosplayers.

В данной функции пользователь должен ввести имя и баланс. После ввода баланса происходит проверка на положительное значение. Далее создадим объект типа Practicant с переданными параметрами: имя, баланс и объект коллекции collection.

Запускается цикл, в котором пользователь должен выбрать один из пунктов действий и в зависимости от выбора пользователя выполняются определенные действия: покупка мерча, взаимодействие с косплеерами, проверка баланса или выход из цикла.

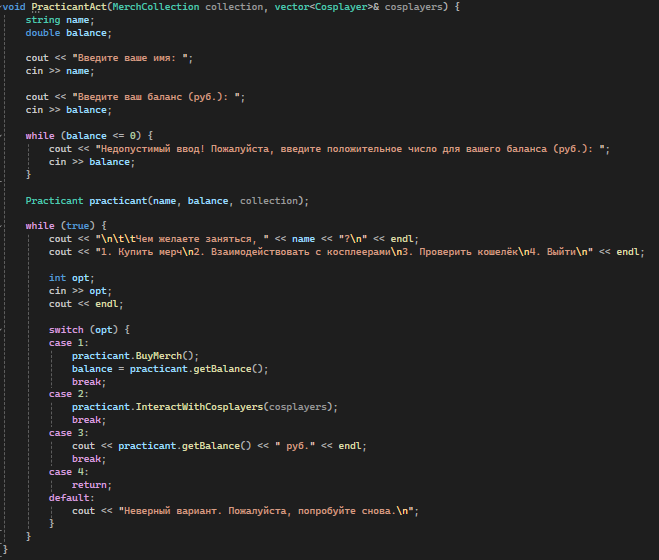


Рисунок 9. Функция для работы с участником

Результат:

