

Аналіз патернів проектування у веброзробці та їх застосування у розробці вебзастосунку для автоматизації створення навчального розкладу

Підготував: Мисько Юрій Мирославович

Науковий керівник: Борозенний Сергій Олександрович

Завдання

1. Створити клієнт-серверний застосунок, який автоматизує процес складання шкільного розкладу.
2. Проаналізувати та реалізувати патерни проєктування при розробці вищезгаданого застосунку.

Алгоритм для створення шкільного розкладу

1. Шкала важкості предметів.
2. Пріоритетність днів та уроків.
3. Побажання та доступність вчителів.
4. Обмеження для спарених уроків.
5. Поділ на групи.
6. Тижневе навантаження.
7. Мінімізація вільних часових проміжків між уроками.

Архітектура

Клієнт

 ANGULAR

Презентаційний шар

Шар бізнеслогіки

Шар доступу до серверу

Сервер

 nest

 TYPEORM

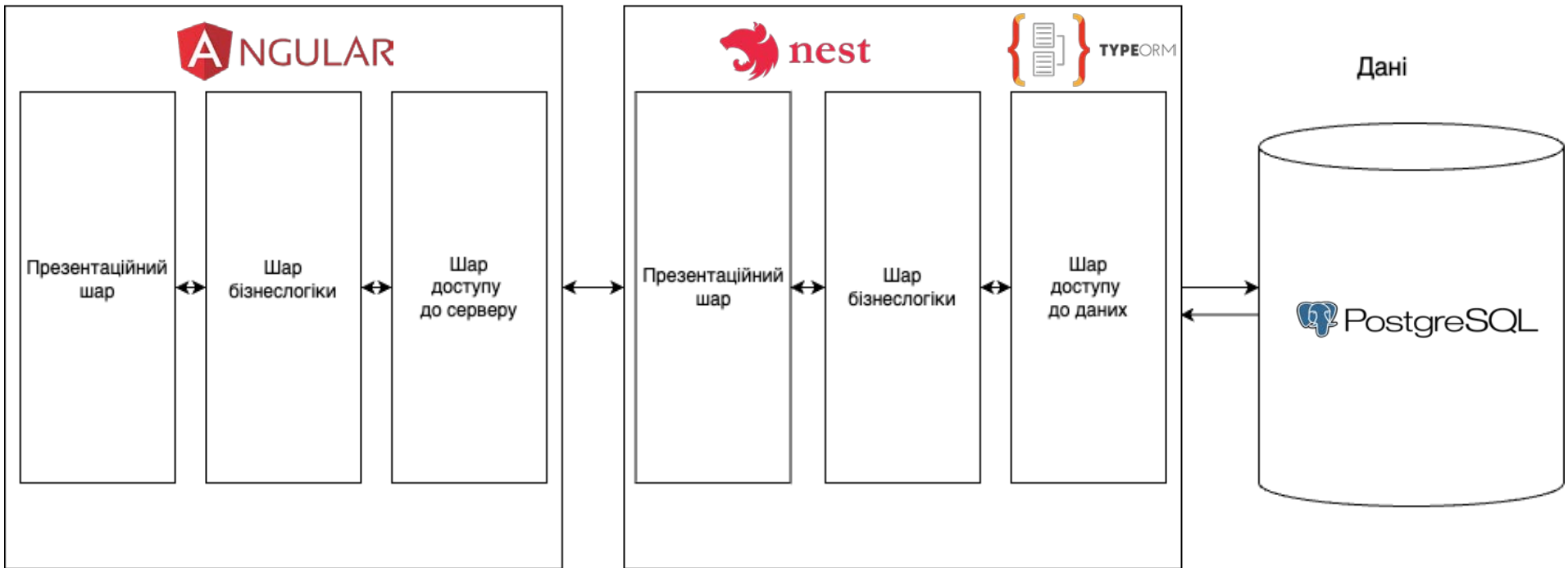
