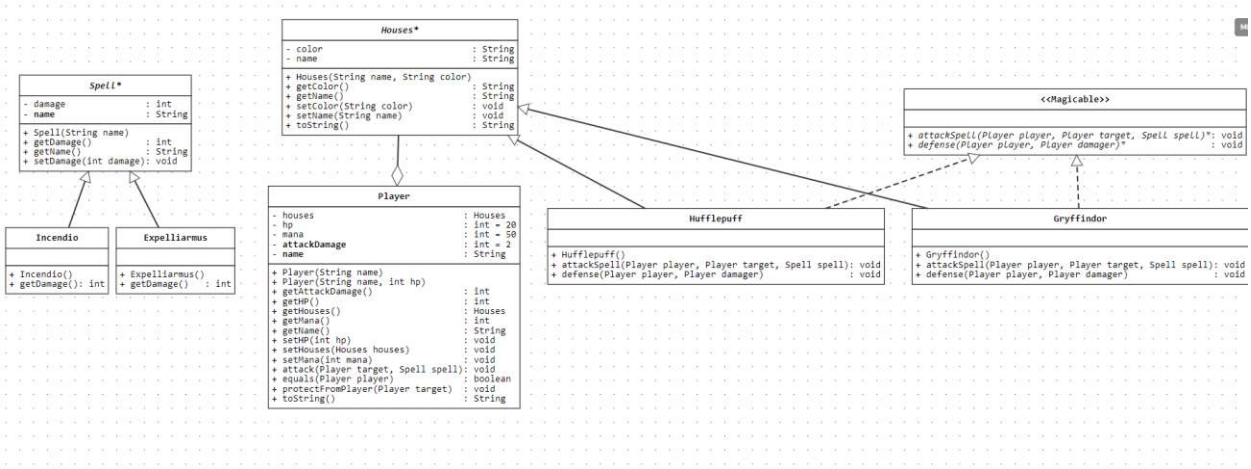
ข้อควรปฏิบัติ (20 คะแนน)

1. ให้นักศึกษาบีบอัดไฟล์ *.java ในโฟลเดอร์ src ของโปรเจกต์ใน IDE ในรูปแบบ ***.zip, *.rar, *.7z เท่านั้น** โดยกำหนดให้ตั้งชื่อไฟล์เป็น **"ชื่อ_นามสกุล"** ด้วย
2. การตรวจข้อสอบปฏิบัติมีข้อกำหนดดังนี้
    - ไม่สามารถประกาศแอททริบิวท์แบบย่อได้
    - ระบบไม่รองรับการละปีกกา { ... } และไม่รองรับการเขียนในรูปแบบย่อ
    - ปีกกาเปิดและปิด { ... } ของคำสั่ง if, else if, else, while, for, รวมถึงของคลาสและเมธฑอด ห้ามไว้บรรทัดเดียวกัน
    - นักศึกษาต้องสร้าง Default Constructor ด้วยตนเอง ถ้านักศึกษาไม่สร้าง ระบบจะไม่สามารถตรวจสอบและให้คิดคะแนนได้
    - นักศึกษาต้องตั้งชื่อคลาสหรือชื่อไฟล์ให้ตรงกับโจทย์ที่กำหนด มิฉะนั้น ระบบจะไม่นำไฟล์หรือคลาสนั้นมาคิดคะแนน

ข้อที่ 1 ให้นักศึกษาสร้างคลาส, คลาสไม่สมบูรณ์ และอินเตอร์เฟส จากคลาสไดอะแกรมต่อไปนี้ โดยที่หนึ่งคลาสหรืออินเตอร์เฟส ต่อหนึ่งไฟล์เท่านั้น ทุกคลาสและอินเตอร์เฟสต้องตั้งค่า access modifier เป็น public โดยกำหนดให้แต่ละอินเตอร์เฟส คลาสไม่สมบูรณ์ และคลาสสมบูรณ์มีรายละเอียดดังต่อไปนี้