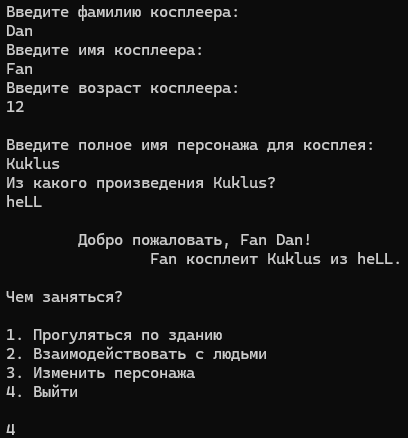


Рисунок 10. Создание косплеера\_1

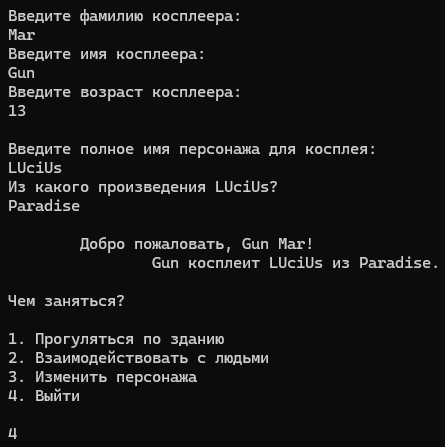


Рисунок 11. Создание косплеера\_2

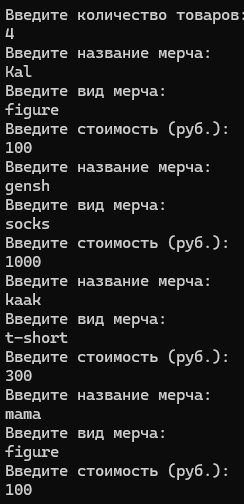


Рисунок 12. Создание художника и его товаров

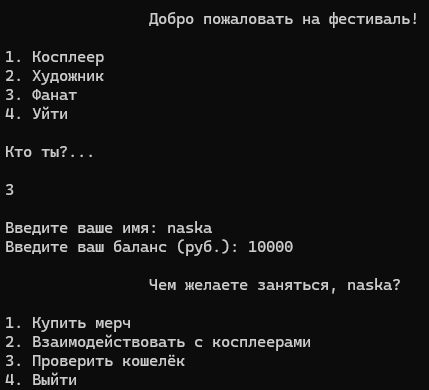


Рисунок 13. Создание посетителя

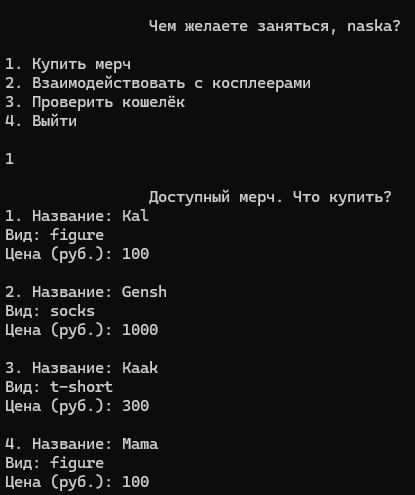


Рисунок 14. Покупка мерча\_1

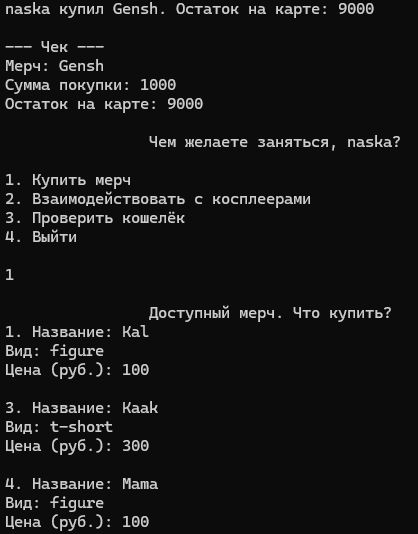


Рисунок 15. Чек и Покупка мерча\_2

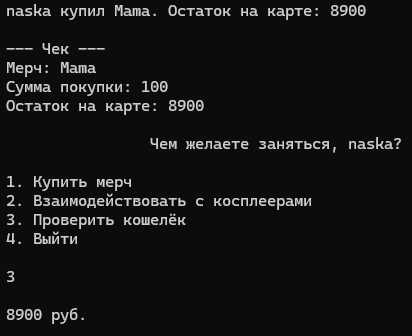


Рисунок 16. Чек и Проверка кошелька

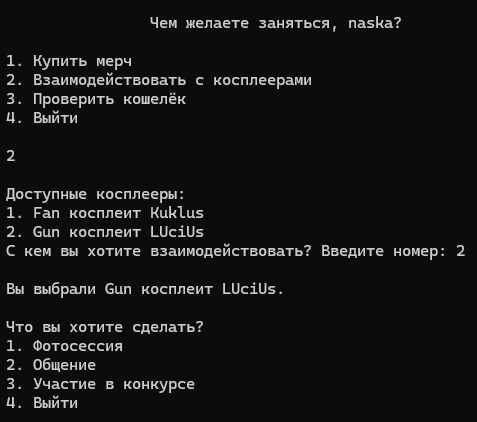


Рисунок 17. Взаимодействие с косплеерами

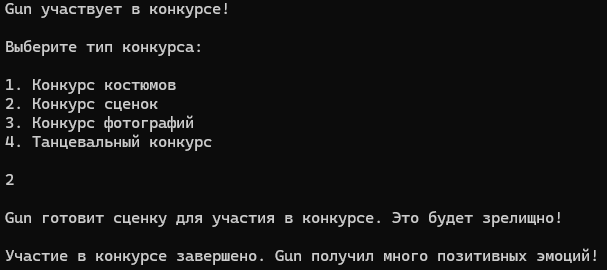


Рисунок 18. Пример взаимодействия с косплеером

**Вывод**: в ходе данной практической работы была написана программа, которая соответствует базовым принципам ООП и содержит в себе 4 класса и более 6 функций.