Universidad de Costa Rica

Escuela de Ciencias de la Computación e Informática CI-0136 Diseño de Software

Tema: Laboratorio sobre Composición y Cadena de Responsabilidades

Integrantes

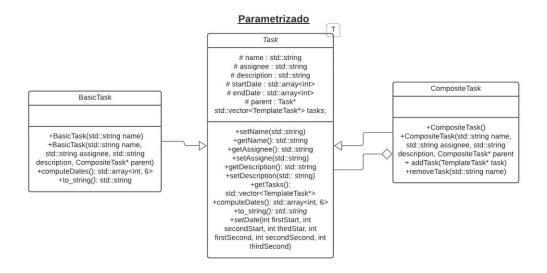
Andrey Gerardo Elizondo Vargas B92732 David Atias Ramos B90807 Bryan Umaña B87997

Profesores:

Alan Calderón Castro

19 de Setiembre 2021

Polimorfico Task # name : std::string # assignee : std::string # description : std::string # startDate : std::arrav<int> ProjectManager # endDate : std::array<int> # parent : Task* projects: std::vector<CompositeTask> +setName(std::string) + CreateProject(std::string name, BasicTask +getName(): std::string std::string inCharge) +getAssignee(): std::string + removeProject(std::string name) + printProjectState() -setAssigne(std::string) +getDescription(): std::string + getProjects() : std::vector<CompositeTask>* +setDescription(std:: string) +BasicTask(std::string name) +computeDates(): std::array<int, 6> +BasicTask(std::string name, + to_string(): std::string std::string assignee, std::string +setDate(int firstStart, int description, CompositeTask* parent) secondStart, int thirdStar, int CompositeTask firstSecond, int secondSecond, int thirdSecond) - tasks : std::vector<Task*> +CompositeTask(std::string name) +CompositeTask(std::string name. std::string assignee, std::string description. CompositeTask* parent + addTask(Task* task) +removeTask(std::string name)



¿Qué ventaja se puede obtener al aplicar esta táctica (parametrizado)?

Naturalmente al implementar las clases heredadas mediante template brinda la ventaja de anidar las distintas clases como si fuesen capas una sobre otra, dando gran facilidad para incorporar múltiples niveles. Además, es difícil acceder a métodos de las clases heredadas desde la clase base al aplicar solo polimorfismo, lo que en ocasiones nos lleva a redundar código.