ข้อที่ 1.1 ให้นักศึกษาสร้างอินเตอร์เฟส <<Magicable>> โดยมีรายละเอียดดังต่อไปนี้

- ให้เมธอดประกาศ attackSpell(Player player, Player target, Spell spell) แบบไม่สมบูรณ์ดังแสดงในไดอะแกรม
- ให้เมธอดประกาศ defense(Player player, Player damager) แบบไม่สมบูรณ์ดังแสดงในไดอะแกรม

ข้อที่ 1.2 ให้นักศึกษาสร้างคลาสไม่สมบูรณ์ Houses โดยมีรายละเอียดดังต่อไปนี้

- ให้นักศึกษาประกาศแอททริบิวท์ name และ color ตามที่กำหนดในไดอะแกรมข้างต้น
- ให้นักศึกษาสร้างคอนสตัคเตอร์แบบ Default และแบบรับค่า โดยที่
    - สำหรับคอนสตัคเตอร์แบบ Default ให้กำหนดค่าของแต่ละแอททริบิวท์เป็นค่าเริ่มต้นของแต่ละ data type ดังตารางต่อไปนี้

ชนิดข้อมูล (Data type)	int	double	string	char	boolean	Class อื่นๆ
ค่าเริ่มต้น	0	0	“ ”	‘ ’	false	null

    - สำหรับคอนสตัคเตอร์แบบรับค่า ให้นำค่าพารามิเตอร์ที่รับเข้ามาไปกำหนดให้แอททริบิวท์ที่มีชื่อเหมือนกัน
- ให้นักศึกษาสร้างเมธอดประเภท setter() และ getter() แบบปกติของทุกแอททริบิวท์
- ให้นักศึกษาสร้างเมธอด toString() ที่ได้รับการ overridden และคืนค่าดังต่อไปนี้

```
[House] : [แอททริบิวท์ name] , Color : [แอททริบิวท์ color]
```

ข้อที่ 1.3 ให้นักศึกษาสร้างคลาสไม่สมบูรณ์ Spell โดยมีรายละเอียดดังต่อไปนี้

- ให้นักศึกษาประกาศแอททริบิวท์ name เป็นแบบ final และ damage เป็นแบบปกติตามที่กำหนดในไดอะแกรมข้างต้น
- ให้นักศึกษาสร้างคอนสตัคเตอร์แบบ Default และแบบรับค่า โดยที่
    - สำหรับคอนสตัคเตอร์แบบ Default ให้กำหนดค่าของแต่ละแอททริบิวท์เป็นค่าเริ่มต้นของแต่ละ data type ดังตารางต่อไปนี้

ชนิดข้อมูล (Data type)	int	double	string	char	boolean	Class อื่นๆ
ค่าเริ่มต้น	0	0	“ ”	‘ ’	false	null

    - สำหรับคอนสตัคเตอร์แบบรับค่า ให้นำค่าพารามิเตอร์ที่รับเข้ามาไปกำหนดให้แอททริบิวท์ที่มีชื่อเหมือนกัน
- ให้นักศึกษาสร้างเมธอดประเภท setter() และ getter() แบบปกติของทุกแอททริบิวท์

ข้อที่ 1.4 ให้นักศึกษาสร้างคลาส Incendio ซึ่งสอบทอดมาจากคลาส Spell โดยมีรายละเอียดดังต่อไปนี้

- ให้นักศึกษาสร้างคอนสตัคเตอร์เมธอดแบบ<u>ไม่รับค่าเท่านั้น</u> กำหนดให้แอททริบิวท์ name มีชื่อ Incendio
- ให้นักศึกษาสร้างเมธอด getDamage() ได้รับการ overridden ด้วยการคืนค่าเป็น 5

ข้อที่ 1.5 ให้นักศึกษาสร้างคลาส Expelliarmus ซึ่งสอบทอดมาจากคลาส Spell โดยมีรายละเอียดดังต่อไปนี้