Презентаційний шар

Шар бізнеслогіки

Шар доступу до даних

Дані

 PostgreSQL



Сутності та зв'язки

1. Вчителі
2. Предмети
3. Групи
4. Уроки
5. Користувачі

```
interface Time {
  day: Day;
  slot: Slot;
}

declare interface Teacher {
  id: string;
  name: string;
  subjectKeys?: Subject["key"][];
  notAvailable?: Time[];
}

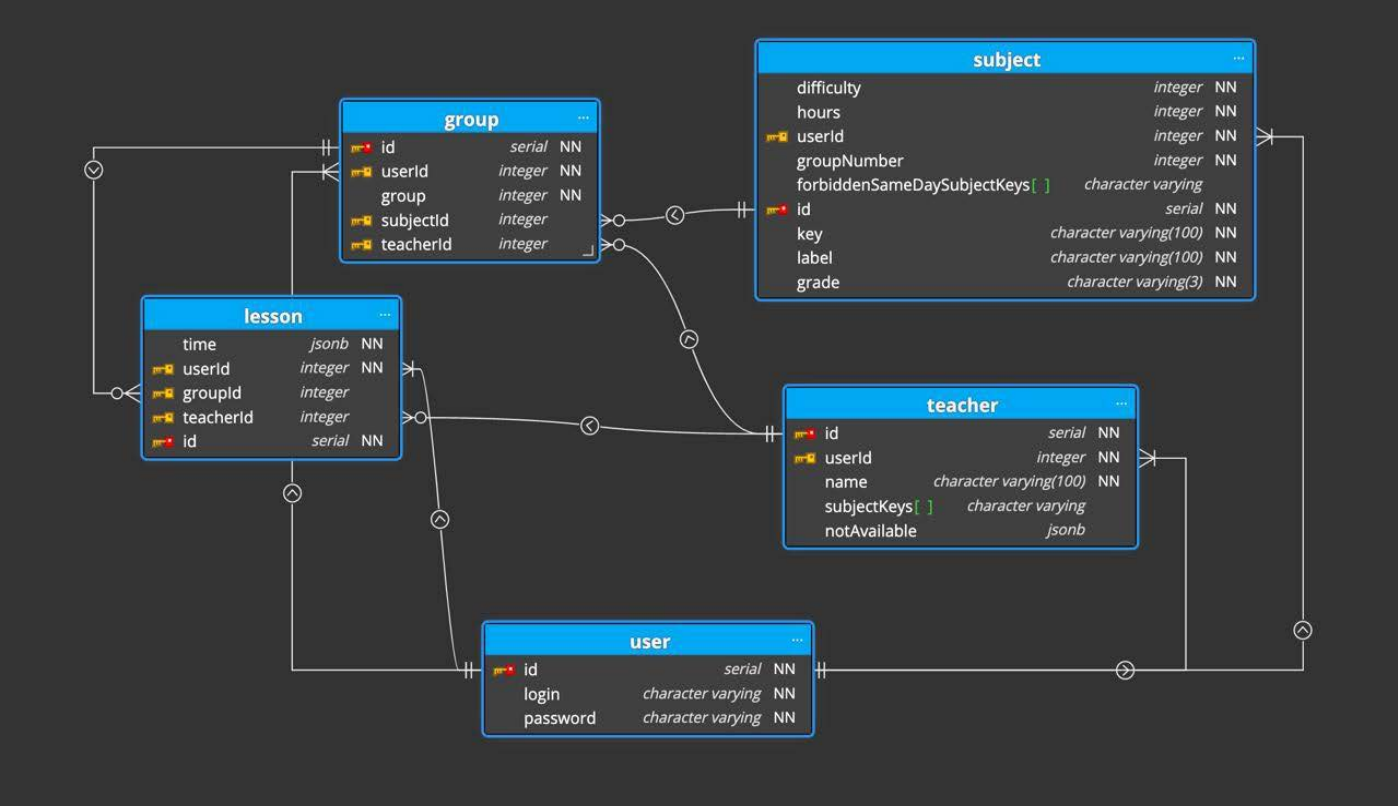
declare interface Lesson {
  id: string;
  time: Time;
  group: Pick<SubjectGroup, "id">;
  teacher: Teacher;
}

declare interface SubjectGroup {
  id: string;
  group: number;
  subject: Subject;
  teacher: Teacher;
}

declare interface Subject {
  id: string;
  key: string;
  label: string;
  difficulty: number;
  grade: Grade;
  hours: number;
  groupNumber: number;
  forbiddenSameDaySubjectKeys?: Subject["key"][];
}

declare interface User {
  login: string;
  password: string;
}
```

