

ACH2003 - Computação Orientada a Objetos

EP1 – Pong!

Introdução

Você foi recentemente contratado pela *EACH Game Dev. Co.* para atuar no desenvolvimento de **Pong!** seu mais novo e aguardado lançamento. Entretanto, poucos dias antes do início do seu trabalho, o único desenvolvedor da equipe deixou o emprego para trabalhar na maior concorrente da empresa. Cabe a você agora a tarefa de finalizar o desenvolvimento do projeto **Pong!**

O projeto **Pong!** é escrito Java, e é composto pelas seguintes classes:

- **Pong**: classe principal do jogo (contém o método `main` que gerencia o todo andamento da partida e a interação entre os demais elementos).
- **GameLib/MyFrame/MyKeyAdapter**: três classes que implementam funcionalidades gráficas (criação de janela em modo gráfico, métodos para desenhos de formas geométricas) e para processar entrada via teclado.
- **Wall**: implementa as paredes do jogo.
- **Ball**: implementa a bola do jogo.
- **Player**: implementa os jogadores (controláveis) pelos usuários.
- **Score**: implementa o placar do jogo.

Felizmente, algumas destas classes já estavam 100% prontas quando o antigo desenvolvedor deixou a empresa: **Pong**, **GameLib**, **MyFrame**, **MyKeyAdapter** e **Wall**. Entretanto, o antigo desenvolvedor apagou, por acidente, os códigos fontes destas classes, e só restaram os arquivos `.class`. Desta forma, não há a possibilidade modificar o comportamento destas classes, de modo que as classes restantes, que deverão ser implementadas por você, precisam obrigatoriamente “conversar em sintonia” com as classes que já estão prontas.

Para ajudar, um esqueleto de cada uma das classes que precisam ser implementadas (**Ball**, **Player** e **Score**) já foi disponibilizado para você. Desta forma, sua tarefa consiste em completar cada um dos arquivos fontes de modo que cada classe implemente o comportamento esperado delas, fazendo com que o jogo passe a funcionar de forma plena. É extremamente importante que os construtores e as declarações de métodos fornecidos nos esqueletos das classes não sejam alterados, tanto em relação ao nome, quanto em relação aos parâmetros recebidos. Juntamente com os esqueletos das classes, também foi disponibilizado para você a documentação das mesmas em formato *javadoc* explicando o papel de cada método e parâmetro.

Entrega

Este Exercício-Programa pode ser feito de modo individual ou em dupla, e deve ser entregue na tarefa disponível no portal eDisciplinas até o dia 27 de junho de 2021.

Você deve entregar um arquivo compactado nos formatos `.zip` ou `.tar.xz` contendo:

- um arquivo texto `LEIAME.txt`, com o nome completo e número USP do(s) autor(es) e com instruções detalhadas para a **compilação e execução** do programa (**em linha de comando**; não dê instruções que envolvam o uso de alguma IDE específica). Encorajamos (mas não se sintam obrigados) a fazer o uso de uma ferramenta que facilite a compilação de um projeto como o GNU Make¹ (mais simples, amplamente adotada pela comunidade de desenvolvimento de código livre e a ferramenta usada para compilar, por exemplo, o código do próprio Linux) ou o Gradle² (um pouco mais complicada, uma ferramenta utilizada pela comunidade de desenvolvimento Java e adotada, por exemplo, pela comunidade Android).
- um diretório `src`, com todos os arquivos que fazem parte do código-fonte do programa e que são necessários para compilar o programa.

Ambiente de correção

O Exercício-Programa será avaliado em um computador equipado com o **sistema operacional Linux** e com os seguintes softwares e versões:

```
$ java -version
openjdk version "11.0.11" 2021-04-20
OpenJDK Runtime Environment (build 11.0.11+9-post-Debian-1)
OpenJDK 64-Bit Server VM (build 11.0.11+9-post-Debian-1, mixed
mode, sharing)
```

```
$ make --version
GNU Make 4.3
Compilado para x86_64-pc-linux-gnu
Copyright (C) 1988-2020 Free Software Foundation, Inc.
```

¹ GNU Make: <https://www.gnu.org/software/make/>

² Gradle: <https://gradle.org/>

```
$ gradle -version
```

```
-----  
Gradle 4.4.1  
-----
```

```
Build time: 2012-12-21 00:00:00 UTC
```

```
Revision: none
```

```
Groovy: 2.4.17
```

```
Ant: Apache Ant(TM) version 1.10.7 compiled on October 24  
2019
```

```
JVM: 11.0.11 (Debian 11.0.11+9-post-Debian-1)
```

```
OS: Linux 5.10.0-6-amd64 amd64
```