- ให้นักศึกษาสร้างคอนสตัคเตอร์เมธอดแบบ<u>ไม่รับค่าเท่านั้น</u> กำหนดให้แอททริบิวท์ name มีชื่อ Expelliarmus
- ให้นักศึกษาสร้างเมธอด getDamage() ได้รับการ overridden และคืนค่าเป็น 7

ข้อที่ 1.6 ให้นักศึกษาสร้างคลาส Hufflepuff ซึ่งสืบทอดมาจากคลาส Houses และ implement มาจากอินเตอร์เฟส Magicable โดยมีรายละเอียดดังต่อไปนี้

- ให้นักศึกษาสร้างคอนสตัคเตอร์เมธอดแบบ<u>ไม่รับค่าเท่านั้น</u> กำหนดให้แอททริบิวท์ name มีค่าเท่ากับ Hufflepuff และแอททริบิวท์ color มีค่าเท่ากับ YELLOW
- ให้นักศึกษาสร้างเมธอด attackSpell(Player player, Player target, Spell spell) ที่เป็นเมธอดสำหรับการโจมตีไปยังผู้เล่นคนอื่น (หรือด้วยพารามิเตอร์ target) ให้ลดค่าแอททริบิวท์ hp ของค่าพารามิเตอร์ target ลงโดยการเอาค่า damage ของพารามิเตอร์ spell ร่วมกับ attackDamage ของพารามิเตอร์ player และลดค่าแอททริบิวท์ mana ลงไป 3 หน่วยของพารามิเตอร์ player ถ้า Spell ของเราสร้างมาจากคลาส Incendio ขณะที่ Expelliarmus ให้ลดค่าแอททริบิวท์ hp ของค่าพารามิเตอร์ target ลงโดยการเอาค่า damage ของพารามิเตอร์ spell ร่วมกับ attackDamage ของพารามิเตอร์ player และลดค่าแอททริบิวท์ mana ลงไป 4 หน่วยของพารามิเตอร์ player และแสดงข้อความต่อไปนี้

```
[[แอททริบิวท์ name ของ player]]: use spell ([แอททริบิวท์ name ของ spell])!
```

- ให้นักศึกษาสร้างเมธอด defense(Player player, Player damager) ที่เป็นเมธอดสำหรับป้องกันจากการโจมตีผู้อื่น (หรือด้วยพารามิเตอร์ damager) ให้เพิ่มค่าแอททริบิวท์ hp ของพารามิเตอร์ player ไป 4 หน่วยและเพิ่มค่าแอททริบิวท์ mana ไป 3 หน่วย และแสดงข้อความต่อไปนี้

```
[[แอททริบิวท์ name ของ player]]: Protego !
```

ข้อที่ 1.7 ให้นักศึกษาสร้างคลาส Gryffindor ซึ่งสืบทอดมาจากคลาส Houses และ implement มาจากอินเตอร์เฟส Magicable โดยมีรายละเอียดดังต่อไปนี้

- ให้นักศึกษาสร้างคอนสตัคเตอร์เมธอดแบบ<u>ไม่รับค่าเท่านั้น</u> กำหนดให้แอททริบิวท์ name มีค่าเท่ากับ Gryffindor และแอททริบิวท์ color มีค่าเท่ากับ RED
- ให้นักศึกษาสร้างเมธอด attackSpell(Player player, Player target, Spell spell) ที่เป็นเมธอดสำหรับการโจมตีไปยังผู้เล่นคนอื่น (หรือด้วยพารามิเตอร์ target) ให้ลดค่าแอททริบิวท์ hp ของค่าพารามิเตอร์ target ลงโดยการเอาค่า damage ของพารามิเตอร์ spell ร่วมกับ attackDamage ของพารามิเตอร์ player และลดค่าแอททริบิวท์ mana ลงไป 4 หน่วยของพารามิเตอร์ player ถ้า Spell ของเราสร้างมาจากคลาส Incendio ขณะที่ Expelliarmus ให้ลดค่าแอททริบิวท์ hp ของค่าพารามิเตอร์ target ลงโดยการเอาค่า damage ของพารามิเตอร์ spell ร่วมกับ attackDamage ของพารามิเตอร์ player และลดค่าแอททริบิวท์ mana ลงไป 3 หน่วยของพารามิเตอร์ player และแสดงข้อความต่อไปนี้

```
[[แอททริบิวท์ name ของ player]]: use spell ([แอททริบิวท์ name ของ spell])!
```

