

Mudanças em relação à Etapa anterior

Colocamos todo conteúdo previsto do diagrama de classes em código implementada, porém com os nomes ajustados em inglês.

Na superclasse *Task* (tarefa), as alterações foram a retirada de disciplina dos atributos, visto que a classe tarefa está dentro de uma disciplina e não precisaria ter essa informação dentro de tarefa, além disso, foi criado método `getStatus` e foi removido o método `visualizarTarefa`, pois isso se dá diretamente na interface visual. Na subclasse *Work* foi ajustado o único atributo para *todo*, visto que é importante ao ter um trabalho extenso, poder colocar o que precisa ser feito para concluí-lo.

Implementação

A classe *Task* é a superclasse de *Exam*, *Lab* e *Work*, usando a ideia de herança, criamos atributos e métodos que são comuns a todas as *tasks*. Os atributos comuns são *name*, *description*, *date* e *status*. *Status* informa se foi concluído ou não. Além do construtor com todos os atributos, temos os métodos *getters* e *setters* de todos os atributos.

A classe *Work* é subclasse de *Task*, como atributo exclusivamente dela temos *todo*, que irá uma explicação do que precisa ser feito para o trabalho e, junto disso, terá o *getter* e *setter* desse atributo, o construtor usando o “super” para os atributos da classe-mãe. A classe *Lab* é também subclasse de *Task*, como atributo criado nela temos *room* e nos métodos o *getter* e *setter* de *room*, além do construtor. Na subclasse *Exam* temos dois atributos *room* e *content*, e seus *getters* e *setters* e o construtor.

Teste

Para a superclasse *Task*, casos de teste: Construtor e *getters* é testado os valores atribuídos ao criar o objeto pelo construtor e o de *Setters*, onde busca modificar os atributos após a criação.

Para a subclasse *Work*, casos de teste: Construtor é testado a herança dos atributos e o *todo* e o de *getters* e *setters* para ver se os valores estão sendo atualizados corretamente. Os testes da subclasse *Exam* são semelhantes a anterior, primeiro faz-se um teste para verificar a herança e os novos atributos da subclasse e depois testa se *getters* e *setters* modificam valores corretamente e o mesmo ocorre nos testes da subclasse *Lab*.

Detalhar testes unitários e colocar os casos de teste

Executável (aplicação, interface)