Сутності та зв'язки



Структура

```

  ▾ src
    > app
    ▾ decorators
      TS current-user.decorator.ts
    ▾ gurads
      TS auth.gurad.ts
    ▾ middlewares
      TS current-user.middleware.ts
    ▾ modules
      > grades
      > groups
      > lessons
      > subjects
      > teachers
      > users
    > types
      TS main.ts
    > test
```

```

  ▾ features
    > grades
    > login
    > register
    > schedule
    > teachers
  ▾ shared
    > components
    > guards
    > interceptors
  ▾ services
    > data-service
    > download
    > grades
    > groups
    > lessons
    > notifications
    > parallels
    > schedule
    > subjects
    > teachers
    > users
  TS types.d.ts
```

Сторінка вибору паралелей для класів

School1

Класи

Предмети

Вчителі

Забронювати час

Вчителі/Предмети

Вчителі/Класи

Розклад

Вихід

Виберіть паралелі для класів



5 клас: Б
6 клас:
7 клас:
8 клас:
9 клас:
10 клас:
11 клас:
--
А
Б
В
Г
Д

Зберегти

Сторінка вибору предметів і тижневого навантаження

School1

Класи

Предмети

Вчителі

Забронювати час

Вчителі/Предмети

Вчителі/Класи

Розклад

Вихід

Виберіть предмети, к-ть груп і тижневе навантаження для кожного класу

Виберіть клас
5

Вибрано	Назва предмету	К-ть груп	К-ть годин
<input checked="" type="checkbox"/>	Українська мова	<input type="text" value="1"/>	<input type="text" value="3"/>
<input type="checkbox"/>	Фізична культура	<input type="text" value="1"/>	<input type="text" value="3"/>
<input type="checkbox"/>	Трудове навчання	<input type="text" value="1"/>	<input type="text" value="3"/>
<input type="checkbox"/>	Образотворче Мистецтво	<input type="text" value="1"/>	<input type="text" value="3"/>
<input type="checkbox"/>	Музичне Мистецтво	<input type="text" value="1"/>	<input type="text" value="3"/>
<input checked="" type="checkbox"/>	Українська література	<input type="text" value="1"/>	<input type="text" value="3"/>
<input checked="" type="checkbox"/>	Англійська мова	<input type="text" value="1"/>	<input type="text" value="3"/>
<input checked="" type="checkbox"/>	Зарубіжна література	<input type="text" value="1"/>	<input type="text" value="3"/>
<input checked="" type="checkbox"/>	Історія (Вступ до історії)	<input type="text" value="1"/>	<input type="text" value="3"/>
<input checked="" type="checkbox"/>	Математика	<input type="text" value="1"/>	<input type="text" value="3"/>
<input checked="" type="checkbox"/>	Природознавство	<input type="text" value="1"/>	<input type="text" value="3"/>
<input checked="" type="checkbox"/>	Інформатика	<input type="text" value="1"/>	<input type="text" value="3"/>
<input checked="" type="checkbox"/>	Основи здоров'я	<input type="text" value="1"/>	<input type="text" value="3"/>

Зберегти всі

Сторінка введення списку викладачів

School1

[Класи](#)

[Предмети](#)

[Вчителі](#)

[Забронювати час](#)

[Вчителі/Предмети](#)

[Вчителі/Класи](#)

[Розклад](#)

[Вихід](#)

Введіть ПІБ викладачів

Іваненко Олександр Миколайович
Савченко Марія Василівна
Петренко Вікторія Ігорівна
Лисенко Андрій Сергійович
Жукова Катерина Володимирівна
Васильєв Олег Петрович
Кравченко Юлія Олексіївна
Гончарова Ніна Вікторівна
Шевченко Дмитро Ярославович