- ให้นักศึกษาสร้างเมธอด defense(Player player, Player damager) ที่เป็นเมธอดสำหรับป้องกันจากการโจมตีผู้อื่น (หรือด้วยพารามิเตอร์ damager) ให้เพิ่มค่าแอททริบิวท์ hp ของพารามิเตอร์ player ไป 3 หน่วยและเพิ่มค่าแอททริบิวท์ mana ไป 4 หน่วย และแสดงข้อความต่อไปนี้

```
[[แอททริบิวท์ name ของ player]]: Episkey !
```

ข้อที่ 1.8 ให้นักศึกษาสร้างคลาส Player โดยมีรายละเอียดดังต่อไปนี้

- ให้นักศึกษาประกาศแอททริบิวท์ name, attackDamage เป็นแบบ final และ house, hp, mana เป็นแบบปกติตามที่กำหนดในไดอะแกรมข้างต้น
- ให้นักศึกษาสร้างคอนสตัคเตอร์แบบ Default และแบบรับค่า โดยที่
    - สำหรับคอนสตัคเตอร์แบบ Default ให้กำหนดค่าของแต่ละแอททริบิวท์เป็นค่าเริ่มต้นของแต่ละ data type ดังตารางต่อไปนี้

ชนิดข้อมูล (Data type)	int	double	string	char	boolean	Class อื่นๆ
ค่าเริ่มต้น	0	0	“ ”	‘ ’	false	null

    - สำหรับคอนสตัคเตอร์แบบรับค่า ให้นำค่าพารามิเตอร์ที่รับเข้ามาไปกำหนดให้แอททริบิวท์ที่มีชื่อเหมือนกัน <u>นอกจากนี้สำหรับค่าแอททริบิวท์ที่ไม่ได้รับเข้ามา ให้นักศึกษากำหนดเป็นค่าเริ่มต้นของแต่ละ data type ต่อไปนี้</u>

ชนิดข้อมูล (Data type)	int	double	string	char	boolean	Class อื่นๆ
ค่าเริ่มต้น	0	0	“ ”	‘ ’	false	null

- ให้นักศึกษากำหนดแอททริบิวท์ของ hp มีค่าเป็นเท่ากับ 20 หน่วยตามที่กำหนดในไดอะแกรมข้างต้น
- ให้นักศึกษากำหนดแอททริบิวท์ของ mana มีค่าเป็นเท่ากับ 50 หน่วยตามที่กำหนดในไดอะแกรมข้างต้น
- ให้นักศึกษากำหนดแอททริบิวท์ของ attackDamage มีค่าเป็นเท่ากับ 2 ตามที่กำหนดในไดอะแกรมข้างต้น
- ให้นักศึกษาสร้างเมธอดประเภท setter() และ getter() แบบปกติของทุกแอททริบิวท์
- ให้นักศึกษาสร้างเมธอด toString() ที่ได้รับการ overridden และคืนค่าดังต่อไปนี้

```
[Player] : [แอททริบิวท์ name] HP: [แอททริบิวท์ hp] Mana: [แอททริบิวท์ mana] || [แอททริบิวท์ houses]
```

