МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. І.І. МЕЧНИКОВА

ФАКУЛЬТЕТ МАТЕМАТИКИ, ФІЗИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**Курсова робота**

З дисципліни: “Програмування”

на тему:

Створення ієрархії класів на тему «Первісне плем'я»

виконав: Мовчан І.О,

студент 1 курсу факультету

МФІТ

керівник:

канд. ф. -м. н., доц.

Антоненко О. С.

Одеса – 2022

1. **Постановка задачі**

Створити класи: "Людина", а також спеціалізовані класи за видами діяльності - "Збирач", "Рибалка", "Мисливець". У людини передбачити набір характеристик - наприклад, таких як фізична сила, які впливають на успішність його професійної діяльності (для різних спеціалізацій по-різному). Під час виконання відповідні характеристики поліпшуються. Також передбачити наявність умінь, властивих усім людям – як є, спати тощо.

Промоделювати взаємодію людей у первісному племені.

1. **Алгоритм роботи програми**

Користувач має змогу інтерактивно корситуватися програмою. Тобто можна викликати методи в різному порядку та різноманітну кількість рядів. Основні методи реалізовані в класі Human.

Спочатку користувач потрапляє на створення персонажу. Там йому буде запропановано вибрати професію, а потім розподілити очки атрибутів для того щоб зробити персонажу деякі характеристики. Є 3 види атрибутів: сила, спритність та інтелект. Кожен із них допомагає людині при виконанні праці, а саме: сила випромінює опірність до травм, спритність дозволяє уникати їх частіше, а інтелект допомагає отримувати більше видобутку.

Далі користувач може вибрати 5 методів роботи зі своїм персонажем: printAttributes, printHealthInfo, work, eat, sleep та 1 інформаційний метод: help, який виводить усі можливі команди для взаємиодії. Так як продукти додаються на полицю, то і основні методи реалізовані в класі Human.

Якщо у персонажу, не залишилося життів, то людина не може продовжувати роботу з програмою.

1. **Програмна реалізація**

Проєкт налічує в собі 4 класи, а саме:

1. Human,
2. Hunter,
3. Collector,
4. Fisher,

Реалізовано базовий, абстрактний клас Human, який включає в себе наступні поля:

**int** strength;

сила

**int** agility;

спритність

**int** intelligence;

інтелект

**int** remaining\_food;

їжа яка залишилась

**int** earned\_food;

зароблена їжа

**int** hp;

кількість здоров’я

**int** hunger;

голод

**int** fatigue;

втома

**int** upgrade\_points;

кількість очок покращення атрибутів

Конструктори класа Human:

Human();

конструктор без параметрів

Human(int strength, int agility, int intelligence);

конструктор з параметрами

~Human();

деструктор

Основні методи класа Human:

**void** printAttributes() const;

виводить атрибути персонажу

**void** pritnHealthInfo() const;

виводить кількість здоров’я, їжі та втоми персонажу

**int** getStrength() const;

повертає значення поля strength для користування в інших класах

**int** getAgility() const;

повертає значення поля strength для користування в інших класах

**int** getIntelligence() const;

повертає значення поля intelligence для користування в інших класах

**int** getHp() const;

повертає значення поля hp для користування в інших класах

**int** getFood() const;

повертає значення поля remaining\_food для користування в інших класах

**int** getHunger() const;

повертає значення поля hunger для користування в інших класах

**int** getFatigue() const;

повертає значення поля fatigue для користування в інших класах

**int** getUpgradePoints() const;

повертає значення поля upgrade\_points для користування в інших класах

**virtual void** work();

віртуальний метод, реалізований для подальшої реалізації праці разних професій персонажів

**void** eat();

метод, який виконує дію «їсти», тобто додає очки здоров’я, але віднімає кількість залишившиїся їжі

**void** sleep();

метод, який виконує дію «спати», тобто додає очки здоров’я та втоми, але віднімає кількість голоду персонажа

**void** upgradeStrength();

підвищує атрибут «сила» на 1 одиницю

**void** upgradeAgility ();

підвищує атрибут «спритність» на 1 одиницю

**void** upgradeIntelligence ();

підвищує атрибут «інтелект» на 1 одиницю

**void** isDead () const;

перевіряє чи залишились очки здоров’я у персонажа

**void** checkHealth ();

перевіряє чи залишились очки голоду та втоми у персонажу, та якщо вони дорівнюють нулю, віднімає очки здоров’я персонажа

**void** statsRoundUp ();

метод, який перевіряє чи поля не більше 100 та менше 0, та якщо це стається, округляє їх

**virtual void** getInjured ();

метод, який віднімає очки здоров’я персонажу при поранені

Реалізовано клас-спадкоємець Hunter від класу Human, який включає в себе наступні методи:

Конструктори класа Hunter:

Hunter();

конструктор без параметрів

Hunter(int strength, int agility, int intelligence);

конструктор з параметрами

~Hunter();

деструктор

Основні методи класа Hunter:

**void** work () override;

метод «праці», який додає випадкову кількість їжі, в залежності від атрибутів песронажу та з деяким шансом персонаж може травмуватись працюючи

**void** getInjured () override;

метод, який віднімає очки здоров’я персонажу при поранені (для кожної професії різна кількість та шанс поранення)

Аналогічні методи були реалізовані для класів-спадкоємців:

Collector та Fisher

1. **Висновки**

Під час виконання даної роботи, я познайомився з принципами об'єктно-орієнтованого програмування (ООП) на мові програмування С++. Зокрема, я ретельно вивчив поняття ієрархії класів та їх взаємодію між собою.

Для написання правильних методів було використано різні алгоритми. Одним з ключових аспектів у цій роботі було активне використання генераторів псевдовипадкових чисел, оскільки потрібно було генерувати шанси тих чи інших випадкових результатів.

Отже, ця курсова робота чітко продемонструвала мені сутність об'єктно-орієнтованого програмування та його роботу в практиці.