Зберегти

Сторінка вибору доступності вчителів

School1

Класи

Предмети

Вчителі

Забронювати час

Вчителі/Предмети

Вчителі/Класи

Розклад

Вихід

Виберіть час, коли викладач НЕ зможе вести уроки

Виберіть вчителя

Іваненко Олександр Миколайович

Понеділок	Вівторок	Середа	Четвер	П'ятниця
<input type="checkbox"/> 0-й урок	<input type="checkbox"/> 0-й урок	<input type="checkbox"/> 0-й урок	<input type="checkbox"/> 0-й урок	<input type="checkbox"/> 0-й урок
<input checked="" type="checkbox"/> 1-й урок	<input type="checkbox"/> 1-й урок	<input type="checkbox"/> 1-й урок	<input type="checkbox"/> 1-й урок	<input type="checkbox"/> 1-й урок
<input type="checkbox"/> 2-й урок	<input type="checkbox"/> 2-й урок	<input type="checkbox"/> 2-й урок	<input type="checkbox"/> 2-й урок	<input type="checkbox"/> 2-й урок
<input type="checkbox"/> 3-й урок	<input type="checkbox"/> 3-й урок	<input type="checkbox"/> 3-й урок	<input type="checkbox"/> 3-й урок	<input type="checkbox"/> 3-й урок
<input checked="" type="checkbox"/> 4-й урок	<input type="checkbox"/> 4-й урок	<input type="checkbox"/> 4-й урок	<input type="checkbox"/> 4-й урок	<input type="checkbox"/> 4-й урок
<input type="checkbox"/> 5-й урок	<input type="checkbox"/> 5-й урок	<input type="checkbox"/> 5-й урок	<input type="checkbox"/> 5-й урок	<input type="checkbox"/> 5-й урок
<input type="checkbox"/> 6-й урок	<input type="checkbox"/> 6-й урок	<input type="checkbox"/> 6-й урок	<input type="checkbox"/> 6-й урок	<input type="checkbox"/> 6-й урок
<input type="checkbox"/> 7-й урок	<input type="checkbox"/> 7-й урок	<input type="checkbox"/> 7-й урок	<input type="checkbox"/> 7-й урок	<input type="checkbox"/> 7-й урок

Перейти до вибору предметів

Сторінка розподілу викладачів за предметами

School1

Класи

Предмети

Вчителі

Забронювати час

Вчителі/Предмети

Вчителі/Класи

Розклад

Вихід

Розподіл викладачів по предметах

Пошук	
Викладач	Предмети
Іваненко Олександр Миколайович	Предмети Українська мова, Фізична культура, Трудове навчання
Савченко Марія Василівна	Предмети Образотворче Мистецтво, Музичне Мистецтво
Петренко Вікторія Ігорівна	Предмети Українська література, Англійська мова, Зарубіжна література
Лисенко Андрій Сергійович	Предмети Історія (Вступ до історії), Математика, Природознавство, Інфо...
Жукова Катерина Володимирівна	Предмети Основи здоров'я, Історія України та Всесвітня історія (інтегро...
Васильєв Олег Петрович	Предмети Алгебра, Геометрія, Фізика, Хімія
Кравченко Юлія Олексівна	Предмети Громадянська освіта, Біологія і екологія, Захист України
Гончарова Ніна Вікторівна	Предмети Фізика і астрономія, Технології, Фінансова грамотність
Шевченко Дмитро Ярославович	Предмети Українська мова, Фізична культура, Трудове навчання, образо...

Зберегти

Сторінка розподілу викладачів за класами

School1

[Класи](#)

[Предмети](#)

[Вчителі](#)

[Забронювати час](#)

[Вчителі/Предмети](#)

[Вчителі/Класи](#)

[Розклад](#)

[Вихід](#)