- ให้นักศึกษาสร้างเมธอด setHP(int hp) โดยที่พารามิเตอร์ต้องไม่ต่ำกว่า 0 หากต่ำกว่าให้กำหนดแอททริบิวท์ hp มีค่าเท่ากับ 0 และต้องไม่เกิน 20 หากเกินให้กำหนดแอททริบิวท์ hp มีค่าเท่ากับ 20 หน่วย

- ให้นักศึกษาสร้างเมธอด setMana(int mana) โดยที่พารามิเตอร์ต้องไม่ต่ำกว่า 0 หากต่ำกว่าให้กำหนดแอททริบิวท์ mana มีค่าเท่ากับ 0 และต้องไม่เกิน 50 หากเกินให้กำหนดแอททริบิวท์มี mana ค่าเท่ากับ 50 หน่วย
- ให้นักศึกษาสร้างเมธอด equals(Player player) ที่ได้รับการ overloaded และจะคืนเป็น true ต่อเมื่อแอททริบิวท์ name และ houses ของวัตถุทั้ง 2 มีค่าเหมือนกัน แต่ถ้าไม่ใช่จะให้คืนค่าเป็น false
- ให้นักศึกษาสร้างเมธอด attack(Player target, Spell spell) ที่เป็นเมธอดสำหรับการโจมตีไปยังผู้เล่นคนอื่น (หรือด้วยพารามิเตอร์ target) ในคลาส Player ในแต่ละลูกของคลาส Houses จะมีเมธอด attackSpell(Player player, Player target, Spell spell) ให้เรียกใช้งานและหลังโจมตีหากแอททริบิวท์ hp ของ target มีค่าเท่ากับ 0 ให้แสดงข้อความ

  ```
  [DEAD]: [แอททริบิวท์ name ของ target] was killed by [แอททริบิวท์ name]
  ```

- ให้นักศึกษาสร้างเมธอด protectFromPlayer (Player target, Spell spell) ที่เป็นเมธอดสำหรับการป้องกันจากผู้เล่นคนอื่น (หรือด้วยพารามิเตอร์ target) ในคลาส Player ในแต่ละลูกของคลาส Houses จะมีเมธอด defense(Player target) ให้เรียกใช้งาน

กำหนดโค้ดสำหรับทดลองความถูกต้องของโปรแกรมที่นักศึกษาได้พัฒนาขึ้นดังนี้

```
public class Main {
    public static void main(String[] args) {
        Player p1 = new Player("FewPz");
        Player p2 = new Player("Few789");

        p1.setHouses(new Gryffindor());
        p2.setHouses(new Hufflepuff());

        System.out.println(p1);
        System.out.println(p2);

        p1.attack(p2, new Incendio());
        p2.attack(p1, new Expelliarmus());
        p2.attack(p1, new Incendio());
        p1.attack(p2, new Incendio());

        System.out.println(p1);
        System.out.println(p2);

        p2.protectFromPlayer(p1);
        p2.protectFromPlayer(p1);
        p2.protectFromPlayer(p1);

        System.out.println(p1);
        System.out.println(p2);
    }
}
```

ผลลัพท์ที่คาดหวัง

```
[Player] : FewPz HP: 20 Mana: 50 || [House] : Gryffindor , Color : RED
[Player] : Few789 HP: 20 Mana: 50 || [House] : Hufflepuff , Color : YELLOW
[FewPz]: use spell (Incendio)!
[Few789]: use spell (Expelliarmus)!
[Few789]: use spell (Incendio)!
[FewPz]: use spell (Incendio)!
[Player] : FewPz HP: 4 Mana: 42 || [House] : Gryffindor , Color : RED
[Player] : Few789 HP: 6 Mana: 43 || [House] : Hufflepuff , Color : YELLOW
[Few789]: Protego !
[Few789]: Protego !
[Few789]: Protego !
[Player] : FewPz HP: 4 Mana: 42 || [House] : Gryffindor , Color : RED
[Player] : Few789 HP: 18 Mana: 50 || [House] : Hufflepuff , Color : YELLOW
```