

자바 프로젝트

김솔 김채운 최문길 하예진

프로젝트에 대해

- 첫 회의로 여러가지 주제 아이디어 교환
- (건강 관리, 도서 시스템, 퀴즈 게임, 다마고치 등)
- 나왔던 주제중 반응이 나쁘지 않았던 다마고치 게임을 선택해
자바로 구현

동작 화면

캐릭터를 선택하세요: 1. 최문길 2. 하예진 3. 김슬

1

메뉴를 선택하세요: 1. 상태창 보기 2. 애정도 올리기 3. 통신하기 4. 종료

1

최문길의 상태창

체력: 100

스트레스: 0

친밀도: 0

포만감: 0

질병: 0

성장상태: 낮가림 단계

메시지:

정신상태: 위험 상태

메뉴를 선택하세요: 1. 상태창 보기 2. 애정도 올리기 3. 통신하기 4. 종료

2

당신은 최문길의 옆에서 소리를 질렀습니다.

최문길의 스트레스가 10 증가하고, 당신과의 친밀도가 5 감소했습니다.

최문길은 깜짝 놀라 당황한 표정을 지었습니다.

메뉴를 선택하세요: 1. 상태창 보기 2. 애정도 올리기 3. 통신하기 4. 종료

Main.java

```
package Tamagotchi완성;

import java.util.InputMismatchException;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Tamagotchi pet = null;

        while (pet == null) {
            try {
                System.out.println("캐릭터를 선택하세요: 1. 최문길 2. 하예진 3. 김슬");
                int petChoice = sc.nextInt();

                switch (petChoice) {
                    case 1:
                        pet = new 최문길();
                        break;
                    case 2:
                        pet = new 하예진();
                        break;
                    case 3:
                        pet = new 김슬();
                        break;
                    default:
                        System.out.println("잘못된 선택입니다. 다시 시도하세요.");
                }
            } catch (InputMismatchException e) {
                System.out.println("잘못된 입력입니다. 숫자를 입력해주세요.");
                sc.next();
            }
        }
    }
}
```

- Scanner
- While문
- 예외 처리
- 추상 클래스

TamaPet.java

```
package Tamagotchi완성;
```

```
public abstract class TamaPet {
```

```
    protected int 체력;  
    protected int 스트레스;  
    protected int 친밀도;  
    protected int 포만감;  
    protected int 질병;  
    protected static final int MAX_STAT = 200;
```

```
    public TamaPet(int 체력, int 스트레스, int 친밀도, int 포만감, int 질병) {
```

```
        this.체력 = 체력;  
        this.스트레스 = 스트레스;  
        this.친밀도 = 친밀도;  
        this.포만감 = 포만감;  
        this.질병 = 질병;  
        validateStatus();
```

```
    }
```

```
    protected void validateStatus() {
```

```
        체력 = Math.max(0, Math.min(체력, MAX_STAT));  
        스트레스 = Math.max(0, Math.min(스트레스, MAX_STAT));  
        친밀도 = Math.max(0, Math.min(친밀도, MAX_STAT));  
        포만감 = Math.max(0, Math.min(포만감, MAX_STAT));  
        질병 = Math.max(0, Math.min(질병, MAX_STAT));
```

```
    }
```

```
    protected void adjust체력(int amount) {
```

```
        체력 += amount;  
        validateStatus();
```

```
    }
```

- 속성 정의
- 생성자
- 상태 유효성 검증
- 상태 조정 메서드
- 추상 메서드

각 캐릭터 클래스의 공통 요소

```
@Override
public String 캐릭터선택() {
    return "김슬 캐릭터를 선택했습니다.";
}

@Override
public String 성장상태() {
    String 성장상태 = 성장상태(친밀도);
}
```

- override
- (상태창, 애정도올리기, 통신하기) 메서드
- 이벤트 선택창, 랜덤 이벤트

```
@Override
public String 애정도올리기() {
    System.out.println("무엇을 할까요? 1. 간식 먹이기 2. 샤워하기 3. 화장실 가기 4. 놀아주기");
    int 선택 = sc.nextInt();
    sc.nextLine();

    int 결과번호 = random.nextInt(3);
}
```