Виберіть в яких класах викладачі ведуть предмети

Пошук

Викладач	К-ть годин	Предмет	5(1)
Іваненко Олександр Миколайович	6	Українська мова	<input checked="" type="checkbox"/>
Іваненко Олександр Миколайович	6	Українська література	<input checked="" type="checkbox"/>
Савченко Марія Василівна	0	Фізична культура	<input type="checkbox"/>
Савченко Марія Василівна	0	Трудове навчання	<input type="checkbox"/>
Савченко Марія Василівна	0	Образотворче Мистецтво	<input type="checkbox"/>
Петренко Вікторія Ігорівна	9	Англійська мова	<input checked="" type="checkbox"/>
Петренко Вікторія Ігорівна	9	Зарубіжна література	<input checked="" type="checkbox"/>
Петренко Вікторія Ігорівна	9	Історія (Вступ до історії)	<input checked="" type="checkbox"/>
Лисенко Андрій Сергійович	9	Математика	<input checked="" type="checkbox"/>
Лисенко Андрій Сергійович	9	Природознавство	<input checked="" type="checkbox"/>
Лисенко Андрій Сергійович	9	Інформатика	<input checked="" type="checkbox"/>
Жукова Катерина Володимирівна	3	Основи здоров'я	<input checked="" type="checkbox"/>
Жукова Катерина Володимирівна	3	Історія України та Всесвітня історія (інтегрований курс)	<input type="checkbox"/>

Зберегти

Сторінка зі згенерованим розкладом

School1

Класи

Предмети

Вчителі

Забронювати час

Вчителі/Предмети

Вчителі/Класи

Розклад

Вихід

Розклад

Пошук

		Класи			Вчителі	
Клас	День/№	Понеділок	Вівторок	Середа	Четвер	П'ятниця
	0	-/-	-/-	-/-	-/-	-/-
5	1	Природознавство: Група 1 - Лисенко Андрій Сергійович	Зарубіжна література: Група 1 - Петренко Вікторія Ігорівна	Зарубіжна література: Група 1 - Петренко Вікторія Ігорівна	Зарубіжна література: Група 1 - Петренко Вікторія Ігорівна	Природознавство: Група 1 - Лисенко Андрій Сергійович
	2	Англійська мова: Група 1 - Петренко Вікторія Ігорівна	Математика: Група 1 - Лисенко Андрій Сергійович	Математика: Група 1 - Лисенко Андрій Сергійович	Математика: Група 1 - Лисенко Андрій Сергійович	Англійська мова: Група 1 - Петренко Вікторія Ігорівна
	3	Історія (Вступ до історії): Група 1 - Петренко Вікторія Ігорівна	Англійська мова: Група 1 - Петренко Вікторія Ігорівна	Історія (Вступ до історії): Група 1 - Петренко Вікторія Ігорівна	Історія (Вступ до історії): Група 1 - Петренко Вікторія Ігорівна	Українська література: Група 1 - Іваненко Олександр Миколайович
	4	Українська мова: Група 1 - Іваненко Олександр Миколайович	Українська література: Група 1 - Іваненко Олександр Миколайович	Українська література: Група 1 - Іваненко Олександр Миколайович	Українська мова: Група 1 - Іваненко Олександр Миколайович	Українська мова: Група 1 - Іваненко Олександр Миколайович
	5	Інформатика: Група 1 - Лисенко Андрій Сергійович	Природознавство: Група 1 - Лисенко Андрій Сергійович	Інформатика: Група 1 - Лисенко Андрій Сергійович	Інформатика: Група 1 - Лисенко Андрій Сергійович	Основи здоров'я: Група 1 - Жукова Катерина Володимирівна
	6	-/-	Основи здоров'я: Група 1 - Жукова Катерина Володимирівна	Основи здоров'я: Група 1 - Жукова Катерина Володимирівна	-/-	-/-
	7	-/-	-/-	-/-	-/-	-/-
	0	-/-	-/-	-/-	-/-	-/-
	1	-/-	-/-	-/-	-/-	-/-
	2	-/-	-/-	-/-	-/-	-/-

Зберегти PDF

Сторінка зі згенерованим розкладом для вчителів

School1

Класи

Предмети

Вчителі

Забронювати час

Вчителі/Предмети

Вчителі/Класи

Розклад

Вихід

Розклад

Пошук

Викладач	День/ №	Класи		Вчителі		
		Понедлок	Вівторок	Середа	Четвер	П'ятниця
Івченко Олександр Миколайович	0	-/-	-/-	-/-	-/-	-/-
	1	-/-	-/-	-/-	-/-	-/-
	2	-/-	-/-	-/-	-/-	-/-
	3	-/-	-/-	-/-	-/-	Українська література: 5(1)
	4	Українська мова: 5(1)	Українська література: 5(1)	Українська література: 5(1)	Українська мова: 5(1)	Українська мова: 5(1)
	5	-/-	-/-	-/-	-/-	-/-
	6	-/-	-/-	-/-	-/-	-/-
	7	-/-	-/-	-/-	-/-	-/-
Савченко Марія Василівна	0	-/-	-/-	-/-	-/-	-/-
	1	-/-	-/-	-/-	-/-	-/-
	2	-/-	-/-	-/-	-/-	-/-
	3	-/-	-/-	-/-	-/-	-/-
	4	-/-	-/-	-/-	-/-	-/-
	5	-/-	-/-	-/-	-/-	-/-
	6	-/-	-/-	-/-	-/-	-/-
	7	-/-	-/-	-/-	-/-	-/-
	0	-/-	-/-	-/-	-/-	-/-

Зберегти PDF

Патерни проєктування

1. Породжувальні
2. Структурні
3. Поведінкові

Спостерігач(Observer)

Спостерігач



```
fetchData<T>(url: string) {
  return this.http.get<T>(url).pipe(
    tap( observerOrNext: () => {
      this.isLoadingSubject.next( value: true);
    } ),
    catchError( selector: (error) => {
      this.notificationsService.show(error.message);
      throw error;
    } ),
    finalize( callback: () => {
      this.isLoadingSubject.next( value: false);
    } )
  );
}
```

```
fetchGrades() {
  this.dataService.fetchData<GradeObject[]>( url: "grades").subscribe( observerOrNext: {
    next: (grades : GradeObject[] ) => {
      this.gradesSubject.next(grades.map((g : GradeObject ) => g.grade));
    },
  });
}
```

Переваги використання патерну Спостерігач

1. Забезпечує взаємодію з мінімальним рівнем залежності.
2. Забезпечує гнучку передачу інформації між об'єктами.
3. Спостерігачі можуть бути додані/видалені під час виконання програми.

Недоліки використання патерну Спостерігач

1. Проблеми з продуктивністю або витокami пам'яті.
2. Проблеми з конкурентністю або порядком виконання.

Ін'єкція залежностей

Ін'єкція залежностей



Ін'єкція залежностей

```
@Injectable({
    providedIn: "root",
})
export class DataService {
    private isLoadingSubject: BehaviorSubject<boolean> = new Beh
    isLoading$: Observable<boolean> = this.isLoadingSubject.asOb

    constructor(
        private http: HttpClient,
        private notificationsService: NotificationsService
    ) {
    }
}
```

```
@Injectable({
    providedIn: "root"
})
export class NotificationsService {
    constructor(private snackBar: MatSnackBar) {}

    show(message: string, config?: MatSnackBarConfig) {
        this.snackBar.open(
            message,
            action: undefined,
            (config ?? { duration: 3000 })
        );
    }
}
```

Переваги використання патерну Ін'єкція залежностей

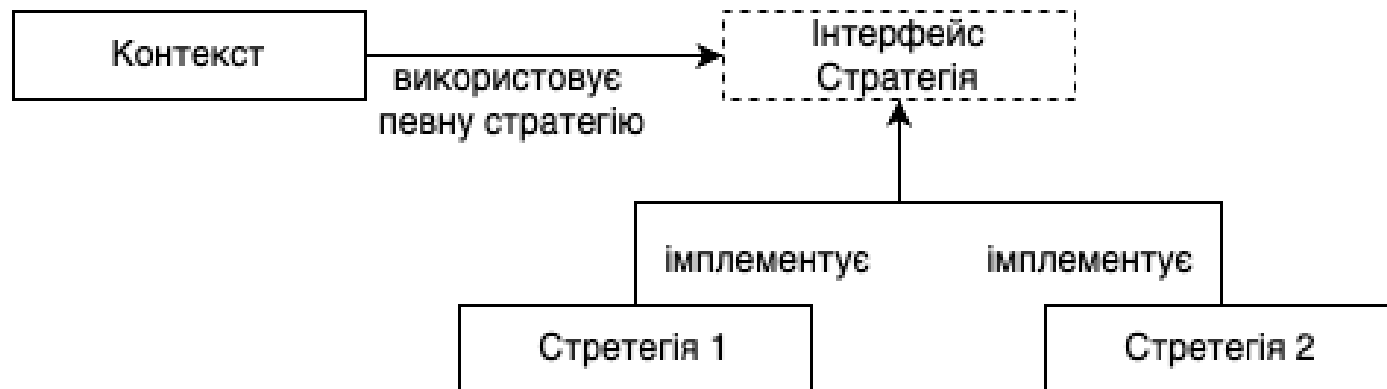
1. Полегшує тестування.
2. Зменшує складність та взаємозалежність коду.