김솔.java

- 상태 체크
- 상세한 상태 변화

```
package Tamagotchi완성;

import java.util.Random;

public class 김솔 extends T {
    private Random random;
    private Scanner sc;

    public 김솔() {
        super(50, 30, 0, 5);
        this.random = new Random();
        this.sc = new Scanner(System.in);
    }

    @Override
    public String 캐릭터선택() {
        return "김솔 캐릭터를 선택했습니다.";
    }

    @Override
    public String 상태창() {
        String 성장상태 = 성장상태;
        String 건강상태 = 건강상태;

        return "-----\n"
            + "상태창\n"
            + "-----";
    }

    private String 성장상태() {
        if (성장상태 < 80) {
            adjust성장상태(-20);
            adjust스트레스(20);
        } else {
            adjust성장상태(10);
            return "마침 화장실에 가고 싶었습니다.\n > 화장실가기()";
        }
    }

    private String 화장실가기() {
        if (질병 < 80) {
            adjust질병(-20);
            adjust스트레스(20);
        } else {
            adjust질병(-20);
            adjust진밀드(10);
            return "마침 화장실에 가고 싶었습니다.\n > 화장실가기()";
        }
    }

    private String 휴식하기() {
        if (체력 < 스트레스) {
            adjust스트레스(-20);
            adjust체력(10);
            adjust진밀드(10);
            return "적절한 휴식을 취했습니다.\n > 스트레스확인()";
        } else {
            adjust스트레스(-20);
            adjust체력(-10);
            return "휴식이 과해 늘어납니다. \n > 스트레스확인()";
        }
    }

    private String 스트레스확인() {
        if (체력 > 포만감) {
            adjust체력(20);
            adjust진밀드(10);
        }
    }

    private String 포만감확인() {
        if (포만감 < 80) {
            adjust포만감(-20);
            adjust스트레스(20);
        } else {
            adjust포만감(10);
            return "마침 배가 고파서 먹고 싶습니다.\n > 먹이기()";
        }
    }

    private String 먹이기() {
        if (체력 < 80) {
            adjust체력(-20);
            adjust스트레스(20);
        } else {
            adjust체력(10);
            return "마침 배가 고파서 먹고 싶습니다.\n > 먹이기()";
        }
    }

    private String PvP() {
        System.out.println("컴퓨터USER 와(과) 대결합니다.");
        System.out.println("-----");

        int 상대체력 = 50 + random.nextInt(51);

        while (true) {
            if (체력 > 상대체력) {
                System.out.println("ROUND 1");
                System.out.println("\n상대방 체력: " + 상대체력 + " vs 내 체력: " + 체력 + "\n");
                System.out.println("1연승 하였습니다.");
                System.out.println("-----");

                상대체력 = 100 + random.nextInt(51);
                if (체력 > 상대체력) {
                    System.out.println("ROUND 2");
                    System.out.println("\n상대방 체력: " + 상대체력 + " vs 내 체력: " + 체력 + "\n");
                    System.out.println("2연승 하였습니다.");
                    System.out.println("-----");

                    상대체력 = 150 + random.nextInt(51);
                    if (체력 > 상대체력) {
                        System.out.println("LAST ROUND");
                        System.out.println("\n상대방 체력: " + 상대체력 + " vs 내 체력: " + 체력 + "\n");
                        System.out.println("3연승 하였습니다.");
                        System.out.println("-----");

                        adjust체력(50);
                        return "마침 승리했습니다. \n > 승리()";
                    }
                }
            }
        }
    }
}
```

최문길.java

- 랜덤 메시지
활동별 메시지

```
package Tanagotchi;

import java.util.ArrayList;

public class 최문길 extends Tanagotchi {
    private Scanner sc;
    private ArrayList<String> 활동별_메시지 = new ArrayList<>(); // '최문길' 메시지

    public 최문길() {
        super(100, 0, 0, 0, 0);
        this.sc = new Scanner(System.in); // Scanner 초기화
    }

    @Override
    public String 자막미선택() {
        return "최문길 캐릭터를 선택하였습니다.";
    }

    @Override
    public String 상태창() {
        String 성장상태 = 성장상태(진행도);
        String 정신상태 = 정신상태(재력, 스트레스, 포만감, 기분);

        StringBuilder result = new StringBuilder();
        for (String str : 활동별_메시지) {
            result.append(str);
        }

        return "최문길의 상태창\n" + "재력: " + 재력 + "\n" + "스트레스: " + 스트레스 +
            + "활력: " + 활력 + "\n" + "정신상태: " + 정신상태 + "\n" + "메시지: "
            + "\n";
    }

    public String 성장상태(int 진행도) {
        if (진행도 < 30) {
            return "초기 단계";
        } else if (진행도 < 50) {
            return "중기 단계";
        } else if (진행도 < 80) {