Недоліки використання патерну Ін'єкція залежностей

1. Можуть виникати циклічні залежності.
2. Збільшує к-ть класів в системі.

Стратегія

Стратегія



Стратегія для перевірки доступності проведення уроку

1. Чи уже є урок для даного класу у цей час?
2. Чи доступний викладач у даний час?
3. Чи немає заборонених у один день предметів з даним предметом?

```
interface Availability {
    available: boolean;
    reason?: string
}
export interface AvailabilityCheckStrategy {
    checkAvailability(subject: Subject, time: Time, allowPairedLessons?: boolean): Availability;
}
```

Стратегія для перевірки доступності проведення уроку

```
export class TeacherAvailabilityStrategy implements AvailabilityCheckStrategy {
  constructor(private lessonsService: LessonsService, private groupsService: GroupService) {}
```

```
  checkAvailability(subject: Subject, time: Time) {
    const subjectGroups : SubjectGroup[] = this.groupsService
      .findGroups(subject.id);

    const isTeacherAvailable : boolean = subjectGroups
      .some((group : SubjectGroup) => {
        const { teacher : Teacher } = group;
        const isTeacherBusy : boolean = this.lessonsService.lessons
          .some((lesson : Lesson) => lesson.teacher.id === teacher.id &&
            lesson.time.slot === time.slot);
        const isTeacherBlocked : boolean | undefined = teacher?.notAvailable
          ?.some( predicate: (blockedLessonTime : Time) => blockedLessonTime
            && blockedLessonTime.slot === time.slot);
        return !isTeacherBusy && !isTeacherBlocked;
      });

    if (!isTeacherAvailable) {
      return {
        available: false,
        reason: "Вчитель зайнятий в цей час"
      };
    }
    return { available: true };
  }
}
```

```
export class ForbiddenSubjectsStrategy implements AvailabilityCheckStrategy {
  constructor(private lessonsService: LessonsService, private groupsService: GroupService) {}
```

```
  checkAvailability(subject: Subject, time: Time) {
    const { grade : Grade, forbiddenSameDaySubjectKeys : string[] | undefined } = subject;
    const lessonsThisDay : Lesson[] = this.lessonsService
      .findLessonsByGradeAndTime(grade, time: { day: time.day });
    const groupIdsThisDay : string[] = lessonsThisDay.map((l : Lesson) => l.groupId);
    const gradeGroups : SubjectGroup[] = this.groupsService
      .findGradeGroups(grade);
    const groupsThisDay : SubjectGroup[] = gradeGroups
      .filter((g : SubjectGroup) => groupIdsThisDay.includes(g.id));
    const subjectsThisDay : string[] = groupsThisDay
      .map((g : SubjectGroup) => g.subject.key);

    if (forbiddenSameDaySubjectKeys && forbiddenSameDaySubjectKeys.find((disabledKey) =>
      disabledKey === subject.key)) {
      return {
        available: false,
        reason: `Предмет ${subject.key} не може бути в один день з ${forbiddenSameDaySubjectKeys}`
      };
    }
    return { available: true };
  }
}
```

N2



Стратегія для перевірки доступності проведення уроку

```
export class TeacherAvailabilityStrategy implements AvailabilityCheckStrategy {
  constructor(private lessonsService: LessonsService, private groupsService: GroupService) {}
```

```
  checkAvailability(subject: Subject, time: Time) {
    const subjectGroups : SubjectGroup[] = this.groupsService
      .findGroups(subject.id);

    const isTeacherAvailable : boolean = subjectGroups
      .some((group : SubjectGroup) => {
        const { teacher : Teacher } = group;
        const isTeacherBusy : boolean = this.lessonsService.lessons
          .some((lesson : Lesson) => lesson.teacher.id === teacher.id &&
            lesson.time.slot === time.slot);
        const isTeacherBlocked : boolean | undefined = teacher?.notAvailable
          ?.some( predicate: (blockedLessonTime : Time) => blockedLessonTime
            && blockedLessonTime.slot === time.slot);
        return !isTeacherBusy && !isTeacherBlocked;
      });

    if (!isTeacherAvailable) {
      return {
        available: false,
        reason: "Вчитель зайнятий в цей час"
      };
    }
    return { available: true };
  }
}
```

```
export class ForbiddenSubjectsStrategy implements AvailabilityCheckStrategy {
  constructor(private lessonsService: LessonsService, private groupsService: GroupService) {}
```

```
  checkAvailability(subject: Subject, time: Time) {
    const { grade : Grade, forbiddenSameDaySubjectKeys : string[] | undefined } = subject;
    const lessonsThisDay : Lesson[] = this.lessonsService
      .findLessonsByGradeAndTime(grade, time: { day: time.day });
    const groupIdsThisDay : string[] = lessonsThisDay.map((l : Lesson) => l.groupId);
    const gradeGroups : SubjectGroup[] = this.groupsService
      .findGradeGroups(grade);
    const groupsThisDay : SubjectGroup[] = gradeGroups
      .filter((g : SubjectGroup) => groupIdsThisDay.includes(g.id));
    const subjectsThisDay : string[] = groupsThisDay
      .map((g : SubjectGroup) => g.subject.key);

    if (forbiddenSameDaySubjectKeys && forbiddenSameDaySubjectKeys.find((disabledKey) =>
      disabledKey === subject.key)) {
      return {
        available: false,
        reason: `Предмет ${subject.key} не може бути в один день з ${forbiddenSameDaySubjectKeys}`
      };
    }
    return { available: true };
  }
}
```

N2



Стратегія для перевірки доступності проведення уроку

```
@Injectable({
  providedIn: "root"
})
export class LessonAvailabilityCheckerService implements AvailabilityCheckStrategy {
  private strategies: AvailabilityCheckStrategy[];

  constructor(
    private scheduleMatrixService: ScheduleMatrixService,
    private lessonsService: LessonsService,
    private groupsService: GroupsService
  ) {
    this.strategies = [
      new SlotOccupancyStrategy(scheduleMatrixService),
      new ForbiddenSubjectsStrategy(lessonsService, groupsService),
      new PairedLessonsStrategy(lessonsService, groupsService),
      new TeacherAvailabilityStrategy(lessonsService, groupsService)
    ];
  }

  checkAvailability(subject: Subject, time: Time, allowPairedLessons?: boolean) {
    for (const strategy : AvailabilityCheckStrategy of this.strategies) {
      const result : Availability = strategy.checkAvailability(subject, time, allowPairedLessons);
      if (!result.available) {
        return result;
      }
    }
    return { available: true };
  }
}
```

Переваги використання патерну Стратегія

1. Сприяє принципу єдиної відповідальності.
2. Полегшує тестування.
3. Полегшує читання коду.

Недоліки використання патерну Стратегія

1. Збільшує к-ть об'єктів в системі.

Висновки

1. В результаті роботи було створено гнучкий та ефективний інструмент, що значно прискорює та полегшує процес створення шкільного розкладу
2. Патерни забезпечують покращення процесів розробки. Їх використання допомагає створювати компоненти, що взаємодіють через інтерфейси, що знижує прямі залежності та забезпечує більшу гнучкість у розширенні та модифікації системи. Також патерни такі як Спостерігач, В чи Декоратор сприяють слабкому зв'язуванню компонентів, що робить систему менш чутливою до змін.
3. Патерни проектування не є універсальними рішеннями.

Дякую за увагу!