```

```
@Override
public String 통신하기() {
    System.out.println("무엇을 할까요? 1. 놀기 2. 아이템 교환 3. 운동 4. 음악 5. 영화");
    int 선택 = sc.nextInt();
    sc.nextLine(); // 입력 바퍼 비우기

    String 통신;

    switch (선택) {
        case 1:
            통신 = "당신은 최문길과 함께 집에서 여러 가지 재미있는 놀이를 즐겼습니다.";
            통신 = "최문길은 기분이 좋아졌고, 스트레스가 줄어들었습니다.";
            통신 = "최문길은 새로운 장난감을 발견하고 즐거움을 느꼈습니다.";
            break;

        case 2:
            통신 = "당신은 최문길과 아이템을 교환했습니다.";
            통신 = "최문길은 당신이 준 새로운 아이템에 만족했습니다.";
            통신 = "교환한 아이템 덕분에 최문길의 표현력이 커졌습니다.";
            break;

        case 3:
            통신 = "당신은 최문길과 함께 운동을 했습니다.";
            통신 = "간단한 운동을 통해 최문길은 체력을 기르고 스트레스를 해소했습니다.";
            통신 = "운동이 끝난 후, 최문길은 상쾌한 기분을 느낍니다.";
            break;

        case 4:
            통신 = "당신과 최문길은 함께 음악을 들었습니다.";
            통신 = "최문길은 마음이 편안해졌고, 리듬에 맞춰 춤을 추었습니다.";
            통신 = "음악을 들으며 최문길의 창의력이 자극되었습니다.";
            break;

        case 5:
            통신 = "당신은 최문길과 함께 영화를 보았습니다.";
            통신 = "영화를 보며 최문길은 여러 가지 감정을 느끼고, 새로운 통찰을 얻었습니다.";
            통신 = "영화 후 보고 느끼는 흥미와 감정을 느낍니다.";

```


하예진.java

```
@Override
public String 애정도올리기() {
    System.out.println("무엇을 할까요? 1. 간식 먹이기 2. 샤워하기 3. 화장실 가기 4. 놀아주기");
    int 선택 = sc.nextInt();
    sc.nextLine();

    String 결과;

    switch (선택) {
        case 1:
            adjust포만감(20);
            adjust질병(-5);
            결과 = "간식을 드립니다. (포만감 +20)(질병 -5)";
            break;
        case 2:
            adjust친밀도(10);
            adjust질병(30);
            결과 = "샤워를 합니다. (친밀도 +10)(질병 +30)";
            break;
        case 3:
            adjust스트레스(-10);
            adjust질병(10);
            결과 = "화장실을 갑니다. (스트레스 -10)(질병 +10)";
            break;
        case 4:
            adjust체력(-10);
            adjust친밀도(20);
            adjust질병(-20);
            결과 = "놀아줍니다. (체력 -10)(친밀도 +20)(질병 -20)";
            break;
        default:
            결과 = "잘못된 선택입니다.";
            break;
    }
}
```

- 간략한 수치 표시

```
public String 외출() {
    System.out.println("어디로 외출할까요? 1. 거리 2. 산 3. 놀이공원 4. 상점가 5. 집");
    int 선택 = sc.nextInt();
    sc.nextLine();

    String 결과 = "";

    switch (선택) {
        case 1:
            adjust스트레스(20);
            adjust질병(-10);
            adjust친밀도(10);
            결과 = "거리를 다녀옵니다.\n" + "(스트레스 +20)(질병 -10)(친밀도 +10)";
            break;
        case 2:
            adjust스트레스(20);
            adjust체력(-20);
            adjust친밀도(15);
            결과 = "산을 다녀옵니다.\n" + "(스트레스 +20)(체력 -20)(친밀도 +15)";
            break;
        case 3:
            adjust스트레스(-30);
            adjust친밀도(20);
            결과 = "놀이공원을 다녀옵니다.\n" + "(스트레스 -30)(친밀도 +20)";
            break;
        case 4:
            adjust스트레스(-20);
            adjust친밀도(20);
            결과 = "상점가를 다녀옵니다.\n" + "(스트레스 -20)(친밀도 +20)";
            break;
        case 5:
            결과 = "집으로 돌아왔습니다.\n" + "(수치 변화 없음)";
            break;
    }
}
```

들어주셔서 감사합니다.